

1

# XML Paper 2 Specification

---

3 XPS Specification and Reference Guide

4

5

6

7

8 Base Document, Working Draft 1.0.1  
9 Submitted to Ecma TC46 by Microsoft  
10 September 2007

11

12

13

14 This document is "a work in progress". It is the starting point of discussions within Ecma  
15 Technical Committee TC46. It was adapted from the original XPS specification as published by  
16 Microsoft, to match standards editing style guidelines more closely, especially in the area of  
17 distinguishing between normative and informative text.

18 A number of formatting and textual issues remain to be addressed, as well as the bulk of the  
19 work of TC46 in amending the text to meet the scope and goals of the committee. Comments  
20 on this draft are very welcome; this early version is provided to enable the consistent use of  
21 clause and subclause references in those comments.

22



1	<b>Contents</b>	
2	<b>1. SCOPE</b> .....	<b>1</b>
3	<b>2. CONFORMANCE</b> .....	<b>III</b>
4	<b>3. NORMATIVE REFERENCES</b> .....	<b>V</b>
5	<b>4. DEFINITIONS</b> .....	<b>VII</b>
6	<b>5. NOTATIONAL CONVENTIONS</b> .....	<b>XI</b>
7	<b>6. ACRONYMS AND ABBREVIATIONS</b> .....	<b>XIII</b>
8	<b>7. GENERAL DESCRIPTION</b> .....	<b>15</b>
9	<b>8. XPS DOCUMENT FORMAT</b> .....	<b>15</b>
10	<b>8.1 How This Specification Is Organized</b> .....	<b>16</b>
11	<b>8.2 Package</b> .....	<b>18</b>
12	<b>9. PARTS AND RELATIONSHIPS</b> .....	<b>19</b>
13	<b>9.1 Fixed Payload</b> .....	<b>19</b>
14	9.1.1 Fixed Payload Relationships .....	22
15	9.1.2 FixedDocumentSequence Part .....	23
16	9.1.3 FixedDocument Part .....	24
17	9.1.4 FixedPage Part .....	24
18	9.1.5 Image Parts .....	24
19	9.1.6 Thumbnail Parts .....	30
20	9.1.7 Font Parts .....	31
21	9.1.8 Remote Resource Dictionary Parts .....	36
22	9.1.9 PrintTicket Parts .....	36
23	9.1.10 SignatureDefinitions Part .....	38
24	9.1.11 DocumentStructure Part .....	38
25	9.1.12 StoryFragments Part .....	39
26	<b>9.2 Part Naming Recommendations</b> .....	<b>40</b>
27	<b>9.3 XPS Document Markup</b> .....	<b>42</b>
28	9.3.1 Support for Versioning and Extensibility .....	43
29	9.3.2 XML Usage .....	43
30	9.3.3 Markup Model .....	44
31	9.3.4 Whitespace .....	47
32	9.3.5 Language .....	47
33	<b>10. DOCUMENTS</b> .....	<b>49</b>
34	<b>10.1 &lt;FixedDocumentSequence&gt; Element</b> .....	<b>49</b>
35	10.1.1 <DocumentReference> .....	49
36	<b>10.2 &lt;FixedDocument&gt; Element</b> .....	<b>50</b>
37	10.2.1 <PageContent> Element .....	50
38	10.2.2 <PageContent.LinkTargets> Element .....	51
39	10.2.3 <LinkTarget> Element .....	52
40	<b>10.3 &lt;FixedPage&gt; Element</b> .....	<b>53</b>
41	10.3.1 BleedBox Attribute .....	54
42	10.3.2 ContentBox Attribute .....	55
43	10.3.3 Page Size Terminology .....	55
44	10.3.4 Media Orientation and Scaling .....	56
45	<b>10.4 &lt;Canvas&gt; Element</b> .....	<b>61</b>
46	<b>10.5 &lt;Path&gt; Element</b> .....	<b>65</b>

1	<b>10.6</b>	<b>&lt;Glyphs&gt; Element.....</b>	<b>65</b>
2	<b>11. GRAPHICS.....</b>		<b>67</b>
3	<b>11.1</b>	<b>&lt;Path&gt; Element .....</b>	<b>68</b>
4	11.1.1	<Path.Data> Element .....	74
5	11.1.2	<Path.Fill> Element .....	75
6	11.1.3	<Path.Stroke> Element .....	76
7	<b>11.2</b>	<b>Geometries and Figures.....</b>	<b>77</b>
8	11.2.1	Geometries.....	78
9	11.2.2	Figures .....	81
10	11.2.3	Abbreviated Geometry Syntax .....	90
11	<b>12. TEXT .....</b>		<b>97</b>
12	<b>12.1</b>	<b>&lt;Glyphs&gt; Element.....</b>	<b>98</b>
13	12.1.1	Glyph Metrics.....	104
14	12.1.2	Mapping Code Units to Glyphs .....	105
15	12.1.3	Indices Attribute .....	109
16	12.1.4	UnicodeString Attribute .....	111
17	12.1.5	StyleSimulations Attribute.....	112
18	12.1.6	IsSideways Attribute .....	112
19	12.1.7	DeviceFontName Attribute .....	118
20	12.1.8	xml:lang Attribute .....	119
21	12.1.9	CaretStops Attribute .....	119
22	12.1.10	Optimizing Glyph Markup .....	120
23	12.1.11	Glyph Markup Examples.....	121
24	<b>12.2</b>	<b>&lt;Glyphs.Fill&gt; Element .....</b>	<b>124</b>
25	<b>13. BRUSHES .....</b>		<b>125</b>
26	<b>13.1</b>	<b>&lt;SolidColorBrush&gt; Element.....</b>	<b>126</b>
27	<b>13.2</b>	<b>&lt;ImageBrush&gt; Element .....</b>	<b>127</b>
28	<b>13.3</b>	<b>&lt;VisualBrush&gt; Element.....</b>	<b>131</b>
29	13.3.1	<VisualBrush.Visual> Element .....	133
30	<b>13.4</b>	<b>Common Attributes for Tiling Brushes .....</b>	<b>136</b>
31	13.4.1	Viewbox, Viewport, ViewboxUnits, and ViewportUnits Attributes .....	136
32	13.4.2	TileMode Attribute .....	141
33	<b>13.5</b>	<b>&lt;LinearGradientBrush&gt; Element .....</b>	<b>151</b>
34	13.5.1	SpreadMethod Attribute.....	154
35	13.5.2	<LinearGradientBrush.GradientStops> Element .....	157
36	<b>13.6</b>	<b>&lt;RadialGradientBrush&gt; Element .....</b>	<b>158</b>
37	13.6.1	SpreadMethod Attribute.....	162
38	13.6.2	<RadialGradientBrush.GradientStops> Element .....	165
39	<b>13.7</b>	<b>&lt;GradientStop&gt; Element.....</b>	<b>166</b>
40	<b>13.8</b>	<b>Using a Brush as an Opacity Mask .....</b>	<b>167</b>
41	<b>14. COMMON PROPERTIES.....</b>		<b>171</b>
42	<b>14.1</b>	<b>Opacity .....</b>	<b>172</b>
43	<b>14.2</b>	<b>Resources and Resource References .....</b>	<b>172</b>
44	14.2.1	<FixedPage.Resources> Element .....	173
45	14.2.2	<Canvas.Resources> Element .....	175
46	14.2.3	<ResourceDictionary> Element.....	176
47	14.2.4	Resource References .....	180
48	14.2.5	Scoping Rules for Resolving Resource References .....	180
49	14.2.6	Support for Markup Compatibility .....	181
50	<b>14.3</b>	<b>Clipping.....</b>	<b>182</b>
51	14.3.1	<Canvas.Clip> Element .....	182

1	14.3.2	<Path.Clip> Element.....	183
2	14.3.3	<Glyphs.Clip> Element.....	184
3	<b>14.4</b>	<b>Positioning Content.....</b>	<b>185</b>
4	14.4.1	<MatrixTransform> Element.....	185
5	14.4.2	<Canvas.RenderTransform> Element.....	189
6	14.4.3	<Path.RenderTransform> Element.....	190
7	14.4.4	<Glyphs.RenderTransform> Element.....	191
8	14.4.5	<PathGeometry.Transform> Element.....	192
9	14.4.6	<ImageBrush.Transform> Element.....	193
10	14.4.7	<VisualBrush.Transform> Element.....	194
11	14.4.8	<LinearGradientBrush.Transform> Element.....	197
12	14.4.9	<RadialGradientBrush.Transform> Element.....	198
13	<b>14.5</b>	<b>OpacityMask.....</b>	<b>200</b>
14	14.5.1	<Canvas.OpacityMask> Element.....	200
15	14.5.2	<Path.OpacityMask> Element.....	201
16	14.5.3	<Glyphs.OpacityMask> Element.....	202
17	<b>15. COLOR.....</b>		<b>205</b>
18	<b>15.1</b>	<b>Color Support.....</b>	<b>205</b>
19	15.1.1	sRGB Color Space.....	205
20	15.1.2	scRGB Color Space.....	206
21	15.1.3	Gray Color Space.....	206
22	15.1.4	CMYK Color Space.....	206
23	15.1.5	N-Channel Color Spaces.....	206
24	15.1.6	Named Color for Spot Colors and N-tone Images.....	206
25	15.1.7	Device Color Spaces.....	206
26	15.1.8	ICC Profiles.....	206
27	15.1.9	WcsProfilesTag.....	207
28	15.1.10	WCS Color Profiles.....	208
29	15.1.11	Vector Color Syntax.....	208
30	15.1.12	sRGB Color Syntax.....	209
31	15.1.13	scRGB Color Syntax.....	209
32	15.1.14	CMYK Color Syntax.....	210
33	15.1.15	N-Channel Color Syntax.....	210
34	15.1.16	Named Color Syntax.....	212
35	<b>15.2</b>	<b>Rich Colors in Raster Images.....</b>	<b>213</b>
36	15.2.1	sRGB Raster Images.....	213
37	15.2.2	scRGB Raster Images.....	214
38	15.2.3	Gray Raster Images.....	214
39	15.2.4	CMYK Raster Images.....	215
40	15.2.5	N-channel Raster Images.....	215
41	15.2.6	Named Color Raster Images.....	216
42	15.2.7	Device Color Raster Images.....	216
43	15.2.8	Images and Color Profile Association.....	216
44	15.2.9	Color Space Pixel Formats for Raster Images.....	216
45	<b>15.3</b>	<b>Color Separation.....</b>	<b>217</b>
46	<b>15.4</b>	<b>Alpha and Gradient Blending with Rich Colors.....</b>	<b>217</b>
47	<b>15.5</b>	<b>PrintTicket Color Settings.....</b>	<b>219</b>
48	<b>16. DOCUMENT STRUCTURE AND INTERACTIVITY.....</b>		<b>223</b>
49	<b>16.1</b>	<b>Document Structure Markup.....</b>	<b>223</b>
50	16.1.1	DocumentStructure Part.....	224
51	16.1.2	StoryFragments Part.....	230
52	<b>16.2</b>	<b>Hyperlinks.....</b>	<b>243</b>
53	16.2.1	Hyperlink Activation.....	243
54	16.2.2	Hyperlink Addressing.....	244
55	16.2.3	Name Attribute.....	245
56	16.2.4	FixedPage.NavigateUri Attribute.....	245
57	<b>16.3</b>	<b>Selection.....</b>	<b>246</b>

1	<b>16.4</b>	<b>Accessibility</b> .....	<b>246</b>
2	16.4.1	Reading Order.....	246
3	16.4.2	Screen Reader Applications .....	247
4	16.4.3	Text Alternatives for Graphics and Images.....	247
5	<b>17</b>	<b>XPS DOCUMENT PACKAGE FEATURES</b> .....	<b>249</b>
6	<b>17.1</b>	<b>Interleaving Optimizations</b> .....	<b>249</b>
7	17.1.1	Empty PrintTicket .....	251
8	17.1.2	Optimizing Interleaving Order.....	251
9	17.1.3	Consuming Interleaved Packages .....	255
10	17.1.4	Consumers with Resource Constraints .....	255
11	17.1.5	Interleaving Optimizations and Digital Signatures .....	257
12	<b>17.2</b>	<b>Digital Signatures</b> .....	<b>258</b>
13	17.2.1	Signature Policy .....	258
14	17.2.2	Signature Definitions.....	261
15	<b>17.3</b>	<b>Core Properties</b> .....	<b>265</b>
16	<b>18</b>	<b>RENDERING RULES</b> .....	<b>267</b>
17	<b>18.1</b>	<b>Coordinate System and Rendering Placement</b> .....	<b>267</b>
18	18.1.1	Page Dimensions .....	267
19	18.1.2	Rounding of Coordinates.....	267
20	18.1.3	Transforms.....	268
21	18.1.4	Pixel Center Location, Pixel Placement, and Pixel Inclusion .....	269
22	18.1.5	Maximum Placement Error .....	269
23	18.1.6	Pixel Placement for Glyphs .....	269
24	18.1.7	Abutment of Shapes.....	269
25	18.1.8	Clipping Behavior .....	270
26	<b>18.2</b>	<b>Implementation Limits</b> .....	<b>270</b>
27	<b>18.3</b>	<b>Gradient Computations</b> .....	<b>272</b>
28	18.3.1	All Gradients.....	272
29	18.3.2	Linear Gradients.....	274
30	18.3.3	Radial Gradients.....	276
31	<b>18.4</b>	<b>Opacity Computations</b> .....	<b>279</b>
32	18.4.1	Pre-Multiplied Alpha and Superluminous Colors.....	281
33	<b>18.5</b>	<b>Composition Rules</b> .....	<b>282</b>
34	18.5.1	Optimization Guidelines .....	283
35	18.5.2	Composition Examples.....	284
36	<b>18.6</b>	<b>Stroke Rendering</b> .....	<b>288</b>
37	18.6.1	Stroke Edge Parallelization .....	288
38	18.6.2	Phase Control .....	288
39	18.6.3	Symmetry of Stroke Drawing Algorithms .....	288
40	18.6.4	Rules for Dash Cap Rendering.....	289
41	18.6.5	Rules for Line Cap Rendering .....	291
42	18.6.6	Line Caps for Dashed Strokes .....	292
43	18.6.7	Rules for Line Join Rendering.....	293
44	18.6.8	Rules for Degenerate Line and Curve Segments.....	297
45	18.6.9	Stroking and Fill Rule .....	298
46	18.6.10	Mixing Stroked and Non-Stroked Segments.....	298
47	18.6.11	Stroke Behavior with Multiple Path Figures .....	298
48	18.6.12	Consistent Nominal Stroke Width .....	298
49	<b>18.7</b>	<b>Brushes and Images</b> .....	<b>299</b>
50	18.7.1	Small Tiles .....	299
51	18.7.2	Image Scaling.....	299
52	18.7.3	Tile Placement .....	299
53	18.7.4	Tiling Transparent Visual Brushes and Image Brushes.....	300

1 **19. ELEMENTS** ..... 301

2 **A. SIGNATURE DEFINITIONS SCHEMA**..... 359

3 **B. XPS DOCUMENT SCHEMA** ..... 361

4 **C. RESOURCE DICTIONARY KEY SCHEMA** ..... 383

5 **D. DOCUMENT STRUCTURE SCHEMA** ..... 385

6 **E. DISCARD CONTROL SCHEMA** ..... 391

7 **F. ABBREVIATED GEOMETRY SYNTAX ALGORITHM** ..... 393

8 **G. sRGB GAMUT BOUNDARY DEFINITION** ..... 399

9 **H. STANDARD NAMESPACES AND CONTENT TYPES**..... 405

10 **H.1 XML Namespace URIs** ..... 405

11 **H.2 Content Types** ..... 406

12 **H.3 Relationship Types** ..... 407

13 **I. CONFORMANCE REQUIREMENTS** ..... 409

14 **I.1 XPS Document Format** ..... 409

15 **I.2 Parts and Relationships** ..... 410

16 **I.3 Documents** ..... 421

17 **I.4 Graphics**..... 423

18 **I.5 Text** ..... 424

19 **I.6 Brushes** ..... 427

20 **I.7 Common Properties** ..... 428

21 **I.8 Color** ..... 429

22 **I.9 Document Structure and Interactivity**..... 434

23 **I.10 XPS Document Package Features** ..... 439

24 **I.11 Rendering Rules** ..... 444

25 **I.12 Additional Conformance Requirements** ..... 450

26 **J. BIBLIOGRAPHY**..... 453

27 **K. INDEX**..... 457

28





1 **List of Figures**

2 Figure 8–1. Package-based XPS Document format..... 18

3 Figure 10–1. Page regions ..... 56

4 Figure 10–2. Matching PrintTicket and fixed page size and orientation ..... 57

5 Figure 10–3. Matching PrintTicket and fixed page size with differing orientation ..... 57

6 Figure 10–4. Matching PrintTicket and fixed page orientation with differing size ..... 58

7 Figure 11–1. Fill using EvenOdd algorithm..... 80

8 Figure 11–2. Fill using NonZero algorithm ..... 80

9 Figure 11–3. Arc choice A..... 84

10 Figure 11–4. Arc choice B..... 84

11 Figure 11–5. Arc choice C..... 84

12 Figure 11–6. Arc choice D ..... 84

13 Figure 12–1. Glyph metrics ..... 104

14 Figure 12–2. Upright (usually horizontal) glyph metrics..... 104

15 Figure 12–3. Sideways (usually vertical) glyph metrics ..... 105

16 Figure 17–1. A sample signature spot ..... 264

17 Figure 18–2. Extreme curvatures and dash rendering ..... 288

18 Figure 18–3. Flat dash caps..... 289

19 Figure 18–4. Square dash caps ..... 290

20 Figure 18–5. Round dash caps ..... 290

21 Figure 18–6. Triangular dash caps..... 291

22 Figure 18–7. Overlapping dash segments..... 291

23 Figure 18–8. Flat start line cap, flat end line cap ..... 292

24 Figure 18–9. Square start line cap, square end line cap ..... 292

25 Figure 18–10. Triangular start line cap, triangular end line cap ..... 292

26 Figure 18–11. Round start line cap, round end line cap ..... 292

27 Figure 18–12. Stroke start or end point within a dash..... 292

28 Figure 18–13. Stroke start or end point within a gap ..... 292

29 Figure 18–14. Round line join with right angle ..... 293

30 Figure 18–15. Round line join with acute angle ..... 293

1 Figure 18-16. Round line join with obtuse angle ..... 294

2 Figure 18-17. Beveled line join with right angle ..... 294

3 Figure 18-18. Beveled line join with acute angle ..... 294

4 Figure 18-19. Beveled line join with obtuse angle ..... 295

5 Figure 18-20. Mitered line join with right angle and miter limit of 1.0 ..... 295

6 Figure 18-21. Mitered line join with acute angle and miter limit of 1.0 ..... 296

7 Figure 18-22. Mitered line join with obtuse angle and miter limit of 1.0 ..... 296

8 Figure 18-23. Mitered line join with right angle and miter limit of 2.0 ..... 296

9 Figure 18-24. Mitered line join with acute angle and miter limit of 2.0 ..... 297

10 Figure 18-25. Mitered line join with acute angle and miter limit of 10.0 ..... 297

11

1 **List of Tables**

2	Table 9–1. XPS Document parts .....	19
3	Table 9–2. Fixed payload relationships .....	22
4	Table 9–3. Supported JPEG APPn markers .....	25
5	Table 9–4. Support for ancillary PNG chunks .....	25
6	Table 9–5. Supported TIFF 6.0 tags .....	26
7	Table 9–6. Supported Windows Media Photo features .....	30
8	Table 9–7. Guidelines for OpenType font embedding .....	32
9	Table 9–8. Cmap table selection .....	35
10	Table 11–9. Arc segment definition .....	83
11	Table 11–10. Commands .....	91
12	Table 12–1. Glyph specifications .....	109
13	Table 12–2. Portions of the cluster specification .....	110
14	Table 12–3. IsSideways and BidiLevel effects on origin placement .....	114
15	Table 13–1. Brush types .....	125
16	Table 13–2. Common attributes for <ImageBrush> and <VisualBrush> elements	136
17	Table 14–1. Common property attributes .....	171
18	Table 14–2. Common property elements .....	172
19	Table 15–3. WcsProfilesTagType structure .....	207
20	Table 15–1. Syntax summary .....	209
21	Table 15–2. Color Space Pixel Format Defaults .....	216
22	Table 15–3. PrintTicket color settings .....	219
23	Table 16–1. StoryFragments part elements .....	230
24	Table 16–2. Unicode character categories .....	245
25	Table 17–1. JobDigitalSignatureProcessing PrintTicket settings .....	261
26	Table 18–1. Recommended minimum processing requirements .....	270
27	Table 18–2. Opacity computation symbols .....	279
28	Table H–1. Package-wide namespaces .....	405
29	Table H–2. XPS Document namespaces .....	405
30	Table H–3. Package-wide content types .....	406

1	Table H-4. XPS Document content types.....	406
2	Table H-5. Package-wide relationship types .....	407
3	Table H-6. XPS Document relationship types.....	407
4	Table I-1. XPS Document format MUST conformance requirements.....	409
5	Table I-2. Parts and Relationships MUST conformance requirements .....	410
6	Table I-3. Parts and Relationships SHOULD conformance requirements .....	415
7	Table I-4. Parts and Relationships OPTIONAL conformance requirements.....	419
8	Table I-5. Document MUST conformance requirements.....	421
9	Table I-6. Document SHOULD conformance requirements.....	422
10	Table I-7. Graphics MUST conformance requirements.....	423
11	Table I-8. Graphics SHOULD conformance requirements.....	423
12	Table I-9. Graphics OPTIONAL conformance requirements .....	423
13	Table I-10. Text MUST conformance requirements .....	424
14	Table I-11. Text SHOULD conformance requirements .....	426
15	Table I-12. Text OPTIONAL conformance requirements.....	426
16	Table I-13. Brushes MUST conformance requirements .....	427
17	Table I-14. Common properties MUST conformance requirements .....	428
18	Table I-15. Common properties OPTIONAL conformance requirements .....	429
19	Table I-16. Color MUST conformance requirements.....	429
20	Table I-17. Color SHOULD conformance requirements .....	432
21	Table I-18. Color OPTIONAL conformance requirements .....	433
22	Table I-19. Document structure MUST conformance requirements .....	434
23	Table I-20. Document structure SHOULD conformance requirements .....	435
24	Table I-21. Document structure OPTIONAL conformance requirements.....	438
25	Table I-22. XPS Document package feature MUST conformance requirements .....	439
26	Table I-23. XPS Document package feature SHOULD conformance requirements .....	441
27	Table I-24. XPS Document package feature OPTIONAL conformance requirements .....	443
28	.....	443
29	Table I-25. Rendering rules MUST conformance requirements .....	444
30	Table I-26. Rendering rules SHOULD conformance requirements .....	445
31	Table I-27. Rendering rules OPTIONAL conformance requirements.....	448

1	Table I-28. Additional MUST conformance requirements .....	450
2		



1 **List of Examples**

2 Example 9-1. A typical XPS Document..... 21

3 Example 9-2. XPS Document part naming..... 42

4 Example 9-3. Property attribute syntax ..... 45

5 Example 9-4. Property element syntax ..... 47

6 Example 10-1. <FixedDocumentSequence> usage..... 49

7 Example 10-2. <FixedDocument> usage ..... 50

8 Example 10-3. <PageContent> usage ..... 51

9 Example 10-4. <PageContent.LinkTargets> usage..... 52

10 Example 10-5. Fixed page markup..... 54

11 Example 10-6. Canvas composition..... 64

12 Example 11-1. <Path.Data> usage ..... 74

13 Example 11-2. <Path.Fill> usage..... 76

14 Example 11-3. <Path.Stroke> usage ..... 77

15 Example 11-4. <PathGeometry> usage..... 79

16 Example 11-5. <ArcSegment> usage..... 84

17 Example 11-6. <PolyBezierSegment> usage..... 86

18 Example 11-7. <PolyLineSegment> usage..... 87

19 Example 11-8. <PolyQuadraticBezierSegment> usage ..... 89

20 Example 11-9. Closed <PathFigure> usage..... 89

21 Example 11-10. A path described using abbreviated syntax ..... 94

22 Example 11-11. Smooth Bézier curve..... 94

23 Example 11-12. Relative commands and curves ..... 95

24 Example 12-1. One-to-one cluster map ..... 106

25 Example 12-2. Many-to-one cluster map ..... 106

26 Example 12-3. One-to-many cluster map ..... 107

27 Example 12-4. Many-to-many cluster map..... 108

28 Example 12-1. Using indices to specify advance width..... 110

29 Example 12-2. Using the Indices attribute to specify glyph replacement for a cluster

30 ..... 111

31 Example 12-3. Text with positive uOffset and vOffset Indices values..... 115

1	Example 12-4. Right-to-left text (odd BidiLevel) .....	115
2	Example 12-5. Sideways text (IsSideways set to true) .....	116
3	Example 12-6. Vertical text.....	116
4	Example 12-7. Japanese vertical text .....	117
5	Example 12-8. Using the CaretStops attribute to determine a valid caret stop	
6	position.....	119
7	Example 12-9. Basic italic font .....	121
8	Example 12-10. Italic font using StyleSimulations attribute.....	121
9	Example 12-11. Kerning .....	122
10	Example 12-12. Ligatures .....	122
11	Example 12-13. Cluster maps .....	123
12	Example 13-1. <SolidColorBrush> usage.....	127
13	Example 13-2. <ImageBrush> usage.....	130
14	Example 13-3. <VisualBrush.Visual> usage .....	134
15	Example 13-4. ViewboxUnits and ViewportUnits attribute usage .....	137
16	Example 13-5. Tiling brush base image and rendering.....	137
17	Example 13-6. Tiling brush Viewport adjustments.....	138
18	Example 13-7. Tiling brush viewbox adjustments.....	139
19	Example 13-8. Image brush with a Viewbox larger than the image .....	140
20	Example 13-9. Image brush with TileMode value of None .....	141
21	Example 13-10. Visual brush with TileMode value of None .....	142
22	Example 13-11. Image brush with a TileMode value of Tile .....	143
23	Example 13-12. Visual brush with a TileMode value of Tile .....	144
24	Example 13-13. Image brush with a TileMode value of FlipX .....	145
25	Example 13-14. Visual brush with a TileMode value of FlipX .....	146
26	Example 13-15. Image brush with a TileMode value of FlipY.....	147
27	Example 13-16. Visual Brush with a TileMode value of FlipY .....	148
28	Example 13-17. Image brush with a TileMode value of FlipXY.....	149
29	Example 13-18. Visual brush with a TileMode value of FlipXY .....	150
30	Example 13-19. <LinearGradientBrush> usage .....	153
31	Example 13-20. Linear gradient brush with a SpreadMethod value of Pad .....	154



1 Example 13–21. Linear gradient brush with a SpreadMethod value of Reflect..... 155

2 Example 13–22. Linear gradient brush with a SpreadMethod value of Repeat ..... 156

3 Example 13–23. A radial gradient brush..... 161

4 Example 13–24. RadialGradientBrush usage ..... 161

5 Example 13–25. Radial gradient brush with a SpreadMethod value of Pad ..... 162

6 Example 13–26. Radial gradient brush with a SpreadMethod value of Reflect..... 163

7 Example 13–27. Radial gradient brush with a SpreadMethod value of Repeat ..... 164

8 Example 13–28. Opacity mask with linear gradient..... 167

9 Example 13–29. Opacity mask with radial gradient..... 169

10 Example 14–1. <FixedPage.Resources> usage ..... 173

11 Example 14–2. <Canvas.Resources> usage ..... 175

12 Example 14–3. Resource dictionary markup ..... 177

13 Example 14–4. A remote resource dictionary and reference..... 178

14 Example 14–5. Using a resource reference to fill a brush..... 180

15 Example 14–6. Using scoping rules..... 181

16 Example 14–7. Canvas clip markup and rendering..... 182

17 Example 14–8. <Path.Clip> usage ..... 183

18 Example 14–9. <Glyphs.Clip> usage ..... 184

19 Example 14–10. Matrix scaling..... 186

20 Example 14–11. Matrix reversing the x axis..... 186

21 Example 14–12. Matrix reversing the y axis..... 186

22 Example 14–13. Matrix skewing ..... 186

23 Example 14–14. Matrix Rotating ..... 187

24 Example 14–15. Matrix positioning ..... 187

25 Example 14–16. <MatrixTransform> usage ..... 187

26 Example 14–17. Using abbreviated matrix transformation syntax..... 189

27 Example 14–18. <Canvas.RenderTransform> usage..... 189

28 Example 14–19. <Path.RenderTransform> usage..... 190

29 Example 14–20. <Glyphs.RenderTransform> usage ..... 191

30 Example 14–21. <PathGeometry.Transform> usage..... 192

31 Example 14–22. <ImageBrush.Transform> usage..... 193

1	Example 14–23. <VisualBrush.Transform> usage .....	194
2	Example 14–24. <VisualBrush.Transform> usage with tiling behavior.....	196
3	Example 14–25. <LinearGradientBrush.Transform> usage .....	197
4	Example 14–26. <RadialGradientBrush.Transform> usage .....	198
5	Example 14–27. <Canvas.OpacityMask> usage .....	200
6	Example 14–28. <Path.OpacityMask> usage .....	201
7	Example 14–29. <Glyphs.OpacityMask> usage .....	203
8	Example 16–1. Document structure markup .....	224
9	Example 16–2. Document outline markup .....	227
10	Example 16–3. Simple multi-story document.....	230
11	Example 16–4. Story flowing back and forth across a page boundary .....	230
12	Example 16–5. Content structure spanning pages .....	232
13	Example 16–6. StoryFragments part markup.....	236
14	Example 16–7. Story fragments markup using a fragment name.....	237
15	Example 16–8. A relative, internal, named-address hyperlink .....	244
16	Example 16–9. A relative internal page address hyperlink .....	244
17	Example 17–1. Optimized interleaving for a single-threaded parsing architecture.	251
18	Example 17–2. Optimized interleaving for a multi-threaded parsing architecture..	254
19	Example 17–3. A DiscardControl part.....	256
20	Example 17–4. A SignatureDefinitions part.....	262
21	Example 18–1. Path opacity behavior for overlapping path figures .....	284
22	Example 18–2. Opacity behavior of path stroke intersections.....	285
23	Example 18–3. Opacity behavior of paths with stroked edges .....	285
24		

## 1 1. Scope

2 This specification defines [XPS](#), the XML Paper Specification. XPS describes the set of  
3 conventions for the use of XML and other widely available technologies to describe  
4 the content and appearance of paginated documents. It is written for developers who  
5 are building systems that process XPS content.

6 A primary goal is to ensure the interoperability of independently created software  
7 and hardware systems that produce or consume XPS content. This specification  
8 defines the formal requirements that producers and consumers must satisfy in order  
9 to achieve interoperability.

10 This specification describes a paginated-document format called the [XPS Document](#).  
11 The format requirements are an extension of the packaging requirements described  
12 in the Open Packaging Conventions (OPC) specification. That specification describes  
13 packaging and physical format conventions for the use of XML, Unicode, ZIP, and  
14 other technologies and specifications, to organize the content and resources that  
15 make up any document. They are an integral part of the XPS specification, and are  
16 included by reference.

17 Many XML-based building blocks within XPS make use of the conventions described  
18 in the Markup Compatibility specification that is relied upon by the OPC specification  
19 to facilitate future enhancement and extension of XPS markup. As such, that Markup  
20 Compatibility specification is included by reference.



## 1 2. Conformance

### 2 Language Notes

3 In this specification, the words that are used to define the significance of each  
4 requirement are written in uppercase. These words are used in accordance with their  
5 definitions in RFC 2119, and their respective meanings are reproduced below:

- 6 • MUST: This word, or the adjective "REQUIRED", means that the item is an  
7 absolute requirement of the specification.
- 8 • SHOULD: This word, or the adjective "RECOMMENDED", means that there may  
9 exist valid reasons in particular circumstances to ignore this item, but the full  
10 implications should be understood and the case carefully weighed before  
11 choosing a different course.
- 12 • MAY: This word, or the adjective "OPTIONAL", means that this item is truly  
13 optional. [*Example*: One implementation may choose to include the item  
14 because a particular marketplace or scenario requires it or because it  
15 enhances the product. Another implementation may omit the same item. *end*  
16 *example*]

17 The words MUST NOT and SHOULD NOT are the negative forms of MUST and  
18 SHOULD, respectively. There is no negative form of MAY.

### 19 Software Conformance

20 Most requirements are expressed as format or package requirements rather than  
21 implementation requirements.

22 In order for consumers to be considered conformant, they must observe the  
23 following rules:

- 24 • They MUST NOT report errors when processing conforming instances of the  
25 documented formats except when forced to do so by resource exhaustion.
- 26 • They SHOULD report errors when processing non-conforming instances of the  
27 documented formats when doing so does not pose an undue processing or  
28 performance burden.

29 In order for producers to be considered conformant, they must observe the following  
30 rules:

- 31 • They MUST NOT generate any new, non-conforming instances of a  
32 documented format.
- 33 • They MUST NOT introduce any non-conformance when modifying an instance  
34 of a documented format.

35 Editing applications are subject to all of the **above rules**.

**Comment [rcj1]:** rules for consumers and producers.

36 Conformance requirements are documented inline in this specification, and each  
37 requirement is denoted by a letter (M – MUST; S – SHOULD; O – OPTIONAL) and a  
38 rule number, all enclosed in brackets ([...]). [*Example*: [M1.2] is a MUST  
39 requirement, [S2.4] is a SHOULD requirement, and [O3.9] is a MAY requirement.

- 1 *end example*] These rules are collected in I, "Conformance Requirements" for
- 2 convenient reference.

### 1 **3. Normative References**

2 The following normative documents contain provisions, which, through reference in  
3 this text, constitute provisions of this specification. For dated references, subsequent  
4 amendments to, or revisions of, any of these publications do not apply. However,  
5 parties to agreements based on this specification are encouraged to investigate the  
6 possibility of applying the most recent editions of the normative documents indicated  
7 below. For undated references, the latest edition of the normative document referred  
8 to applies. Members of ISO and IEC maintain registers of currently valid International  
9 Standards.

10

11 ECMA-376, "Office Open XML File Formats" (December 2006), Part 2, "Open  
12 Packaging Conventions", which is commonly referred to as OPC.

13 ECMA-376, "Office Open XML File Formats" (December 2006), Part 5, "Markup  
14 Compatibility and Extensibility".

15 ISO/IEC 2382.1:1993, Information technology — Vocabulary — Part 1: Fundamental  
16 terms.

17 ISO/IEC 10646:2003 (all parts), Information technology — Universal Multiple-Octet  
18 Coded Character Set (UCS).





## 1 4. Definitions

2 For the purposes of this specification, the following definitions apply. Terms explicitly  
3 defined in this specification are not to be presumed to refer implicitly to similar terms  
4 defined elsewhere.

5 **alpha blending** — Transparently blending two elements when rendering.

6 **broken digital signature** — A digital signature that conforms to the XPS Document  
7 signing rules but does not meet the digital signature validity requirements due to  
8 incorrect hash calculation or similar problems.

9 **Compliant digital signature** — A digital signature that conforms to the signing  
10 rules described in the XPS Document signing policy, regardless of signature validity.

Comment [rcj2]: Never used

11 **consumer** — A piece of software or a device that reads XPS packages.

12 **content structure** — The set of markup elements that allow expression of well-  
13 understood semantic blocks such as paragraphs, tables, lists, and figures.

14 **content type** — Describes the type of content stored in a part. Content types define  
15 a media type, a subtype, and an optional set of parameters, as defined in RFC 2045.

16 **contour intersection point** — The intersection of the flat line ending a dash and  
17 the contour of the shape.

18 **device** — A piece of hardware, such as a printer or scanner, that performs a single  
19 function or a set of functions.

20 **document content** — A document structural concept that identifies each block of  
21 individually readable content in an XPS Document.

22 **document outline** — A document structural concept that contains a structured  
23 index of the content in an XPS Document, much like a table of contents.

24 **driver** — A producer that has specific knowledge of the consumer of the XPS  
25 Document.

26 **effective coordinate space** — The default coordinate space (X,Y in the upper-left  
27 corner, units of 1/96") as modified by any RenderTransform or Transform attributes  
28 of the current element and any ancestor elements.

29 **fixed payload** — A payload that is rooted with a FixedDocumentSequence part.

30 **fixed payload root** — The root of a fixed payload is the FixedDocumentSequence  
31 part.

32 **FixedDocument part** — A common, easily indexed root for all pages within the  
33 document.

34 **FixedDocumentSequence part** — The part that assembles a set of FixedDocument  
35 parts within the fixed payload.

- 1 **FixedPage part** — The part that contains all of the visual elements to be rendered  
2 on a page.
- 3 **Incompliant digital signature** — A digital signature that does not conform to the  
4 XPS Document signing rules.
- 5 **interleaved ordering** — The layout style of a physical package where parts are  
6 broken into pieces and “mixed-in” with pieces from other parts. When delivered,  
7 interleaved packages help improve the performance of the consumer processing the  
8 package.
- 9 **named color** — An industry-defined color specification that identifies a particular  
10 color in a well-defined color schema, usually for purposes of printing.
- 11 **named element** — An element in the document structure markup that refers to an  
12 element in the fixed-page markup with a specified name.
- 13 **package** — A logical entity that holds a collection of parts.
- 14 **package model** — Defines a package abstraction that holds a collection of parts.
- 15 **package relationship** — A relationship whose target is a part and whose source is  
16 the package as a whole. Package relationships are found in the package relationships  
17 part named “/\_rels/.rels”.
- 18 **part** — A stream of bytes with a MIME content type and associated common  
19 properties. Typically corresponds to a file (as on a file system), a stream (as in a  
20 compound file), or a resource (as in an HTTP URI).
- 21 **part name** — A part name is used to refer to a part in the context of a package,  
22 typically as part of a URI. The part name is, by definition, the path component of a  
23 pack URI.
- 24 **payload** — A complete collection of interdependent parts and relationships within a  
25 package.
- 26 **physical imageable size** — Represents the area of a page that is printable by a  
27 specific device.
- 28 **physical media size** — Represents the physical media on which the content will be  
29 printed.
- 30 **physical model** — Defines the mapping between the components of the package  
31 model to the features of a particular physical format based on the ZIP specification.
- 32 **piece** — A portion of a part. Pieces of different parts may be interleaved together.  
33 The individual pieces are named using a unique mapping from the part name. The  
34 piece name grammar does not conform to the part name grammar. Pieces are not  
35 addressable in the package model.
- 36 **primary fixed payload root** — The fixed payload root that is referenced by the XPS  
37 package StartPart relationship.
- 38 **PrintTicket part** — A PrintTicket part provides the settings used when a package is  
39 printed. PrintTicket parts may be attached to the entire package, or at lower levels in  
40 the structure, such as individual pages.

- 1 **producer** — A piece of software or a device that writes XPS packages.
- 2 **producer bleed size** — Represents the overflow (or “bleed”) box used by the  
3 producer for registration and layout.
- 4 **producer content size** — Represents the content bounding box specified by the  
5 producer.
- 6 **producer media size** — Represents the physical media on which the content will be  
7 printed.
- 8 **property** — A characteristic of a markup element, referred to as an attribute of the  
9 element.
- 10 **property attribute** — XPS Document property values can be expressed as either  
11 property attributes or property elements.
- 12 **property element** — An XPS Document property value can be expressed as either a  
13 property attribute or a property element.
- 14 **property value** — The value of a property, expressed as an XML attribute, an XML  
15 child element, or an entry in the resource dictionary.
- 16 **questionable digital signature** — A digital signature that conforms to the XPS  
17 Document signing rules but has a problem during validation of the signature such as  
18 the inability to contact the certificate authority to validate its authenticity or the  
19 markup contains markup compatibility elements and attributes that may change the  
20 representation of the signed content.
- 21 **relationships** — A relationship represents the kind of connection between a source  
22 part and a target part in a package. Relationships make the connections between  
23 parts directly discoverable without looking at the content in the parts, and without  
24 altering the parts themselves. See also, package relationship.
- 25 **relationships part** — A part containing an XML representation of relationships.
- 26 **remote resource dictionary** — A part containing a resource dictionary.
- 27 **required part** — A part, such as an image or font, that is referenced from other  
28 parts, and is required for valid processing of the referencing part.
- 29 **resource definition** — A shareable property value, with a name, defined within a  
30 resource dictionary. Any property value defined by fixed page markup may be held in  
31 a resource dictionary. Each resource definition has a key that is unique within the  
32 scope of the resource dictionary.
- 33 **resource dictionary** — A resource dictionary holds resources. Each resource in a  
34 resource dictionary carries a name. The resource’s name can be used to reference  
35 the resource from a property’s XML attribute.
- 36 **resource reference** — An attribute whose value refers to an entry in a resource  
37 dictionary. Resource references appear in the format “{StaticResource RscName}”  
38 where RscName corresponds to a matching entry in the resource dictionary with an  
39 x:Key attribute value.
- 40 **signature definition** — The means by which XPS Document authors provide co-  
41 signature requirements and workflow-specific signature information.

- 1 **signature spot** — A visual element that indicates that a digital signature has been  
2 applied or requested.
- 3 **signing rules** — The set of rules that define whether a particular digital signature is  
4 compliant with the XPS Document **signature** policy.
- 5 **simple ordering** — Simple ordering the parts in the package are laid out with a  
6 defined ordering. When such a package is delivered in a purely linear fashion,  
7 starting with the first byte in the package through to the last that, all of the bytes for  
8 the first part arrive first, then all of the bytes for the second part, and so on.
- 9 **stories** — Meaningful semantic blocks that provide structure to markup elements  
10 that appear in a FixedPage part.
- 11 **story** — A block of individually readable content in an XPS Document.
- 12 **story fragment** — A portion of a story that appears within the scope of a single  
13 fixed page.
- 14 **stream** — A linearly ordered sequence of bytes.
- 15 **thumbnail** — Images that help end-users identify parts of a package or a package  
16 as a whole.
- 17 **valid digital signature** — A digital signature that conforms to the XPS Document  
18 signing rules and is not a broken digital signature or questionable digital signature.
- 19 **XPS Document** — A package that contains a discoverable fixed payload and is a  
20 format for storing paginated documents defined by the XPS specification.
- 21 **XPS Document StartPart relationship** — The specific relationship type that  
22 identifies the root of a fixed payload within an XPS Document.
- 23 **ZIP Archive** — A physical ZIP file that is displayed by the file system. A ZIP archive  
24 contains **ZIP items**.

## 1 5. Notational Conventions

### 2 Document Conventions

3 Except where otherwise noted, syntax descriptions are expressed in the ABNF format  
4 as defined in RFC 4234.

5 Definition terms are formatted like `this`.



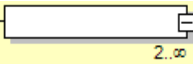
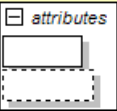


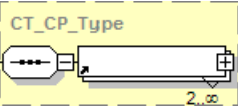
6 Syntax descriptions and code are formatted in `monospace` type.

7 Replaceable items are formatted in `monospace cursive` type.

### 8 Diagram Notes

9 In some cases, markup semantics are described using diagrams. The diagrams place  
10 the parent element on the left, with attributes and child elements to the right. The  
11 symbols are described below.

12

Symbol	Description
	Required element. This box represents an element that <b>MUST</b> appear exactly once in markup when the parent element is included. The "+" and "-" symbols on the right of these boxes have no semantic meaning.
	Optional element. This box represents an element that can appear zero or one times in markup when the parent element is included.
	Range indicator. These numbers indicate that the designated element or choice of elements can appear in markup any number of times within the range specified.
	Attribute group. This box indicates that the enclosed boxes are each attributes of the parent element. Solid-border boxes are required attributes; dashed-border boxes are optional attributes.
	Sequence symbol. The element boxes connected to this symbol can appear in markup in the illustrated sequence only, from top to bottom.
	Choice symbol. Only one of the element boxes connected to this symbol can appear in markup.
	Type indicator. The elements within the dashed box are of the complex type indicated.



1 **6. Acronyms and Abbreviations**

2 **This clause is informative**

3 The following acronyms and abbreviations are used throughout this specification:

4 IEC — the International Electrotechnical Commission

5 ISO — the International Organization for Standardization

6 W3C — World Wide Web Consortium

7 **End of informative text**





## 1 **7. General Description**

2 This specification is intended for use by implementers, academics, and application  
3 programmers. As such, it contains explanatory material that, strictly speaking, is not  
4 necessary in a formal specification.

5 This specification is divided into the following subdivisions:

- 6 1. Front matter (clauses 1–7).
- 7 2. XPS Documents (clauses 8–18), which presents the details of the primarily  
8 XML-based XPS Document format. These clauses describe the XML markup  
9 that defines the composition of documents and the appearance of each page.  
10 They also include rendering rules that enable devices and applications to  
11 display and print XPS Documents with full fidelity in a wide range of  
12 environments and scenarios.
- 13 3. XPS Document Markup Reference (clause 19), which presents a consolidated  
14 reference of XPS Document markup elements and their attributes.
- 15 4. Annexes (clauses A–J), which contain additional technical details and schemas,  
16 as well as convenient reference information.

17 Examples are provided to illustrate possible forms of the constructions described.  
18 References are used to refer to related clauses. Notes are provided to give advice or  
19 guidance to implementers or programmers. Annexes provide additional information  
20 or summarize the information contained elsewhere in this specification.

21 Clauses 1–5 and 7–19, and annexes A–H and J, form a normative part of this  
22 specification; and the clause 6, annexe I, examples, notes, and the index, are  
23 informative.

24 Except for whole clauses or annexes that are identified as being informative,  
25 informative text that is contained within normative text is indicated in the following  
26 ways:

- 27 1. Examples within narrative are indicated as follows: [*Example: ... end example*]
- 28 2. Examples of XML are indicated as follows: *Example m.n: caption ... end*  
29 *example*]
- 30 3. [*Note: ... end note*]

## 31 **8. XPS Document Format**

32 This specification describes how the XPS Document format is organized internally and  
33 rendered externally. It is built upon the principles described in the OPC specification.  
34 XPS Documents MUST observe all conformance requirements and recommendations  
35 of that specification, except where indicated otherwise [M1.1]. The information  
36 presented here is intended both for producers, which emit content in the XPS  
37 Document format, and consumers, which access and render or process the contents  
38 of an XPS Document.

1 The XPS Document format represents a set of related pages with a fixed layout,  
 2 which are organized as one or more *documents*, in the traditional meaning of the  
 3 word. A file that implements this format includes everything necessary to render fully  
 4 those documents on a display device or physical medium (such as paper). This  
 5 includes all resources such as fonts and images that might be required to render  
 6 individual page markings.

7 In addition, the format includes optional components that build on the minimal set of  
 8 components required to render a set of pages. This includes the ability to specify  
 9 print job control instructions, to organize the minimal page markings into larger  
 10 semantic blocks such as paragraphs, and to rearrange physically the contents of the  
 11 format for easy consumption in a streaming manner, among others.

12 Finally, the XPS Document format implements the common package features  
 13 specified by the Open Packaging Conventions specification that support digital  
 14 signatures and core properties.

---

## 15 8.1 How This Specification Is Organized

16

17 **This subclause is informative**

18

Clause	Description
Parts and Relationships (§9)	This clause describes how XPS Documents use the packaging model (as described in the OPC specification) to organize data. All part and relationship types are described in detail, including how they are used and what they can contain.  This clause also describes the XPS Document markup model, in particular, its parts, and how the XML markup relates to the packaging conventions and recommendations it builds on.
Documents (§10)	The fundamental building blocks of the XPS Document format are described here. This clause describes how pages are composed into larger documents and how documents are composed into document sequences. These components are represented in markup.
Graphics (§11)	This is the first of several clauses that describe page markings, in particular, vector graphics. The concepts of paths, geometries, and figures are introduced. Vector graphics are represented in page-layout XML markup.
Text (§12)	This clause describes how to include text markings in page-layout markup. It describes how to reference a font and extract information from a font to render the page.
Brushes (§13)	Both vector graphics and text are rendered by applying any of the brushes described in this clause. This includes brushes that are created from solid colors, gradients, images, or other page-layout markup.

Clause	Description
Common Properties (§14)	Several page-layout markup elements share a common set of properties. These properties can be expressed either as XML attributes or as XML child or descendant elements. This clause describes these common properties.
Color (§15)	XPS Documents support a wide range of color options and color spaces, both for vector and raster images. This clause describes the combinations of image formats and color markup that may be used. A number of color-related topics are discussed, including color separation, color profiles, and color blending.
Document Structure and Interactivity (§16)	This clause describes the components of the XPS Document format that support assigning larger semantic meaning to individual page markings in order. [ <i>Example</i> : Such markings might be tables or paragraphs. <i>end example</i> ] It also provides a mechanism to describe an outline of the document.  Additionally, this clause provides guidance on how consumers that enable interactive features such as hyperlinks, selection, and accessibility tools should use the format. It also describes how producers should emit content to enable interactive features.
XPS Document Package Features (§17)	This clause describes how package features (as described in the OPC specification) are used and extended in the XPS Document format. This includes interleaving, digital signatures, and core properties.
Rendering Rules (§18)	This clause provides precise instructions for rendering XPS Document contents to ensure a consistent result among various implementations.
Elements (§19)	The full list of elements described throughout the preceding clauses is assembled in this clause, in alphabetical order, for easy reference.
Signature Definitions Schema (§A)	This annex includes the XSD schema for the Signature Definitions part.
XPS Document Schema (§B)	This annex includes the XSD schema for the FixedDocumentSequence, FixedDocument, and FixedPage parts.
Resource Dictionary Key Schema (§C)	This annex provides the XSD schema for the resource dictionary <b>Key</b> attribute, used by several elements in the XPS Document schema.
Document Structure Schema (§D)	This annex provides the XSD schema for the DocumentStructure and StoryFragments parts.
Discard Control Schema (§E)	This annex includes the XSD schema for the DiscardControl part for interleaving.
Abbreviated Geometry Syntax Algorithm (§F)	This annex provides a sample algorithm for interpreting the abbreviated geometry syntax provided to succinctly describe geometric regions in a single attribute.

Clause	Description
scRGB Gamut Boundary Definition (§G)	The scRGB color space's gamut boundary used by this specification is defined in this annex.
Standard Namespaces and Content Types (§H)	This annex defines all of the XML namespace names, content types, and relationship types used by all XPS Document parts and relationships.
Conformance Requirements (§I)	This annex assembles all the conformance requirements specified throughout the previous clauses and annexes into a comprehensive list for reference purposes.

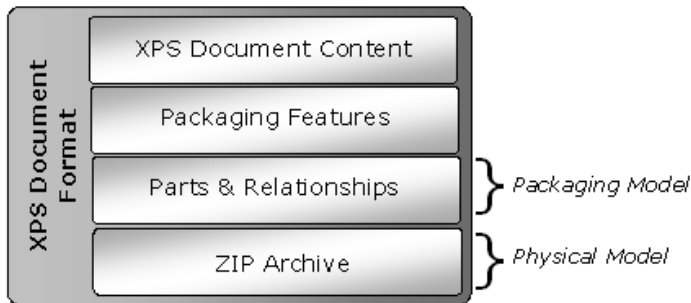
1 **End of informative text**

2 **8.2 Package**

3 The XPS Document format MUST use a ZIP archive for its **physical model** [M1.2].  
 4 The OPC specification describes a packaging model; that is, how the package is  
 5 represented internally with parts and relationships.

6 The XPS Document format includes a well-defined set of parts and relationships,  
 7 each fulfilling a particular purpose in the document. The format also extends the  
 8 package features, including digital signatures, thumbnails, and interleaving.

9 *Figure 8-1. Package-based XPS Document format*



10

## 1 9. Parts and Relationships

2 The packaging conventions described in the OPC specification can be used to carry  
3 any payload. A [payload](#) is a complete collection of interdependent parts and  
4 relationships within a package. This specification defines a particular payload that  
5 contains a static or “fixed-layout” representation of paginated content: the fixed  
6 payload.

7 A package that holds at least one [fixed payload](#) and follows the rules described in  
8 this specification is referred to as an [XPS Document](#). Producers and consumers of  
9 XPS Documents can implement their own parsers and rendering engines based on  
10 this specification.

11 XPS Documents address the requirements that information workers have for  
12 distributing, archiving, rendering, and processing documents. Using known rendering  
13 rules, XPS Documents can be unambiguously reproduced or printed without tying  
14 client devices or applications to specific operating systems or service libraries.  
15 Because the XPS Document is expressed in a neutral, application-independent way,  
16 the content can be viewed and printed without the application used to create the  
17 package.

---

### 18 9.1 Fixed Payload

19 A [payload](#) that has a [FixedDocumentSequence](#) root part is known as a [fixed payload](#).  
20 A [fixed payload root](#) is a [FixedDocumentSequence](#) part that references  
21 [FixedDocument](#) parts that, in turn, reference [FixedPage](#) parts. There can be more  
22 than one fixed payload in an XPS Document.

23 A specific relationship type is defined to identify the root of a [fixed payload](#) within an  
24 XPS Document: the [XPS Document StartPart relationship](#). The [primary fixed payload](#)  
25 [root](#) is the [FixedDocumentSequence](#) part that is referenced by the XPS Document  
26 [StartPart](#) relationship. Consumers such as viewers or printers use the XPS Document  
27 [StartPart](#) relationship to find the primary fixed payload in a package. The XPS  
28 Document [StartPart](#) relationship MUST point to the [FixedDocumentSequence](#) part  
29 that identifies the root of the fixed payload [M2.14].

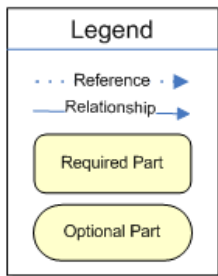
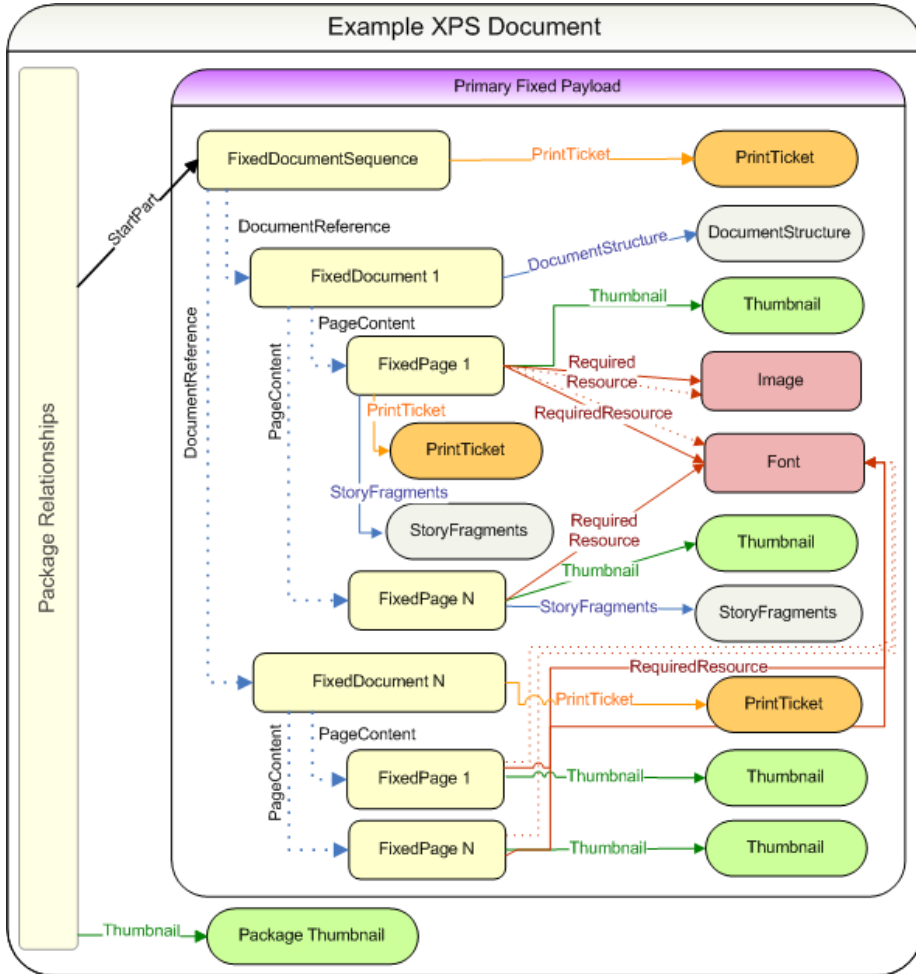
30 The payload includes the full set of parts required for processing the  
31 [FixedDocumentSequence](#) part. All content to be rendered MUST be contained in the  
32 XPS Document [M2.1]. The parts that can be found in an XPS Document are listed in  
33 Table 9–1. Relationships and content types for these parts are defined in H,  
34 “Standard Namespaces and Content Types.” Each part MUST use *only* the  
35 appropriate content type specified in H [M2.2].

36 Table 9–1. XPS Document parts

Name	Description	Required/Optional
<a href="#">FixedDocumentSequence</a> (§9.1.2)	Specifies a sequence of fixed documents.	REQUIRED [M2.3]
<a href="#">FixedDocument</a> (§9.1.3)	Specifies a sequence of fixed pages.	REQUIRED [M2.4]

FixedPage (§9.1.4)	Contains the description of the contents of a page.	REQUIRED [M2.5]
Font (§9.1.7)	Contains an OpenType or TrueType font.	REQUIRED if a <Glyphs> element is present [M2.6]
Image (§9.1.5) JPEG image (§9.1.5.1) PNG image (§9.1.5.2) TIFF image (§9.1.5.3) Windows Media Photo image (§9.1.5.4)	References an image file.	REQUIRED if an <ImageBrush> element is present [M2.7]
Remote resource dictionary (§9.1.8)	Contains a resource dictionary for use by fixed page markup.	REQUIRED if a key it defines is referenced [M2.8]
Thumbnail (§)9.1.6	Contains a small JPEG or PNG image that represents the contents of the page or package.	OPTIONAL [O2.1]
PrintTicket (§9.1.9)	Provides settings to be used when printing the package.	OPTIONAL [O2.2]
ICC profile (§??)	Contains an ICC Version 2 color profile optionally containing an embedded Windows Color System (WCS) color profile.	OPTIONAL [O2.3]
DocumentStructure (§9.1.11)	Contains the document outline and document contents (story definitions) for the XPS Document.	OPTIONAL [O2.4]
StoryFragments (§)9.1.12	Contains document content structure for a fixed page.	OPTIONAL [O2.5]
SignatureDefinitions (§9.1.10)	Contains a list of digital signature spots and signature requirements.	OPTIONAL [O2.6]
DiscardControl (§??)	Contains a list of resources that are safe for consumers to discard during processing.	OPTIONAL [O2.7]

1 Example 9-1. A typical XPS Document



2

3 end example]

### 1 9.1.1 Fixed Payload Relationships

2 Internal resources are associated with parts by relationships and inline references.  
 3 XPS Documents MUST NOT reference external XPS resources [M2.1]. In general,  
 4 inline resource references are represented inside the referring part in ways that are  
 5 specific to the content type of the part, that is, in arbitrary markup or application-  
 6 specific encoding. Relationships represent the type of connection between a source  
 7 part and a target resource, and they allow parts to be related without modifying  
 8 them. For more information, see the OPC specification.

9 Resources, which include fonts, images, color profiles, and remote resource  
 10 dictionaries, that are referenced by inline URIs but are necessary to render the page  
 11 MUST use the Required Resource relationship from the FixedPage part to the  
 12 resource [M2.10]. If any resource references *other* resources, the producer MUST  
 13 also use the Required Resource relationship from the FixedPage part to the indirectly  
 14 referenced resource [M2.10].

15 It is RECOMMENDED that there be exactly *one* Required Resource relationship from  
 16 the FixedPage part for each resource referenced from markup [S2.1]. Multiple  
 17 Required Resource relationships from a FixedPage part to a resource are not  
 18 considered an error, but they reduce efficiency. It is not considered an error if a  
 19 FixedPage part that does not use a specific resource in its markup references the  
 20 resource via a Required Resource relationship; however, doing so may reduce  
 21 efficiency for consumers.

22 Relationship types are defined in H, "Standard Namespaces and Content Types."

23 Table 9–2. Fixed payload relationships

Name	Description	Required/Optional
Core Properties	Relationship from the package to the Core Properties part.	OPTIONAL [O2.8]
Digital Signature Origin	Relationship from the package to the Digital Signature Origin part.	OPTIONAL [O2.9]
Digital Signature	Relationship from the Digital Signature Origin part to an Digital Signature XML Signature part.	OPTIONAL [O2.10]
Digital Signature Certificate	Relationship from a Digital Signature XML Signature part to a Digital Signature Certificate part.	OPTIONAL [O2.11]
Digital Signature Definitions	Relationship from the FixedDocument part to a Digital Signature Definitions part.	OPTIONAL [O2.12]
DiscardControl	Relationship from the package to a DiscardControl part.	OPTIONAL [O2.13]
DocumentStructure	Relationship from the FixedDocument part to a DocumentStructure part.	OPTIONAL [O2.14]
PrintTicket	Relationship from a FixedDocumentSequence part, a FixedDocument part, or a FixedPage	OPTIONAL [O2.15]



	part to a PrintTicket part.	
Required Resource	Relationship from a FixedPage part to a required resource, including Font, Image, ColorProfile, and Remote Resource Dictionary parts. Required resources can be shared between pages.	REQUIRED for each resource referenced from a FixedPage [M2.10]
Restricted Font	Relationship from a FixedDocument part to a Font part. Specifies the referenced font as restricted, disallowing any modification or editing of any <Glyphs> element text using the referenced font.	REQUIRED for each preview and print font used [M2.12]
StartPart	Relationship from the package to the FixedDocumentSequence part that is the primary fixed payload root.	REQUIRED [M2.13, M2.14]
StoryFragments	Relationship from a FixedPage part to the StoryFragments part for the page.	OPTIONAL [O2.16]
Thumbnail	Relationship from the package to an Image part or from a FixedPage part to an Image part.	OPTIONAL [O2.17]

1 Producers that generate a relationship MUST include the target part in the XPS  
 2 Document for any of the following relationship types: DiscardControl,  
 3 DocumentStructure, PrintTicket, Required Resource, Restricted Font, StartPart,  
 4 StoryFragments, and Thumbnail. Consumers that access the target part of any  
 5 relationship with one of these relationship types MUST generate an error if the part is  
 6 not included in the XPS Document [M2.77].

### 7 9.1.2 FixedDocumentSequence Part

8 The FixedDocumentSequence part assembles a set of fixed documents within the  
 9 fixed payload. [Example: A printing client can assemble two separate documents, a  
 10 two-page cover memo and a twenty-page report (both are FixedDocument parts),  
 11 into a single package to send to the printer. end example]

12 The FixedDocumentSequence part is the only valid root of a fixed payload. Even if an  
 13 XPS Document contains only a single fixed document, the FixedDocumentSequence  
 14 part is still used. One FixedDocumentSequence part per fixed payload is REQUIRED  
 15 [M2.3].

16 Fixed document sequence markup specifies each fixed document in the fixed payload  
 17 in sequence, using <DocumentReference> elements. The order of  
 18 <DocumentReference> elements determines document order and MUST be  
 19 preserved by editing consumers [M2.15]. Each <DocumentReference> element  
 20 SHOULD reference a FixedDocument part by relative URI [S2.2]. For more  
 21 information, see §10.1, "<FixedDocumentSequence> Element," on page 49.

22 The content type of the FixedDocumentSequence part is defined in H, "Standard  
 23 Namespaces and Content Types."

### 1 **9.1.3 FixedDocument Part**

2 The **FixedDocument part** is a common, easily indexed root for all pages within the  
3 document. A fixed document identifies the set of fixed pages for the document.

4 The markup in the FixedDocument part specifies the pages of a document in  
5 sequence using <PageContent> elements. The order of <PageContent> elements  
6 determines page order and MUST be preserved by editing consumers [M2.16]. Each  
7 <PageContent> element SHOULD reference a FixedPage part by relative URI [S2.3].  
8 For more information, see §10.2, "<FixedDocument> Element," on page 50.

9 The content type of the FixedDocument part is defined in H, "Standard Namespaces  
10 and Content Types."

### 11 **9.1.4 FixedPage Part**

12 The **FixedPage part** contains all of the visual elements to be rendered on a page.  
13 Each page has a fixed size and orientation. The layout of the visual elements on a  
14 page is determined by the fixed page markup. This applies to both graphics and text,  
15 which are represented with precise typographic placement. The contents of a page  
16 are described using a powerful but simple set of visual primitives.

17 Each FixedPage part specifies the contents of a page within a <FixedPage> element  
18 using <Path> and <Glyphs> elements (using various brush elements) and the  
19 <Canvas> grouping element. The <ImageBrush> and <Glyphs> elements (or their  
20 child or descendant elements) can reference Image parts or Font parts by URI. They  
21 SHOULD reference these parts by relative URI [S2.4]. For more information,  
22 see §10.3, "<FixedPage> Element," on page 53.

23 The content type of the FixedPage part is defined in H, "Standard Namespaces and  
24 Content Types."

### 25 **9.1.5 Image Parts**

26 Image parts reference image files. A single image may be shared among multiple  
27 fixed pages in one or more fixed documents. Images referenced in markup MUST be  
28 internal to the package [M2.1]. References to images that are external to the  
29 package are invalid.

30 Images are included in XPS Documents with an <ImageBrush> element and an  
31 **ImageSource** attribute to reference a part with the appropriate content type. For  
32 more information, see §H.2, "Content Types," on page 406. Fixed pages MUST use a  
33 Required Resource relationship to each Image part referenced [M2.10]. For more  
34 information, see §H.3, "Relationship Types," on page 407.

35 XPS Documents support the following image formats:

- 36 • JPEG
- 37 • PNG
- 38 • TIFF
- 39 • Windows Media Photo

40 Color profiles MAY be embedded in image files [O2.18] and SHOULD be used by  
41 consumers [S2.5]. See 15, "Color," for more details.

1 **9.1.5.1 JPEG Images**

2 It is RECOMMENDED that JPEG image part names end with the extension “.jpg”  
 3 [S2.6]. JPEG image parts MUST contain images that conform to the JPEG  
 4 specification [M2.17]. Consumers SHOULD support JPEG images that contain APP0,  
 5 APP2, APP13, and APP14 markers [S2.34]. Consumers MUST support JPEG images  
 6 that contain the APP1 marker and interpret the EXIF color space correctly [M2.78].

7 *Table 9–3. Supported JPEG APPn markers*

APPn marker	Originating source
APP0	JFIF specification
APP1	EXIF extension defined by JEITA
APP2	ICC profile marker defined by the ICC specification
APP13	Photoshop 3.0 extension
APP14	Adobe DCT Filters in PostScript Level 2 extension

8 Some JPEG implementations have limited support for CMYK JPEG images, such as:

- 9 • CMYK is converted to RGB in the decoder using fixed tables instead of the  
 10 supplied ICC profile.
- 11 • ICC Profiles embedded using APP2 are limited in length, because APPn marker  
 12 chunking is not supported.

13 Therefore, the use of CMYK images is NOT RECOMMENDED in XPS Documents  
 14 because rendering results can differ significantly between implementations. TIFF or  
 15 Windows Media Photo images SHOULD be used instead to represent CMYK images  
 16 [S2.7].

17 If both APP2 and APP13 markers are specified, the APP2 marker takes precedence. If  
 18 the JPEG image is embedded in a TIFF image, the TIFF ICC profile settings are used.

19 **9.1.5.2 PNG Images**

20 It is RECOMMENDED that PNG image part names end with the extension “.png”  
 21 [S2.8]. PNG image parts MUST contain images that conform to the PNG specification  
 22 [M2.18].

23 *Table 9–4. Support for ancillary PNG chunks*

Chunk	Support Level
tRNS	MUST Support [M2.19]
iCCP	MUST Support [M2.20]
sRGB	MUST Ignore [M2.21]
cHRM	MUST Ignore [M2.22]
gAMA	MUST Ignore [M2.23]
sBIT	MUST Ignore [M2.24]

24 **9.1.5.3 TIFF Images**

25 It is RECOMMENDED that TIFF image part names end with the extension “.tif”  
 26 [S2.9]. TIFF image parts MUST contain images that conform to the TIFF specification

1 [M2.25]. XPS Document consumers MUST support baseline TIFF 6.0 with some  
 2 extensions, as noted in Table 9-5 [M2.26]. These tags MUST be supported for the  
 3 specified image types [M2.26]. If consumers encounter a tag that is not included  
 4 below, they SHOULD ignore that tag [S2.10].

5 *Table 9-5. Supported TIFF 6.0 tags*

<b>Image type</b>	<b>Tags</b>
Bilevel images	PhotometricInterpretation (0 and 1) Compression (1, 2, 3, 4, 5, or 32773) ImageLength ImageWidth ResolutionUnit (1, 2, or 3) RowsPerStrip StripByteCounts StripOffsets XResolution YResolution
Grayscale images	PhotometricInterpretation (0 and 1) BitsPerSample (4, 8, or 16) Compression (1, 5, 6, or 32773) ImageLength ImageWidth ResolutionUnit (1, 2, or 3) RowsPerStrip StripByteCounts StripOffsets XResolution YResolution
Palette color images	BitsPerSample (1, 4, or 8) ColorMap Compression (1, 5, or 32773) ImageLength ImageWidth PhotometricInterpretation (3) ResolutionUnit (1, 2, or 3) RowsPerStrip StripByteCounts StripOffsets XResolution YResolution
RGB images	BitsPerSample (8,8,8 or 16,16,16; or if ExtraSamples=1 or 2: 8,8,8,8 or 16,16,16,16) Compression (1, 5, 6, or 32773)

---

	ExtraSamples (0, 1, or 2)
	ICC Color Profile [tag 34675]
	ImageLength
	ImageWidth
	PhotometricInterpretation (2)
	PlanarConfiguration (1)
	ResolutionUnit (1, 2, or 3)
	RowsPerStrip
	SamplesPerPixel (3; <i>or</i> if ExtraSamples=1 or 2: 4)
	StripByteCounts
	StripOffsets
	XResolution
	YResolution
CMYK images (TIFF extension)	BitsPerSample (8,8,8,8 or 16,16,16,16; <i>or</i> if ExtraSamples=1 or 2: 8,8,8,8 or 16,16,16,16)
	Compression (1, 5, 6, or 32773)
	ExtraSamples (0, 1, or 2)
	ICC Color Profile [tag 34675]
	ImageLength
	ImageWidth
	InkSet (1)
	NumberOfInks (4)
	PhotometricInterpretation (5)
	PlanarConfiguration (1)
	ResolutionUnit (1, 2, or 3)
	RowsPerStrip
	SamplesPerPixel (4; <i>or</i> if ExtraSamples=1 or 2: 5)
	StripByteCounts
	StripOffsets
	XResolution
	YResolution

---

- 1 If the TIFF image contains multiple image file directories (IFDs), consumers MUST
- 2 use only the first IFD and ignore all others [M2.27].
- 3 If the ResolutionUnit tag is set to 1 (no units), XResolution and YResolution are
- 4 interpreted in the same manner as if the ResolutionUnit was set to 2 (inches).
- 5 If no color profile is embedded in the TIFF image or stored in a separate part
- 6 associated with the TIFF image according to the mechanisms described in §15.2.8,
- 7 "Images and Color Profile Association" on page 216, then the default color space
- 8 MUST be treated as defined in §15.2.9, "Color Space Pixel Formats for Raster
- 9 Images," on page 216 [M8.30].

1 The following features of the TIFF specification MUST be supported in addition to the  
2 tags described in Table 9-5:

- 3 • Baseline TIFF (Sections 1-10) with the exception of the following tags  
4 [M2.26]:
  - 5 ○ CellLength
  - 6 ○ CellWidth
  - 7 ○ GrayResponseCurve
  - 8 ○ GrayResponseUnit
  - 9 ○ MaxSampleValue
  - 10 ○ MinSampleValue
  - 11 ○ Orientation
  - 12 ○ Thresholding
- 13 • CCITT bilevel encodings (Section 11) [M2.28]
- 14 • CMYK images (Section 16) [M2.29]

- 1 • Associated alpha data (Section 18) [M2.30]
- 2     o ExtraSamples tag value of 1: Treat alpha as pre-multiplied alpha
- 3         (see §18.4.1, "Pre-Multiplied Alpha and Superluminous Colors," on page
- 4         281 for details)
- 5     o ExtraSamples tag value of 2: Treat alpha as non-pre-multiplied alpha
- 6 • LZW compression (Section 13) [M2.31]
- 7 • Differencing predictors (Section 14) [M2.32]
- 8 • JPEG compression (Section 22)
- 9     o Only compression mode 6 MUST be supported [M2.33]
- 10 • Embedded ICC Profile (described in the ICC specification) [M2.34]
- 11 • EXIF IFD (tag 34665) as described in the EXIF specification. The EXIF color
- 12     space MUST be interpreted correctly [M2.79].

13 Consumers that support tags and features not described above can result in  
 14 undesirable differences in the appearance of XPS Documents. Producers cannot rely  
 15 on a consistent interpretation of tags or features that are not described above and  
 16 therefore SHOULD NOT use any such tags or features [S2.10].

17 [Note: Many TIFF images in circulation today deviate from the TIFF Specification.  
 18 Over time, TIFF-consuming implementations have developed a certain tolerance for  
 19 such deviations by attempting to deduce the intent of the TIFF image author and  
 20 correct for apparent errors or deviations. *end note*]

21 Therefore, XPS Document consumers SHOULD do the following [S2.11]:

- 22 • Test with as many different TIFF images as possible.
- 23 • Correct common mistakes in TIFF images, such as:
  - 24     o Not all BitsPerSample hold the same value
  - 25     o Number of BitsPerSample does not match SamplesPerPixel
  - 26     o PhotometricInterpretation 1 or 2 (instead of 3) used when BitsPerSample
  - 27         is set to "8,8,8"
  - 28     o Missing ExtraSamples tag, but SamplesPerPixel does not match the value
  - 29         expected by the PhotometricInterpretation tag, thus necessitating
  - 30         deduction of the ExtraSamples value.
- 31 • Implement a reasonable recovery strategy when a problematic TIFF image is
- 32     encountered.

**Comment [rcj3]:** What is "reasonable"?

#### 33 9.1.5.4 Windows Media Photo Images

34 It is RECOMMENDED that Windows Media Photo image part names end with the  
 35 extension ".wdp" [S2.12]. Windows Media Photo image files MUST conform to the  
 36 Windows Media Photo specification [M2.35]. XPS Documents support Windows Media  
 37 Photo images with the characteristics identified in Table 9–6.

1 Table 9–6. Supported Windows Media Photo features

Color space	Pixel formats	Compression	Alpha
Grayscale	8-bit integer	Lossy	None
	16-bit integer	Lossless	
	16-bit half-float*		
	16-bit fixed point*		
sRGB	8-bit integer	Lossy	1-channel
	16-bit integer	Lossless	1-channel pre-multiplied
scRGB	16-bit half-float	Lossy	1-channel
	16-bit fixed point	Lossless	1-channel pre-multiplied
	32-bit IEEE float		RGBE-Radiance (no alpha channel)
	32-bit fixed point RGBE-Radiance		
CMYK	8-bit integer	Lossy	1-channel independent
	16-bit integer	Lossless	
N-channel (including named color N-tone)	8-bit integer	Lossy	1-channel independent
	16-bit integer	Lossless	
Profiled RGB (3-channel)	8-bit integer	Lossy	1-channel
	16-bit integer	Lossless	1-channel pre-multiplied

2 \* The value range of these formats is the same as scRGB.

3 If no color profile is embedded in the Windows Media Photo image or stored in a  
 4 separate part associated with the Windows Media Photo image according to the  
 5 mechanisms described in §15.2.8, “Images and Color Profile Association” on page  
 6 216, then the default color space MUST be treated as defined in §15.2.9, “Color  
 7 Space Pixel Formats for Raster Images,” on page 216 [M8.30].

#### 8 9.1.6 Thumbnail Parts

9 Thumbnails are small images that represent the contents of a fixed page or an entire  
 10 XPS Document. Thumbnails enable users of viewing applications to select a page  
 11 easily.

12 Thumbnail images MAY be attached using a relationship to the FixedPage parts  
 13 [O2.19]. Each FixedPage part MUST NOT have more than one thumbnail part  
 14 attached [M2.36]. Relationships to thumbnail parts are defined in H, “Standard  
 15 Namespaces and Content Types.” It is RECOMMENDED that if thumbnails are used  
 16 for pages, a thumbnail SHOULD be included for every page in the document [S2.13].

17 Although the OPC specification allows thumbnails to be attached to any part, XPS  
 18 Document consumers SHOULD only process thumbnails associated via a package

**Comment [rcj4]:** Does this mean a different thumbnail for each page? Clarify.



1 relationship from the package as a whole or via a relationship from a FixedPage part  
2 [S2.14]. These thumbnails MUST be in either JPEG or PNG format [M2.37].  
3 Thumbnails attached to any other part SHOULD be ignored by XPS Document  
4 consumers [S2.14]. The content types of thumbnail parts are specified in §H.2,  
5 “Content Types,” on page 406.

6 For more information about the relationship type for thumbnail parts, see §H.3,  
7 “Relationship Types,” on page 407.

### 8 **9.1.7 Font Parts**

9 Fonts are stored in font parts. XPS Documents MUST support the OpenType font  
10 format, which includes TrueType and CFF fonts [M2.39]. To support portability,  
11 Unicode-encoded fonts SHOULD be used (see §9.1.7.5, “Non-Standard Font  
12 Compatibility Encoding” for additional information) [S2.15].

13 Font parts are referenced using the **FontUri** attribute of the <Glyphs> element. A  
14 single font may be shared among multiple fixed pages in one or more fixed  
15 documents. Font references MUST be internal to the package; external references to  
16 fonts are invalid [M2.1].

17 If the referenced font part is a TrueType Collection, the fragment portion of the URI  
18 indicates the font face to be used. The use of URI fragments is specified in the BNF  
19 of Generic URI Syntax specification. The fragment contained in the **FontURI**  
20 attribute value MUST be an integer between 0 and  $n-1$  inclusive, where  $n$  is the  
21 number of font faces contained in the TrueType Collection [M2.38]. [*Example*: To  
22 reference the first font face in the font part “../Resources/Fonts/CJKSuper.ttc”, the  
23 value of the **FontUri** attribute is “../Resources/Fonts/CJKSuper.ttc#0”. *end*  
24 *example*] If no fragment is specified, the first font face is used in the same way as if  
25 the URI had specified “#0”.

26 Content types for fonts differ depending on whether the font is non-obfuscated or  
27 obfuscated (see §9.1.7.2, “OpenType Font Embedding,” on page 31). Content types  
28 are summarized in H, “Standard Namespaces and Content Types.”

29 Fixed pages MUST use a Required Resource relationship to each Font parts  
30 referenced [M2.10]. For more information, see H.

#### 31 **9.1.7.1 Subsetting Fonts**

32 XPS Documents represent text using the <Glyphs> element. Since the format is  
33 fixed, it is possible to create a font subset that contains only the glyphs required by  
34 the package. Fonts MAY be subsetted based on glyph usage [O2.20]. Although a  
35 subsetted font does not contain all the glyphs in the original font, it MUST be a valid  
36 OpenType font file [M2.39]. Requirements for valid OpenType font files are described  
37 in the OpenType Font File specification.

#### 38 **9.1.7.2 OpenType Font Embedding**

39 Protecting the intellectual property of font vendors is a goal of the XPS Document  
40 format. Therefore, producers MUST observe the guidelines and mechanisms  
41 described below in order to honor the licensing rights specified in OpenType fonts  
42 [M2.40]. It is not the responsibility of consumers to enforce font licensing intent,  
43 although consumers MUST be able to process XPS Documents using any combination

1 of these embedding and obfuscation mechanisms, even if produced in violation of  
2 these guidelines [M2.41].

3 The licensing rights of an OpenType font are specified in the fsType field of the  
4 required OS/2 table in the font file. Table 9–7 lists the bit mask values that can  
5 appear in arbitrary combinations in the fsType field. Also listed are short descriptions  
6 of the licensing right intents and requirements or recommendations. These  
7 requirements represent the “rules” that producers and consumers must follow in  
8 order to respect licensing rights specified in the font.

9 For further details on licensing rights of OpenType fonts, see the description of the  
10 OS/2 table in “OS/2 and Windows Metrics.”

11 Table 9–7. Guidelines for OpenType font embedding

Bit/mask	Licensing right intent	Producer rules	Consumer rules
– / 0x0000	Installable embedding.	SHOULD do embedded font obfuscation [S2.16] (see §9.1.7.3, “Embedded Font Obfuscation,” on page 33 for details).	SHOULD NOT extract or install permanently (see below) [S2.17].
0 / 0x0001	Reserved, must be 0.		
1 / 0x0002	Restricted license embedding. If <i>only</i> this bit is set, the font MUST NOT be modified, embedded or exchanged in any manner without obtaining permission from the legal owner.	MUST NOT embed [M2.42]. SHOULD generate a path filled with an image brush referencing an image of rendered characters [S2.18]. SHOULD include the text in the <b>AutomationProperties.Name</b> attribute of the <Path> element [S2.18].	Render embedded images.
2 / 0x0004	For preview and print embedding, font may be embedded and temporarily used on remote systems. However, documents containing <i>any</i> preview and print fonts MUST NOT be modified or edited [M2.43].	MUST do embedded font obfuscation [M2.44] (see §9.1.7.3). MUST add a Restricted Font relationship to the FixedDocument part of the document containing the font [M2.12]. See §12.1.7, “DeviceFontName Attribute,” on page 115 and §H.3, “Relationship Types,” on page 407 for details.	MUST NOT extract or install permanently [M2.45]. MUST NOT modify or edit the XPS Document markup or hierarchical structure starting from the <FixedDocument> element [M2.43].
3 / 0x0008	Editable embedding.	MUST do embedded font obfuscation [M2.46] (see §9.1.7.3).	MUST NOT extract or install permanently [M2.47].
4–7	Reserved, must be		

	0.		
8 / 0x0100	No subsetting.	MUST do embedded font obfuscation (see §9.1.7.3) [M2.48]. MUST NOT subset font before embedding.	MUST NOT extract or install permanently [M2.50].
9 / 0x0200	Bitmap embedding only.	MUST do embedded font obfuscation [M2.51] (see §9.1.7.3). MUST embed <i>only</i> bitmap characters contained in the font [M2.51]. If no bitmap characters are present in the font, MUST NOT embed the font [M2.51].	MUST NOT extract or install permanently [M2.52].
10–15	Reserved, must be 0.		

### 1 9.1.7.3 Embedded Font Obfuscation

2 Embedded font obfuscation is a means of preventing casual misappropriation of  
3 embedded fonts. Specifically, embedded font obfuscation prevents end-users from  
4 using standard ZIP utilities to extract fonts from XPS Document files and install them  
5 on their systems.

6 Embedded font obfuscation is *not* considered a strong encryption of the font data.

7 Embedded font obfuscation achieves the following goals:

- 8 4. Obfuscated font files are embedded within an XPS Document package in a  
9 form that cannot be directly installed on any client operating system.
- 10 5. Obfuscated font files are closely tied to the content referencing them.  
11 Therefore, it is non-trivial to misappropriate fonts by moving them from one  
12 package to another.
- 13 6. The manner in which obfuscated font files are tied to the content referencing  
14 them still allows for document merging.

15 For information on how to determine when fonts must be obfuscated prior to  
16 embedding, see Table 9–7. above.

17 Although the licensing intent allows embedding of non-obfuscated fonts and  
18 installation of the font on a remote client system under certain conditions, this is  
19 NOT RECOMMENDED in XPS Documents. [*Note*: Microsoft implementations for XPS  
20 Documents always perform obfuscated font embedding and do not extract or  
21 permanently install the font [S2.19]. *end note*] However, there are vertical solutions  
22 in which implementations may benefit from un-obfuscated font embedding. In these  
23 cases, implementations could omit obfuscation or extract and install the embedded  
24 font.

1 If a producer is required to perform embedded font obfuscation, it MUST satisfy the  
2 following requirements [M2.53]:

- 3 1. Generate a 128-bit GUID (Globally Unique Identifier) for the font to be  
4 obfuscated. Instead of a true GUID, a 128-bit random number MAY be used  
5 [O2.21]. The 16 bytes of the 128-bit GUID are referred to in the following text  
6 by the placeholder names  $B_{00}$ ,  $B_{01}$ ,  $B_{02}$ ,  $B_{03}$ ;  $B_{10}$ ,  $B_{11}$ ;  $B_{20}$ ,  $B_{21}$ ;  $B_{30}$ ,  $B_{31}$ ,  $B_{32}$ ,  
7  $B_{33}$ ,  $B_{34}$ ,  $B_{35}$ ,  $B_{36}$ , and  $B_{37}$ . The order in which bytes are assigned to these  
8 placeholders does not matter, as long as it is consistent for obfuscation and  
9 de-obfuscation.
- 10 2. Generate a part name for the obfuscated font using the GUID. The last  
11 segment of the part name MUST be of the form " $B_{03}B_{02}B_{01}B_{00}-B_{11}B_{10}-B_{21}B_{20}-$   
12  $B_{30}B_{31}-B_{32}B_{33}B_{34}B_{35}B_{36}B_{37}$ " or " $B_{03}B_{02}B_{01}B_{00}-B_{11}B_{10}-B_{21}B_{20}-B_{30}B_{31}-$   
13  $B_{32}B_{33}B_{34}B_{35}B_{36}B_{37}.ext$ " where each  $B_x$  represents a placeholder for one byte of  
14 the GUID, represented as two hex digits [M2.54]. The part name MAY have an  
15 arbitrary extension (identified by the placeholder ".ext") [O2.22]. It is  
16 RECOMMENDED that the extension for TrueType fonts be ".odttf" and for  
17 TrueType collections be ".odttc" [S2.20].
- 18 3. The content type for the part containing the obfuscated font MUST match the  
19 definition in H, "Standard Namespaces and Content Types" [M2.2].
- 20 4. Perform an XOR operation on the first 32 bytes of the binary data of the font  
21 part with the array consisting of the bytes referred to by the placeholders  $B_{37}$ ,  
22  $B_{36}$ ,  $B_{35}$ ,  $B_{34}$ ,  $B_{33}$ ,  $B_{32}$ ,  $B_{31}$ ,  $B_{30}$ ,  $B_{20}$ ,  $B_{21}$ ,  $B_{10}$ ,  $B_{11}$ ,  $B_{00}$ ,  $B_{01}$ ,  $B_{02}$ , and  $B_{03}$ , in that  
23 order and repeating the array once. The result is an obfuscated font.
- 24 5. Store the obfuscated font in a part with the generated name.

25 When processing fonts, consumers MUST follow these steps [M2.53]:

- 26 1. If the content type of the part containing the font is not the obfuscated font  
27 content type as specified in H, process the font without any de-obfuscation  
28 steps.
- 29 2. For font parts with the obfuscated font content type as specified in H, de-  
30 obfuscate the font by following these rules:
  - 31 a. Remove the extension from the last segment of the name of the part  
32 containing the font.
  - 33 b. Convert the remaining characters of the last segment to a GUID using the  
34 byte ordering described above.
  - 35 c. Perform an XOR operation on the first 32 bytes of the binary data of the  
36 obfuscated font part with the array consisting of the bytes referred to by  
37 the placeholders  $B_{37}$ ,  $B_{36}$ ,  $B_{35}$ ,  $B_{34}$ ,  $B_{33}$ ,  $B_{32}$ ,  $B_{31}$ ,  $B_{30}$ ,  $B_{20}$ ,  $B_{21}$ ,  $B_{10}$ ,  $B_{11}$ ,  $B_{00}$ ,  
38  $B_{01}$ ,  $B_{02}$ , and  $B_{03}$ , in that order and repeating the array once. The result is a  
39 non-obfuscated font.
  - 40 d. Use the non-obfuscated font for the duration of the document processing,  
41 but do not leave any local or otherwise user-accessible copy of the non-  
42 obfuscated font.

1 **9.1.7.4 Print and Preview Restricted Fonts**

2 If a producer embeds a font with the print and preview restriction bit set, it MUST  
 3 also add a Restricted Font relationship from the FixedDocument part that includes  
 4 the FixedPage referencing the font to the restricted font [M2.12].

5 Editing consumers MUST NOT edit a document where the FixedDocument part has a  
 6 Restricted Font relationship [M2.43]. When invoking editing functionality, editing  
 7 consumers MUST treat as an error any font with the print and preview restriction bit  
 8 set for which no Restricted Font relationship has been added to the FixedDocument  
 9 part [M2.12].

10 **Printing and display-only consumers** MUST consider an XPS Document valid even if  
 11 the producer failed to properly set the Restricted Font relationship [M2.12].

**Comment [rcj5]:** Are these subcategories of consumer that need defining?

12 **9.1.7.5 Non-Standard Font Compatibility Encoding**

13 When processing <Glyphs> elements, the consumer MUST first select a cmap table  
 14 from the OpenType font following the order of preference shown below (highest  
 15 listed first) [M2.55]:

16 *Table 9-8. Cmap table selection*

Platform ID	Encoding ID	Description
3	10	Unicode with surrogates
3	1	Unicode without surrogates
3	5	Wansung
3	4	Big5
3	3	Prc
3	2	ShiftJis
3	0	Symbol
0	Any	Unicode (deprecated)
1	0	MacRoman

17 All further processing for that font MUST use the selected cmap table [M2.55].

18 If a Wansung, Big5, Prc, ShiftJis or MacRoman cmap has been selected, the  
 19 consumer MUST correctly map from Unicode codepoints in the UnicodeString to the  
 20 corresponding codepoints used by the cmap before looking up the glyphs [M2.56].  
 21 The Unicode standard provides details of the required mappings.

22 Producers SHOULD avoid using fonts lacking a Unicode-encoded cmap table [S2.15].

23 When processing <Glyphs> elements that reference a cmap (3,0) encoding font,  
 24 consumers MUST be prepared for the case in which the **UnicodeString** attribute  
 25 contains character codes instead of PUA codepoints [M2.57]. This condition is  
 26 indicated by an unsuccessful Unicode lookup of the codepoint specified in the  
 27 Unicode string in the cmap (3,0) table. In this case, the correct glyph index is  
 28 computed by following the general recommendations of the OpenType specification.

29 When processing <Glyphs> elements that use this compatibility encoding, character  
 30 codes in the range 0x20-0xff are mapped to PUA codepoints. Therefore, character

1 codes in the range 0x80-0x9f are not considered non-printable Unicode control  
2 codes.

3 This non-standard encoding has been included to facilitate document production for  
4 certain producers. However, there are significant drawbacks resulting from this  
5 encoding:

- 6 • Search is unpredictable
- 7 • Copy and paste functionality is unpredictable

8 Producers are discouraged from using this non-standard encoding and they SHOULD  
9 write PUA codepoints to the **UnicodeString** attribute [S2.15].

**Comment [rcj6]:** Should this be written using SHOULD NOT?

### 10 **9.1.8 Remote Resource Dictionary Parts**

11 A **remote resource dictionary** allows producers to define resources that can be  
12 reused across many pages, such as a brush. This is stored in a Remote Resource  
13 Dictionary part. For more information, see §14.2.3.1, "Remote Resource  
14 Dictionaries," on page 177.

### 15 **9.1.9 PrintTicket Parts**

16 **PrintTicket parts** provide user intent and device configuration information to printing  
17 consumers. PrintTicket parts MUST be processed when the XPS Document is printed  
18 [M2.58]. PrintTicket parts can be attached only to FixedDocumentSequence,  
19 FixedDocument and FixedPage parts and each of these parts MUST attach no more  
20 than one PrintTicket [M2.59]. PrintTickets can provide override settings to be used  
21 when printing the part to which they are attached.

#### 22 **9.1.9.1 PrintTicket Format**

23 The PrintTicket is XML that provides print settings in a consistent, accessible, and  
24 extensible manner. Valid PrintTicket settings are specified in the Print Schema.  
25 Within the context of an XPS Document, the PrintTicket is generated by the  
26 producer. Producers should note that an XPS Document may be printed on various  
27 devices, and that the settings included in the PrintTicket SHOULD support portability  
28 [S2.21]. Producers and consumers should note that not all PrintTicket keywords  
29 defined in the Print Schema are applicable to XPS Documents.

#### 30 **9.1.9.2 Mapping PrintTicket Parts to Fixed Payload Parts**

31 The PrintTicket defines a hierarchy of print settings to identify the applicability of a  
32 setting to different content levels within a print job. Specifically, the PrintTicket  
33 supports a hierarchy that is rooted in the print job. Multiple documents are derived  
34 from the print job, and multiple pages are defined from each document. A PrintTicket  
35 can be associated with each level in this hierarchy. These PrintTicket parts are  
36 labeled "job-level," "document-level," and "page-level," respectively.

37 Print settings are defined within the PrintTicket. Each print setting has a scoping  
38 prefix to indicate the level at which it applies. [Example: "JobInputBin" is a job-level  
39 setting, "DocumentStaple" is a document-level setting, and "PageOrientation" is a  
40 page-level setting. end example] Settings in level-specific PrintTicket parts are  
41 restricted by the scoping prefix. A level-specific PrintTicket MUST contain only  
42 settings scoped to the current level and child levels [M2.59]. Job-level PrintTicket  
43 parts MUST contain only job-, document-, and page-scoped settings; document-level

1 PrintTicket parts MUST contain only document-scoped and page-scoped settings; and  
 2 page-level PrintTicket parts MUST contain only page-scoped settings [M2.59].

3 Within an XPS Document, there is a direct mapping between the PrintTicket levels  
 4 and the fixed payload parts:

- 5 • Consumers MUST process job-level, document-level and page-level settings of  
 6 PrintTicket parts associated with FixedDocumentSequence parts [M2.60].
- 7 • Producers SHOULD only attach PrintTicket parts containing only document-  
 8 level and page-level settings with FixedDocument parts [S2.22].
- 9 • Consumers MUST process document-level and page-level settings of  
 10 PrintTicket parts associated with FixedDocument parts and MUST ignore job-  
 11 level settings of PrintTicket parts associated with FixedDocument parts  
 12 [M2.61].
- 13 • Producers SHOULD only attach PrintTicket parts containing only page-level  
 14 settings with FixedPage parts [S2.23].
- 15 • Consumers MUST process page-level settings of PrintTicket parts associated  
 16 with FixedPage parts and MUST ignore job-level and document-level settings  
 17 of PrintTicket parts associated with FixedPage parts [M2.62].

18 PrintTicket parts are associated with parts via the PrintTicket relationship defined in  
 19 H, "Standard Namespaces and Content Types."

#### 20 **9.1.9.3 Processing PrintTicket Parts**

21 Printing consumers MUST process all PrintTicket parts within the XPS Document  
 22 [M2.58].

23 When processing a PrintTicket, consumers MUST first remove all levels of PrintTicket  
 24 content not applicable to the current element [M2.63] (see §9.1.9.2, "Mapping  
 25 PrintTicket Parts to Fixed Payload Parts," on page 36).

26 Second, consumers MUST validate the PrintTicket according to the methods defined  
 27 in the PrintTicket Validation Checklist of the Print Schema documentation [M2.64].  
 28 Following validation, the printing consumer MUST properly interpret the print settings  
 29 according to the following rules for merging two PrintTicket parts [M2.65].

30 Print settings are expressed by scoped Print Schema elements. Elements may  
 31 interact between different levels in the PrintTicket hierarchy. Elements that interact  
 32 between levels MUST be specified at the root of each level ticket [M2.59]. A keyword  
 33 merge conflict between PrintTicket settings is defined as the same root-level Print  
 34 Schema element denoted by the same **name** attribute value appearing in multiple  
 35 level tickets. There are two options for interactions:

36 ~~3.~~ 3. If there is no merge conflict, a prefix-scoped element MUST be pushed  
 37 or inherited, from a more general ticket to a more specific ticket [M2.66]. This  
 38 case is isomorphic to the case where both tickets contain an identical element.

39 ~~2.~~ 4. If there is a merge conflict, the setting from the most specific ticket  
 40 precedence [M2.67]. That is, a page-scoped setting in a page-level PrintTicket  
 41 overwrites an identical page-scoped setting in a document-level or job-level  
 42 PrintTicket. Likewise, a document-scoped setting in a document-level  
 43 PrintTicket overwrites an identical document-scoped setting in a job-level  
 44 PrintTicket.

- 1 To determine the print settings in the XPS Document the following algorithm should  
2 be applied:
- 3 1. Validate the job-level PrintTicket associated with the fixed document sequence  
4 by merging and validating the PrintTicket with the default PrintTicket for the  
5 print consumer. The default PrintTicket represents the default configuration of  
6 the print consumer's state; in the case of a print consumer, the state is the  
7 current device and print driver configuration. Call the resulting ticket the  
8 "validated job-level PrintTicket." If no job-level PrintTicket is supplied, use the  
9 default PrintTicket.
  - 10 2. For each FixedDocumentX referenced by the fixed document sequence,  
11 perform the following steps:
    - 12 a. Merge and validate the document-level PrintTicket associated with  
13 FixedDocumentX with the validated job-level PrintTicket from Step 1.
    - 14 b. Call the resulting ticket the "validated DocumentX-level PrintTicket." If no  
15 document-level PrintTicket is supplied for the current fixed document, the  
16 validated job-level PrintTicket from Step 1 should be used.
  - 17 3. For each FixedPageY referenced by each FixedDocumentX in the fixed  
18 document sequence, perform the following steps:
    - 19 a. Merge and validate the page-level PrintTicket associated with FixedPageY  
20 with the validated DocumentX-level PrintTicket from Step 2.
    - 21 b. Call the resulting ticket the "validated PageXY-level PrintTicket." If no page-  
22 level PrintTicket is supplied for the current fixed page, the validated  
23 DocumentX-level PrintTicket from Step 2 should be used.

#### 24 **9.1.10 SignatureDefinitions Part**

25 Producers MAY add digital signature requests and instructions to an XPS Document in  
26 the form of signature definitions [O2.23]. A producer MAY sign against an existing  
27 signature definition to provide additional signature information [O2.24]. A recipient  
28 of the document MAY also sign the XPS Document against a signature definition  
29 [O2.25]. (This is referred to as "co-signing.")

30 Digital signature definitions are stored in a SignatureDefinitions part. A  
31 FixedDocument part refers to a SignatureDefinitions part using a relationship of the  
32 SignatureDefinitions type. For more information, see H, "Standard Namespaces and  
33 Content Types."

34 The SignatureDefinitions part is OPTIONAL [O2.6]. Signature definitions MUST  
35 conform to the Signature Definitions schema as defined in A [M2.72].

36 For more information on digital signature support in XPS Documents, see 17, "XPS  
37 Document Package Features."

#### 38 **9.1.11 DocumentStructure Part**

39 Explicitly authored document structure information is stored in the  
40 DocumentStructure part. This part contains the document outline and defines the  
41 framework for every element in fixed pages in terms of semantic blocks called  
42 stories. Stories are split into StoryFragments parts, which contain content structure  
43 markup that defines semantic blocks such as paragraphs and tables. For more  
44 information, see 16, "Document Structure and Interactivity."



1 Document structure markup contains a root <DocumentStructure> element. See 16,  
2 "Document Structure and Interactivity," for markup details. The  
3 <DocumentStructure> element uses the Document Structure namespace specified  
4 in §H.1, "XML Namespace URIs," on page 405.

5 The DocumentStructure part is referenced by relationship from the FixedDocument  
6 part, with the relationship type as specified in H, "Standard Namespaces and Content  
7 Types." The content type of the DocumentStructure part is also specified in H.

8 Consumers MAY provide an algorithmic construction of the structure of an XPS  
9 Document based on a page-layout analysis [O2.27], but they MUST NOT use such a  
10 method to derive structure for any part of the XPS Document included in the  
11 DocumentStructure part [M2.68]. A consumer capable of calculating reading order  
12 from the layout of the document MUST use the reading order specified in the  
13 DocumentStructure part, even though the derived order may be perceived as  
14 preferable to the specified order [M2.68].

#### 15 **9.1.12 StoryFragments Part**

16 The StoryFragments part contains content structure markup (such as tables and  
17 paragraphs) associated with a single fixed page.

18 StoryFragments part markup contains a root <StoryFragments> element. See 16,  
19 "Document Structure and Interactivity," for markup details. The <StoryFragments>  
20 element uses the Document Structure namespace specified in §H.1, "XML  
21 Namespace URIs," on page 405.

22 The StoryFragments part is referenced by relationship from its associated FixedPage  
23 part. The content type of the StoryFragments part is specified in §H.2, "Content  
24 Types," on page 406.

## 9.2 Part Naming Recommendations

Producers and consumers of XPS Documents refer to parts by name and use relationship names to identify the purpose of related parts. The OPC specification describes the syntax for part names. However, following these rules alone can result in a package that is difficult for users to understand. [Example: A user would have to open every Relationship part to know which parts are necessary to accurately render an XPS Document. end example]

By choosing part names according to a well-defined, human-readable convention, the resulting package is easier to browse and specific parts are more easily located. Part names MUST still conform to the syntax specified in the OPC specification [M1.1].

It is RECOMMENDED that producers of XPS Documents use the following part naming convention:

- The FixedDocumentSequence part name SHOULD contain only one segment with the extension ".fdseq". [Example: "/FixedDocSeq.fdseq" end example] [S2.24].
  - A FixedDocument part name SHOULD contain three segments, using "/Documents/n/" in the first two segments and the extension ".fdoc" [S2.25]. Here, *n* SHOULD be a numeral that represents the ordinal position of the fixed document in the fixed document sequence [S2.25]. [Example: The fixed document referenced by the **Source** attribute of the third <DocumentReference> child of the <FixedDocumentSequence> element could be "/Documents/3/FixedDocument.fdoc". end example]
  - A FixedPage part name SHOULD contain four segments, using "/Documents/n/Pages/" as the first three segments and the extension ".fpage" on the last segment [S2.26]. Here, *n* represents the fixed document that includes this page. [Example: The third page of the second document might be "/Documents/2/Pages/3.fpage". end example]
  - Resource parts MAY be named to indicate whether their intended use is at the document level or as a shared resource for all documents [O2.28]. A resource that is specific to a particular document SHOULD have a part name that begins with the three segments "/Documents/n/Resources/" where *n* is the particular fixed document [S2.27]. A resource intended to be shared across documents SHOULD begin with the segment "/Resources/" and SHOULD have a final segment that is a globally unique identifier followed by the appropriate extension for that resource [S2.27]. [Example: "/Resources/63B51F81-C868-11D0-999C-00C04FD655E1.odttf" end example]
- A Font part name SHOULD append the segment "Fonts/" to the resource part name prefix specified above [S2.27]. [Example: A font might be named "/Documents/1/Resources/Fonts/Arial.ttf" or "/Resources/F2ABC7B7-C60D-4FB9-AAE4-3CA0F6C7038A.odttf". end example]
- An Image part name SHOULD append the segment "Images/" to the resource part name specified above [S2.27]. [Example: An image might be named "/Documents/3/Resources/Images/dog.jpg" or "/Resources/Images/E0D79307-846E-11CE-9641-444553540000.jpg". end example]

**Comment [rcj7]:** Is a comma missing here? That is, should there be only 1 segment, and that segment's has an extension of .fdseq? Or can there be more than 1 segment, but only 1 with that extension?

- 1 A Remote Resource Dictionary part name SHOULD append the segment  
2 "Dictionaries/" to the resource part name specified above [S2.27]. Remote  
3 resource dictionaries SHOULD also use the ".dict" extension [S2.27].  
4 [Example: A resource dictionary might be named  
5 "/Documents/2/Resources/Dictionaries/Shapes.dict" or  
6 "/Resources/0DDF3BE2-E692-15D1-AB06-B0AA00BDD685.dict". end  
7 example]
- 8 • Any DocumentStructure part name SHOULD contain four segments using  
9 "/Documents/n/Structure/" as the first three segments and the extension  
10 ".struct" [S2.28]. Here *n* represents the fixed document that this structure is  
11 associated with. [Example: The DocumentStructure part for the first  
12 document in a fixed document sequence could be  
13 "/Documents/1/Structure/DocStructure.struct". end example]
  - 14 • Any StoryFragments part name SHOULD contain five segments using  
15 "/Documents/n/Structure/Fragments" as the first four segments and the  
16 extension ".frag" [S2.29]. Here *n* represents the fixed document that these  
17 parts are associated with. [Example: A StoryFragment part associated with  
18 the third page of the second document in a fixed document sequence could be  
19 "/Documents/2/Structure/Fragments/3.frag". end example]
  - 20 • ICC profile part names SHOULD contain four segments, using  
21 "/Documents/n/Metadata/" as the first three segments, where *n* is the fixed  
22 document that uses these parts [S2.30]. If an ICC profile part is shared  
23 across documents, the part name SHOULD contain two segments, using  
24 "/Metadata/" as the first segment and a second segment that is a string  
25 representation of a globally unique identifier, followed by an extension  
26 [S2.30]. ICC profiles SHOULD use an appropriate extension for the color  
27 profile type. [S2.30] [Example: ".icm" end example]
  - 28 • Thumbnail part names SHOULD contain four segments, using  
29 "/Documents/n/Metadata/" as the first three segments, where *n* is the fixed  
30 document that uses the thumbnail [S2.31]. If the Thumbnail part relates to  
31 the package as a whole, the part name SHOULD contain two segments, using  
32 "/Metadata/" as the first segment and a second segment that is a string  
33 representation of a globally unique identifier, followed by an extension  
34 [S2.31]. Thumbnails SHOULD use an extension appropriate to the image  
35 type, either ".png" or ".jpg" [S2.31]. [Example: A Thumbnail part for a  
36 particular fixed page might be "/Documents/1/Metadata/5.png". end  
37 example]
  - 38 • PrintTicket part names associated with the entire job SHOULD be associated  
39 via relationship with the FixedDocumentSequence part and contain two  
40 segments, using "/Metadata/" as the first segment [S2.32]. PrintTicket parts  
41 associated with a particular fixed document or fixed page SHOULD contain  
42 four segments, using "/Documents/n/Metadata/" as the first three segments,  
43 where *n* is the fixed document that uses these parts [S2.32]. PrintTicket parts  
44 SHOULD use the extension ".xml" [S2.32]. [Example: A PrintTicket associated  
45 with the entire job could be "/Metadata/Job\_PT.xml" and a PrintTicket  
46 associated with a single page might be  
47 "/Documents/1/Metadata/Page2\_PT.xml". end example]
  - 48 • The names of any non-standard parts that are associated with a particular  
49 fixed document SHOULD contain four segments, using "/Documents/n/Other/"

1 as the first three segments. Here,  $n$  is the fixed document to which the part  
2 belongs [S2.33].

3 *Example 9–2. XPS Document part naming*

4 An XPS Document that contains two FixedDocument parts is represented as follows:

```
5 /FixedDocSeq.fdseq
6 /Documents/1/FixedDocument.fdoc
7 /Documents/1/Pages/1.fpage
8 /Documents/1/Pages/2.fpage
9 /Documents/1/Resources/Fonts/FontA.ttf
10 /Documents/1/Resources/Images/ImageB.jpg
11 /Documents/1/Metadata/Document_PT.xml
12 /Documents/1/Metadata/Page5_PT.xml
13 /Documents/1/Structure/DocStructure.struct
14 /Documents/1/Structure/Fragments/1.frag
15 /Documents/1/Structure/Fragments/2.frag
16 /Documents/1/Other/FabrikamIncBusinessAccount.xml
17 /Documents/2/FixedDocument.fdoc
18 /Documents/2/Pages/1.fpage
19 /Documents/2/Resources/Fonts/FontB.ttf
20 /Documents/2/Resources/Images/ImageA.png
21 /Documents/2/Metadata/ColorProfile.icm
22 /Documents/2/Metadata/Document_PT.xml
23 /Documents/2/Other/FabrikamIncInsuranceInfo.xml
24 /Metadata/Job_PT.xml
25 /Resources/63B51F81-C868-11D0-999C-00C04FD655E1.ttf
```

26 *end example]*

---

### 27 9.3 XPS Document Markup

28 XPS Document markup has been designed to facilitate the independent development  
29 of compatible systems that produce or consume XPS Documents. It also shares  
30 concepts with portions of the Microsoft .NET Framework 3.0 programming platform.

31 The graphics rendering model is shared with that of the Windows Presentation  
32 Foundation, assuring fidelity between on-screen display and printed output. The  
33 syntax of fixed page, fixed document, and fixed document sequence markup is  
34 compatible with that of Windows Presentation Foundation XAML. The elements,  
35 attributes, and attribute values are a subset of those defined by the Windows  
36 Presentation Foundation.

37 The relationship between XPS Document markup and .NET 3.0 technologies does not  
38 impose any requirement on system implementations. Support for XPS Document  
39 markup does not require incorporation of .NET 3.0, the Windows Presentation  
40 Foundation, or managed code. However, the relationship with .NET 3.0 technologies  
41 allows producers to extend XPS Document markup for further use in the Windows  
42 Presentation Foundation framework, such as by including additional presentation  
43 features.

44 The design of XPS Document markup reflects the tradeoffs between two, sometimes  
45 competing, goals:

46 1. XPS Document markup should be parsimonious; that is, it should include only  
47 the minimum set of primitive operations and markup constructs necessary to  
48 render text and graphics with full fidelity. Redundancy in the specification

1 increases the opportunity for independent implementations, such as printer-  
2 resident raster image processors (RIPs), viewers, and interactive applications,  
3 to introduce accidental incompatibilities. Redundancy also increases the cost of  
4 implementation and testing, and, typically, the required memory footprint.

5 2. XPS Document markup should be compact; that is, the most common  
6 graphical primitives for vector graphics and text-rendering should have  
7 compact representations. Bloated representations compromise the  
8 performance of systems handling XPS Documents. As byte-count increases, so  
9 does communication time. Although compression can be used to improve  
10 communication time, it cannot eliminate the performance loss caused by  
11 bloated representations.

### 12 **9.3.1 Support for Versioning and Extensibility**

13 XPS Document markup has been designed in anticipation of the evolution of this  
14 specification. It also allows third parties to extend the markup. XPS Document  
15 markup incorporates the Markup Compatibility specification incorporated by the OPC  
16 specification.

17 The following parts MAY include elements and attributes defined in the Markup  
18 Compatibility specification [O2.29]:

- 19 • DocumentStructure
- 20 • FixedDocument
- 21 • FixedDocumentSequence
- 22 • FixedPage
- 23 • Relationships
- 24 • Remote Resource Dictionary
- 25 • SignatureDefinitions
- 26 • StoryFragments

27 Consumers of these parts MUST support the Markup Compatibility specification  
28 [M2.69]. Before attempting to validate one of these parts against a schema,  
29 processors MUST remove all markup compatibility elements and attributes and all  
30 ignorable elements and attributes not defined in the expected version of XPS  
31 Document markup [M2.69].

32 [*Note: Markup compatibility elements and attributes that appear in one XPS*  
33 *Document part do not carry through to a second part via an inline URI reference in*  
34 *the XML markup. Likewise the markup compatibility mechanisms do not carry*  
35 *through from part to part via relationship. end note]*

### 36 **9.3.2 XML Usage**

37 All XML content of the parts defined in this specification MUST conform to the  
38 following validation rules:

- 39 1. XML content MUST be encoded using either UTF-8 or UTF-16. If any such part  
40 includes an encoding declaration (as defined in §4.3.3 of the XML  
41 specification), that declaration MUST NOT name any encoding other than UTF-  
42 8 or UTF-16 [M2.70].

- 1       2. The XML 1.0 specification allows for the usage of Data Type Definitions  
2       (DTDs), which enable Denial of Service attacks, typically through the use of an  
3       internal entity expansion technique. As mitigation for this potential threat,  
4       DTD content MUST NOT be used in the XML markup defined in this  
5       specification, and consumers MUST treat the presence of DTD content as an  
6       error [M2.71].
- 7       3. If the XML content contains the Markup Compatibility namespace, as described  
8       in the Markup Compatibility specification, it MUST be processed to remove  
9       Markup Compatibility elements and attributes, ignorable namespace  
10       declarations, and ignored elements and attributes before applying further  
11       validation rules below [M2.69].
- 12       4. XML content MUST be valid against the corresponding XSD schema defined in  
13       this specification. In particular, the XML content MUST NOT contain elements  
14       or attributes drawn from namespaces that are not explicitly defined in the  
15       corresponding XSD unless the XSD allows elements or attributes drawn from  
16       any namespace to be present in particular locations in the XML markup  
17       [M2.72].
- 18       5. XML content MUST NOT contain elements or attributes drawn from "xml" or  
19       "xsi" namespaces unless they are explicitly defined in the XSD schema or by  
20       other means in the specification [M2.73].

### 21   **9.3.3 Markup Model**

22   XPS Document markup is an XML-based markup language that uses elements,  
23   attributes, and namespaces. The schema for XPS Document markup includes only  
24   elements and their attributes, comments, and whitespace. Arbitrary character data  
25   intermingled in the markup is not allowed.

26   Fixed page markup is expressed using elements and attributes and is based on a  
27   higher-level abstract model of contents and properties. Some fixed page elements  
28   can hold "contents," which are expressed as child elements. Properties may be  
29   expressed either as attributes or child elements.

30   XPS Document markup also uses resources and resource dictionaries, which allow  
31   elements to share property values.

#### 32   **9.3.3.1 Namespaces**

33   The following XML namespaces are defined for use in XPS Document markup:

- 34       • The XPS Document namespace, the principal namespace used for elements  
35       and attributes in fixed page markup. For more information, see H, "Standard  
36       Namespaces and Content Types."
- 37       • The Resource Dictionary Key namespace, which allows certain XPS Document  
38       elements to be included in a resource dictionary, as described in §14.2,  
39       "Resources and Resource References," on page 172.
- 40       • The Markup Compatibility namespace, which supports the Markup  
41       Compatibility specification as defined in the OPC specification.

### 1 **9.3.3.2 Properties**

2 A **property** is a characteristic of an element. XPS Document property values can be  
 3 expressed either as property attributes or property elements. **Property values** may  
 4 be stored in a resource dictionary and referenced by an attribute that uses a special  
 5 syntax to express its value. For more information, see §14.2, "Resources and  
 6 Resource References," on page 172.

7 Properties MUST NOT be set more than once, regardless of the syntax used to  
 8 specify the value [M2.74]. In certain cases, they can be specified using either  
 9 property attributes or property elements. Consumers MUST treat properties that are  
 10 specified in both ways as an error [M2.74].

11 Some properties are common to several fixed page elements. For more information,  
 12 see 14, "Common Properties."

#### 13 **9.3.3.2.1 Composable Property Values**

14 Some fixed page properties are composable, meaning that the page marking effect is  
 15 determined by combining the property value of a given element with that of its  
 16 parent and ancestor elements. [Example: A <Path> element with an **Opacity** value  
 17 of 0.5 nested inside a <Canvas> element with an **Opacity** value of 0.5 results in an  
 18 effective 25% opacity of the <Path> element when rendered. *end example*]

19 The coordinate space used to render page marking elements is also composable. By  
 20 default, elements are rendered in a coordinate space with units of 1/96". The  
 21 **effective coordinate space** for a particular element is created by sequentially  
 22 applying each parent and ancestor element's affine matrix transformation, specified  
 23 with the Transform or RenderTransform properties, from outermost to innermost,  
 24 including the element's own affine matrix transformation.

25 For more information, see §18.4.1, "Pre-Multiplied Alpha and Superluminous Colors,"  
 26 on page 281.

#### 27 **9.3.3.2.2 Property Attribute Syntax**

28 Some property values can be expressed using simple XML attribute syntax, that is,  
 29 with a text string. The value of properties used to describe geometries may be  
 30 expressed using an abbreviated syntax. For more information, see §11.2.3,  
 31 "Abbreviated Geometry Syntax," on page 90.

32 *Example 9-3. Property attribute syntax*

33 The following syntax can be used to specify the color of a brush:

```
34 <!-- Property Attribute Syntax -->  
35 <SolidColorBrush Color="#FF0000" />
```

36 *end example*]

#### 37 **9.3.3.2.3 Property Element Syntax**

38 Some property values can also be expressed using a child element to describe the  
 39 property value. These property elements are included to enable usage of the Markup  
 40 Compatibility mechanisms described in the Markup Compatibility specification. The  
 41 element name is derived from a combination of a parent element name and the  
 42 property name, separated by a dot (.) character.

- 1 The order of child property elements is significant: they MUST occur before any
- 2 contents of the parent element and they MUST appear in the sequence specified in
- 3 the schema [M2.72].



1 *Example 9-4. Property element syntax*

2 When specifying Clip and RenderTransform properties of the canvas, both must  
3 appear before any path and glyphs contents of the canvas.

```
4 <Canvas>  
5 <!-- First, the property-related child elements -->  
6 <Canvas.RenderTransform>  
7 <MatrixTransform Matrix="1,0,0,1,0,0" />  
8 </Canvas.RenderTransform>  
9 <Canvas.Clip>  
10 <PathGeometry>  
11 ...  
12 </PathGeometry>  
13 </Canvas.Clip>  
14 <!-- Then, the "contents" -->  
15 <Path ...>  
16 ...  
17 </Path>  
18 <Glyphs ... />  
19 </Canvas>
```

20 *end example]*

### 21 9.3.4 Whitespace

22 XPS Documents allow flexible whitespace usage in markup. Wherever a single  
23 whitespace character is allowed, multiple whitespace characters MAY be used  
24 [O2.30]. Attributes that specify comma-delimited attribute values MAY, unless  
25 specified otherwise, OPTIONALLY include whitespace characters preceding or  
26 following the comma [O2.31]. XPS Document markup MUST NOT use the **xml:space**  
27 attribute [M2.75]. Additionally, where the XPS Document schema specifies attributes  
28 of types that allow whitespace collapsing, leading and trailing whitespace in the  
29 attribute value MAY be used along with other whitespace that relies on the  
30 whitespace collapsing behavior specified in the XML Schema Specification [O2.32].

31 [*Note*: Consult the XPS Document Schema for exact whitespace allowed. *end note*]

### 32 9.3.5 Language

33 Language information supports the following features:

- 34 • Language-dependent find features
- 35 • Selection of a text-to-speech dictionary by a screen-reading program (to  
36 provide accessibility to persons with disabilities)
- 37 • Selection of a spelling checker for text copied to another document
- 38 • Selection of a grammar checker for text copied to another document
- 39 • Correct font rendering when copying the text to another document

40 The last point refers to instances in which multiple languages share the same script.  
41 [*Example*: The Devanagari script is shared by the Indic languages Bhojpuri, Bihari,  
42 Hindi, Kashmiri, Konkani, Marathi, Nepali, and Sanskrit. However, these languages  
43 render certain glyph sequences differently. When text is copied from an XPS  
44 Document, the language of the copied characters is needed to ensure proper  
45 rendering of the glyphs when they are pasted into another application. This scenario

1 applies to most Indic-language fonts, some East Asian-language fonts, and others.  
2 *end example*]

### 3 **9.3.5.1 xml:lang Attribute**

4 The language of the contents of an XPS Document MUST be identified using the  
5 **xml:lang** attribute, the value of which is inherited by child and descendant elements  
6 [M2.76]. This attribute is defined in the W3C XML specification.

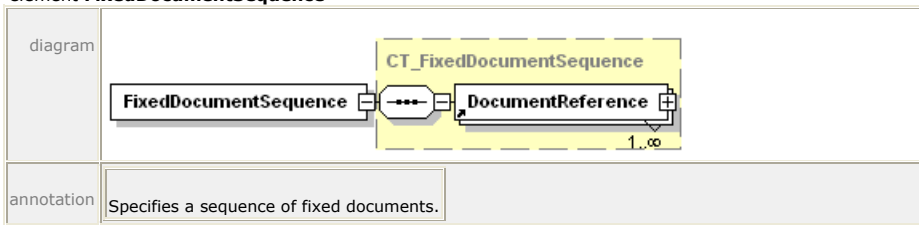
7 **xml:lang** is REQUIRED for <FixedPage> elements and MAY be used with <Canvas>,  
8 <Path>, and <Glyphs> elements; it is not valid on any other fixed page markup  
9 element [M2.72]. **xml:lang** is also REQUIRED for the <DocumentOutline> element  
10 for document structure and OPTIONAL for the <OutlineEntry> element [M2.72].  
11 When the language of the contents is unknown and is required, the value "und"  
12 (undetermined) MUST be used [M2.76].

## 10. Documents

XPS Documents contain a root fixed document sequence that binds a collection of fixed documents which, in turn, bind a collection of fixed pages. All page markings are specified with <Glyphs> or <Path> elements on the fixed page. These elements can be grouped within one or more <Canvas> elements. Page markings are positioned by real-number coordinates in the coordinate space of the fixed page. The coordinate space can be altered by applying a render transformation.

### 10.1 <FixedDocumentSequence> Element

element **FixedDocumentSequence**



The <FixedDocumentSequence> element contains one or more <DocumentReference> elements. The order of <DocumentReference> elements MUST match the order of the documents in the fixed document sequence [M3.1].

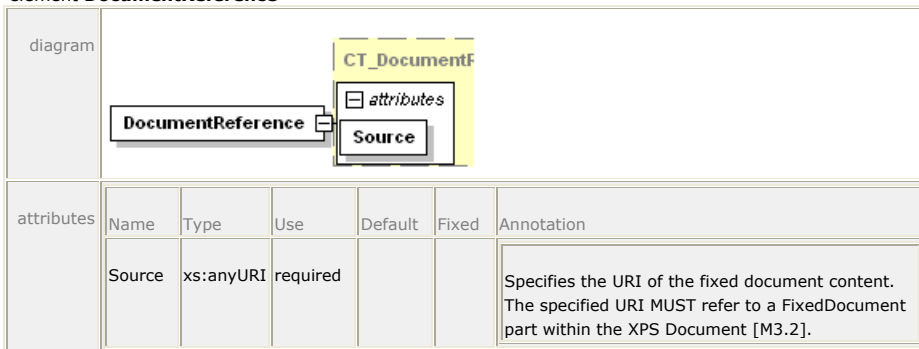
Example 10-1. <FixedDocumentSequence> usage

```
<FixedDocumentSequence xmlns="http://schemas.microsoft.com/xps/2005/06">
  <DocumentReference Source="Documents/1/FixedDocument.fdoc" />
  <DocumentReference Source="Documents/2/FixedDocument.fdoc" />
</FixedDocumentSequence>
```

end example]

#### 10.1.1 <DocumentReference> Element

element **DocumentReference**

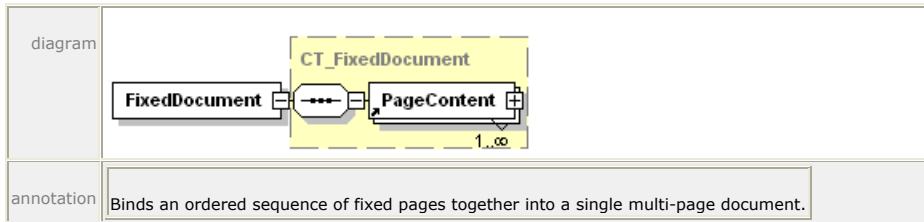


annotation	Contains a reference to a FixedDocument part.
------------	---

1 The <DocumentReference> element specifies a FixedDocument part as a URI in the  
 2 **Source** attribute. Producers MUST NOT produce a document with multiple  
 3 <DocumentReference> elements that reference the same fixed document [M3.3].

## 4 10.2 <FixedDocument> Element

5 element **FixedDocument**



6 The <FixedDocument> element contains one or more <PageContent> elements. The  
 7 order of <PageContent> elements MUST match the order of the pages in the  
 8 document [M3.4].

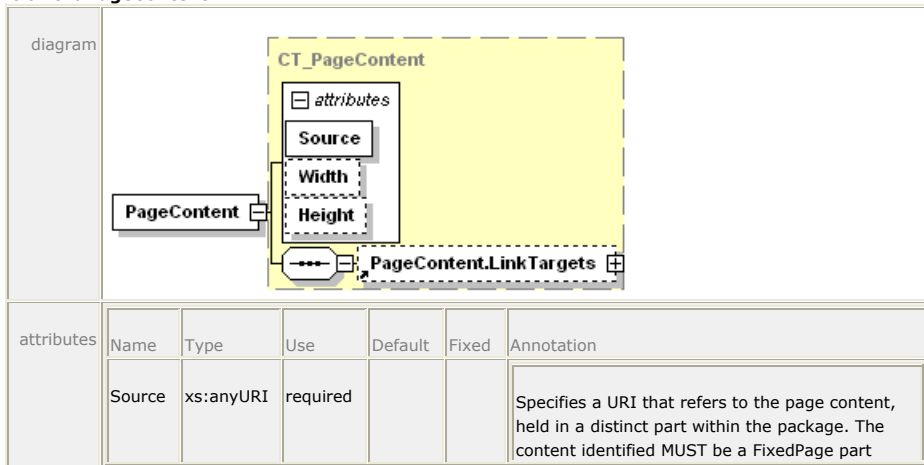
9 *Example 10-2. <FixedDocument> usage*

```
10 <FixedDocument xmlns="http://schemas.microsoft.com/xps/2005/06">
11   <PageContent Source="Pages/1.fpage" />
12   <PageContent Source="Pages/2.fpage" />
13 </FixedDocument>
```

14 *end example]*

### 15 10.2.1 <PageContent> Element

16 element **PageContent**



					within the XPS Document [M3.5].
Width	ST_GEOne				Typical width of pages contained in the page content.
Height	ST_GEOne				Typical height of pages contained in the page content.
annotation	Defines a reference from a fixed document to a part that contains a <FixedPage> element.				

- 1 Each <PageContent> element refers to the source of the content for a single page.
- 2 The number of pages in the document can be determined by counting the number of
- 3 <PageContent> elements.
  
- 4 The <PageContent> element has a single required attribute, **Source**, which refers to
- 5 a FixedPage part. It may optionally include advisory **Height** and **Width** attributes to
- 6 indicate the size of a single page. (The authoritative height and width are specified
- 7 by the fixed page.) The **Height** and **Width** attribute values allow consumers such as
- 8 viewers to make initial visual layout estimates quickly, without loading and parsing
- 9 all of the individual fixed pages. These consumers then update the page dimensions
- 10 when the fixed page is loaded, if they differ.
  
- 11 The <PageContent> element has one allowable child element,
- 12 <PageContent.LinkTargets>, and it MUST NOT contain more than a single child
- 13 element [M2.72].
  
- 14 Producers MUST NOT produce markup where a <PageContent> element references
- 15 the same fixed page referenced by any other <PageContent> element in the entire
- 16 XPS Document, even in other fixed documents within the fixed payload [M3.6].

17 *Example 10-3. <PageContent> usage*

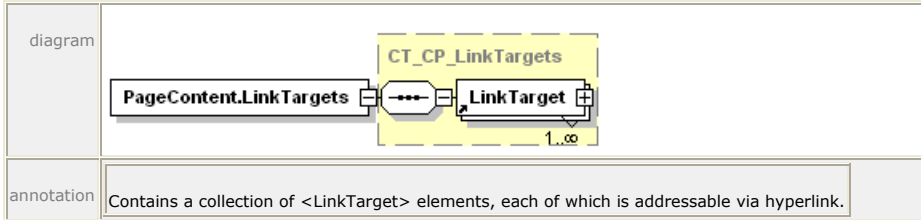
```

18 <FixedDocument xmlns="http://schemas.microsoft.com/xps/2005/06">
19 <PageContent Source="Pages/1.fpage" Height="1056" Width="816" />
20 <PageContent Source="Pages/2.fpage" Height="1056" Width="816" />
21 </FixedDocument>
    
```

22 *end example]*

23 **10.2.2 <PageContent.LinkTargets> Element**

24 element **PageContent.LinkTargets**



- 25 The <PageContent.LinkTargets> element defines the list of link targets that specify
- 26 each named element on the page that may be addressed by hyperlink.

1 *Example 10-4. <PageContent.LinkTargets> usage*

2 In the following markup, *Pages/2.fpage* contains two <LinkTarget> elements with  
 3 **Name** attribute values of *Anchor1* and *Anchor2*:

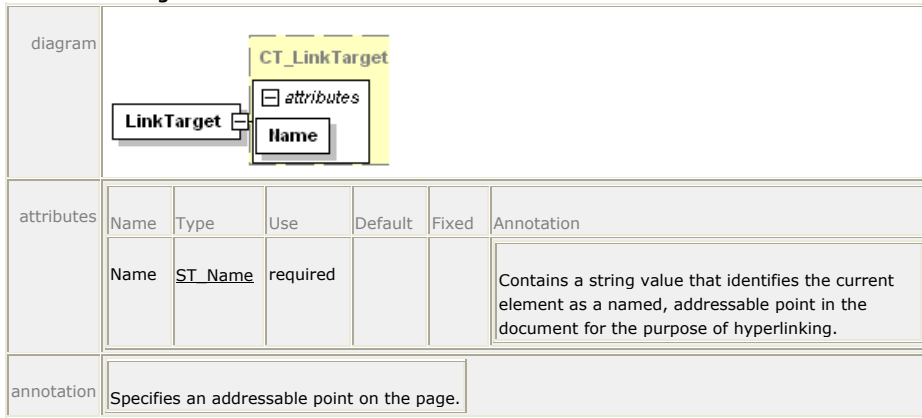
```

4 <FixedDocument xmlns="http://schemas.microsoft.com/xps/2005/06">
5 <PageContent Source="Pages/1.fpage" Height="1056" Width="816" />
6 <PageContent Source="Pages/2.fpage" Height="1056" Width="816">
7 <PageContent.LinkTargets>
8 <LinkTarget Name="Anchor1" />
9 <LinkTarget Name="Anchor2" />
10 </PageContent.LinkTargets>
11 </PageContent>
12 </FixedDocument>
    
```

13 *end example]*

14 **10.2.3 <LinkTarget> Element**

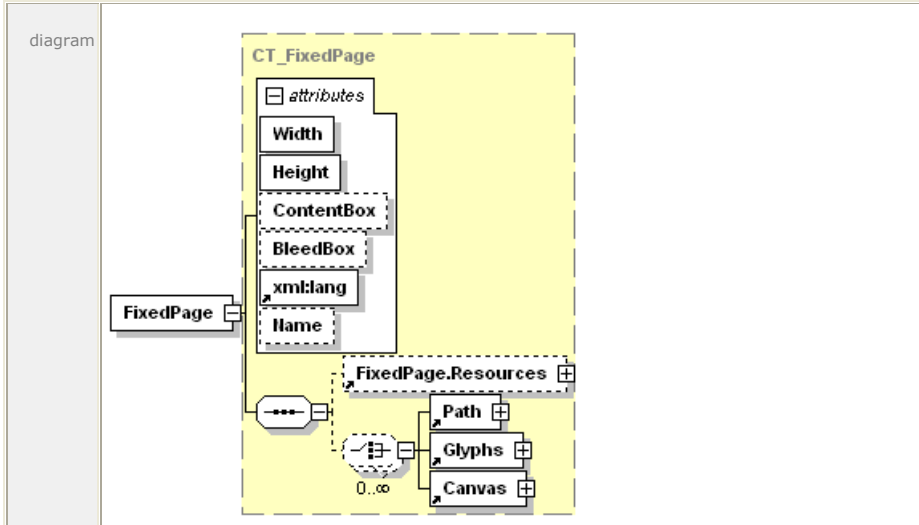
15 element **LinkTarget**



16 The <LinkTarget> element specifies a **Name** attribute, which corresponds to a  
 17 named location within the fixed page specified by its parent <PageContent>  
 18 element. By encapsulating this information in the fixed document, consumers do not  
 19 need to load every FixedPage part to determine if a particular **Name** value exists in  
 20 the document. For more information, see §16.2, "Hyperlinks," on page 243.

1 **10.3 <FixedPage> Element**

2 element **FixedPage**



attributes	Name	Type	Use	Default	Fixed	Annotation
	Width	<a href="#">ST_GEOne</a>	required			Width of the page, expressed as a real number in units of the effective coordinate space.
	Height	<a href="#">ST_GEOne</a>	required			Height of the page, expressed as a real number in units of the effective coordinate space.
	ContentBox	<a href="#">ST_ContentBox</a>				Specifies the area of the page containing imageable content that is to be fit within the imageable area when printing or viewing. Contains a list of four coordinate values (ContentOriginX, ContentOriginY, ContentWidth, ContentHeight), expressed as comma-separated real numbers. Specifying a value is RECOMMENDED [S3.1]. If omitted, the default value is (0,0,Width,Height).
	BleedBox	<a href="#">ST_BleedBox</a>				Specifies the area including crop marks that extends outside of the physical page. Contains a list of four coordinate values (BleedOriginX, BleedOriginY,

					BleedWidth, BleedHeight), expressed as comma-separated real numbers. If omitted, the default value is (0,0,Width,Height).
	xml:lang		required		Specifies the default language used for the current element and for any child or descendant elements. The language is specified according to RFC 3066.
	Name	<u>ST_Name</u>			Contains a string value that identifies the current element as a named, addressable point in the document for the purpose of hyperlinking.
annotation	Contains markup that describes the rendering of a single page of content.				

1 The <FixedPage> element contains the contents of a page and is the root element of  
 2 a FixedPage part. The fixed page contains the elements that together form the basis  
 3 for all markings rendered on the page: <Paths>, <Glyphs>, and the optional  
 4 <Canvas> grouping element.

5 The fixed page MUST specify a height, width, and default language [M2.72].

6 The coordinate space of the fixed page is composable, meaning that the marking  
 7 effects of its child and descendant elements are affected by the coordinate space of  
 8 the fixed page.

9 *Example 10-5. Fixed page markup*

```
10 <FixedPage Height="1056" Width="816" xml:lang="en-US"
11 xmlns="http://schemas.microsoft.com/xps/2005/06">
12 <Glyphs
13   OriginX="96"
14   OriginY="96"
15   UnicodeString="This is Page 1!"
16   FontUri="../Resources/Fonts/Times.TTF"
17   FontRenderingEmSize="16" />
18 </FixedPage>
```

19 *end example]*

### 20 10.3.1 BleedBox Attribute

21 The **BleedBox** attribute defines the area (inclusive of crop marks) that extends  
 22 outside of the physical page. The bleed box is expressed as four comma-separated,  
 23 real-number coordinate values that correspond to BleedOriginX, BleedOriginY,  
 24 BleedWidth, BleedHeight. These values are specified in units of 1/96".

25 Bleed boxes that do not satisfy the following conditions are invalid and SHOULD be  
 26 ignored in favor of the default bleed box [S3.2]:

- 27 • The **BleedBox** BleedOriginX value MUST be less than or equal to 0 [M3.7].
- 28 • The **BleedBox** BleedOriginY value MUST be less than or equal to 0 [M3.8].



- 1 • The **BleedBox** BleedWidth value MUST be greater than or equal to the fixed  
2 page **Width** attribute value plus the absolute value of the **Bleedbox**  
3 BleedOriginX value [M3.9].
- 4 • The **BleedBox** BleedHeight value MUST be greater than or equal to the fixed  
5 page **Height** attribute value plus the absolute value of the **BleedBox**  
6 BleedOriginY value [M3.10].

7 If the **BleedBox** attribute is omitted, the default value is "0,0,Width,Height".

### 8 10.3.2 ContentBox Attribute

9 The **ContentBox** attribute specifies the area of the page that contains imageable  
10 content that must fit in the imageable area when printing or viewing. Specifying this  
11 attribute is RECOMMENDED [S3.1]. The content box is expressed as four comma-  
12 separated, real-number coordinate values that correspond to ContentOriginX,  
13 ContentOriginY, ContentWidth, ContentHeight. These values are specified in units of  
14 1/96".

15 Content boxes that do not satisfy the following conditions are invalid and SHOULD be  
16 ignored in favor of the default content box [S3.3]:

- 17 • The **ContentBox** ContentOriginX value MUST be greater than or equal to 0  
18 and less than the fixed page **Width** attribute value [M3.11].
- 19 • The **ContentBox** ContentOriginY value MUST be greater than or equal to 0  
20 and less than the fixed page **Height** attribute value [M3.12].
- 21 • The **ContentBox** ContentWidth value MUST be less than or equal to the  
22 difference between the fixed page **Width** attribute value and the **ContentBox**  
23 ContentOriginX value [M3.13].
- 24 • The **ContentBox** ContentHeight value MUST be less than or equal to the  
25 difference between the fixed page **Height** attribute value and the  
26 **ContentBox** ContentOriginY value [M3.14].

27 If the **ContentBox** attribute is omitted, the default value is "0,0,Width,Height".

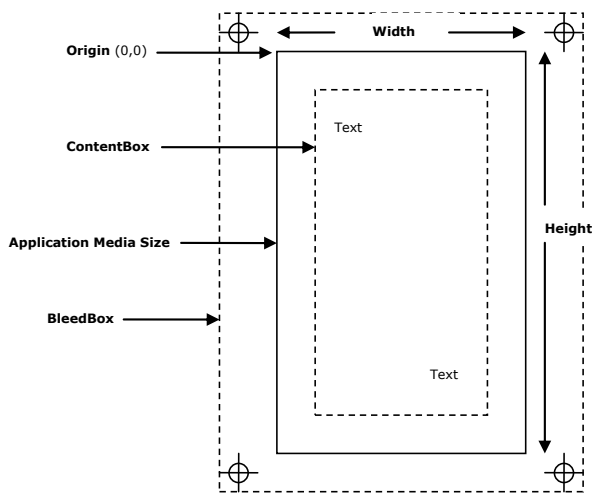
### 28 10.3.3 Page Size Terminology

29 The following terminology is used to describe page sizes:

- 30 • The **producer media size** represents the media size that is used by the  
31 producer for layout purposes. This size is described by the **Height** and **Width**  
32 attributes.
- 33 • The **producer bleed size** represents the overflow or "bleed" box used by the  
34 producer for registration and layout. This size is described by the **BleedBox**  
35 attribute. If the **BleedBox** attribute is not present, then the producer bleed  
36 size is defined as the producer media size.
- 37 • The **producer content size** represents the content bounding box specified by  
38 the producer. This size is described by the **ContentBox** attribute. If the  
39 **ContentBox** attribute is not present, then the producer content size is defined  
40 as the producer media size.
- 41 • The **physical media size** represents the physical media on which the content  
42 will be printed. This size is described by the PageMediaSize keyword in the

- 1 Print Schema. The PageMediaSize is page-orientation-agnostic. This means  
 2 that the representation of physical media size is defined by two dimensions,  
 3 PageMediaSizeWidth and PageMediaSizeHeight. For more information, see the  
 4 Print Schema specification.
- 5 • The **physical imageable size** represents the area that is printable by a specific  
 6 device. This size is described by the PageImageableSize keyword in the Print  
 7 Schema. The PageImageableSize is relative to the physical media size  
 8 (PageMediaSize keyword) and the orientation (PageOrientation keyword). For  
 9 more information, see the Print Schema specification.

10 Figure 10-1. Page regions



11

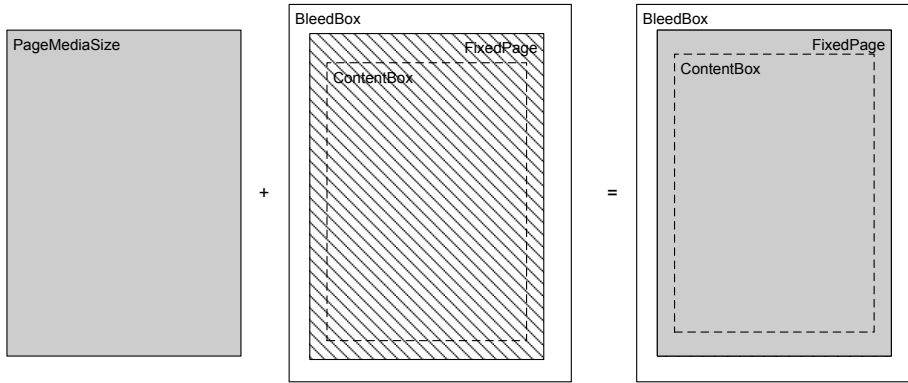
12 **10.3.4 Media Orientation and Scaling**

13 When rendering a fixed page for printing, consumers MUST be aware of the  
 14 interaction between the fixed page markup and the PrintTicket settings [M3.15]. The  
 15 interaction for media scaling is governed by the PageMediaSize, PageImageableSize,  
 16 PageScaling, and PageOrientation Print Schema keywords.

17 For media orientation, the PageMediaSize, PageOrientation and the width and height  
 18 of the <FixedPage> element determine the rendering of the fixed page. In the  
 19 absence of media scaling, the fixed page content is imaged directly to the physical  
 20 media with the origin of the fixed page aligned with the origin of the physical media  
 21 size. Any fixed page content that extends beyond the dimension of the physical  
 22 media size SHOULD be clipped [S3.4].

23 [Example: Consider a fixed page with a width of 816 (8.5") and a height of 1056  
 24 (11"). If the PrintTicket specifies the PageMediaSize value as Letter and the  
 25 PageOrientation value as Portrait then no clipping occurs, as shown below.

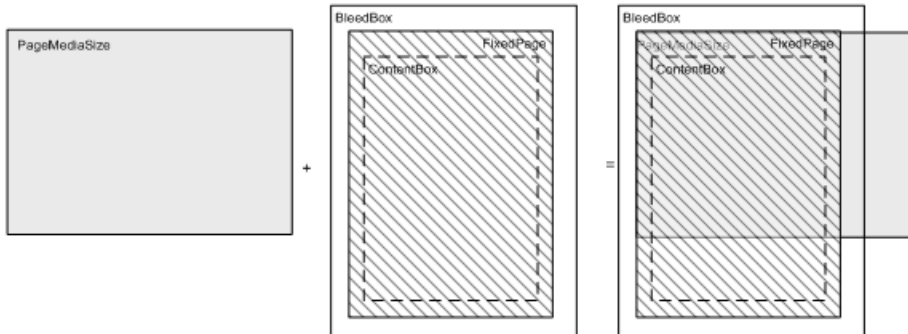
1 *Figure 10–2. Matching PrintTicket and fixed page size and orientation*



2

3 If the PrintTicket specifies the PageMediaSize value as Letter and the  
 4 PageOrientation value as Landscape the bottom of the fixed page content is clipped,  
 5 as shown below.

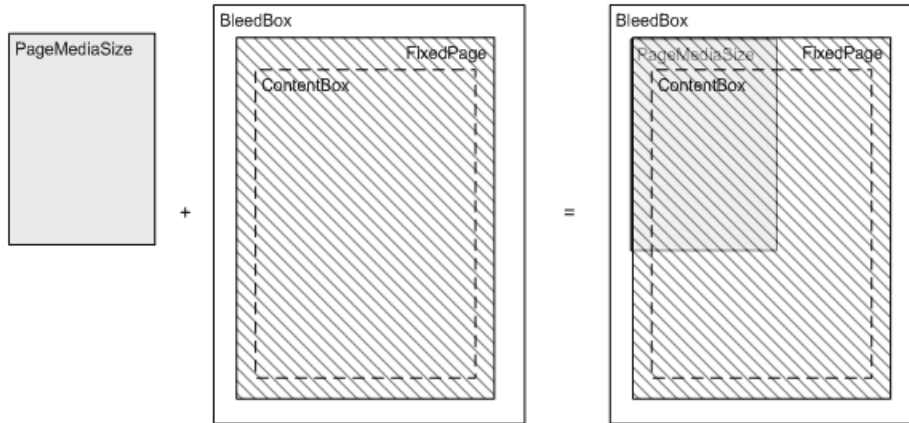
6 *Figure 10–3. Matching PrintTicket and fixed page size with differing orientation*



7

8 Finally, if the PrintTicket specifies the PageMediaSize value as 4x6 and the  
 9 PageOrientation value as Portrait, the bottom and right portions of the fixed page  
 10 content are clipped, as shown below.

1 Figure 10–4. Matching PrintTicket and fixed page orientation with differing size



2  
3 *end example]*

4 [Note: Fixed pages intended for landscape-orientation printing must be produced  
5 differently than those intended for portrait-orientation printing; their **Width** attribute  
6 holds the larger value, their **Height** attribute holds the smaller value.

7 The PageOrientation setting in the PrintTicket does not rotate the fixed page, but  
8 instead determines how the PageMediaSize dimensions relate to the  
9 PageImageableSize dimensions reported by the device.

10 As a consequence, in order to print a fixed page that has been produced with a  
11 landscape orientation, either the PrintTicket would specify PageMediaSize and  
12 PageOrientation values such that the dimensions of the fixed page match the  
13 PageImageableSize reported by the device, or it would specify an appropriate  
14 PageScaling option. *end note]*

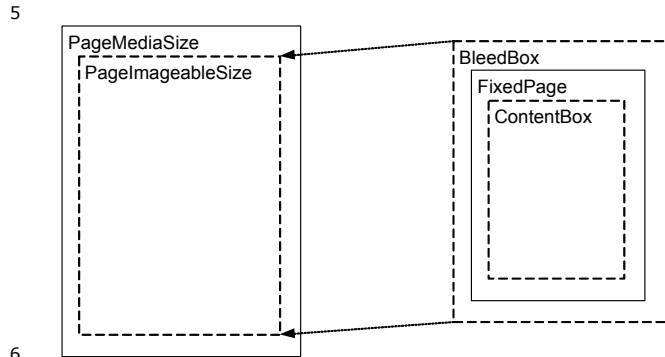
15 For media scaling, the following page scaling options determine the rendering of the  
16 fixed page:

- 17 • FitApplicationBleedSizeToPageImageableSize
- 18 • FitApplicationContentSizeToPageImageableSize
- 19 • FitApplicationMediaSizeToPageImageableSize
- 20 • FitApplicationMediaSizeToPageMediaSize

21 This is not an exhaustive list; for more information, see the Print Schema  
22 specification.

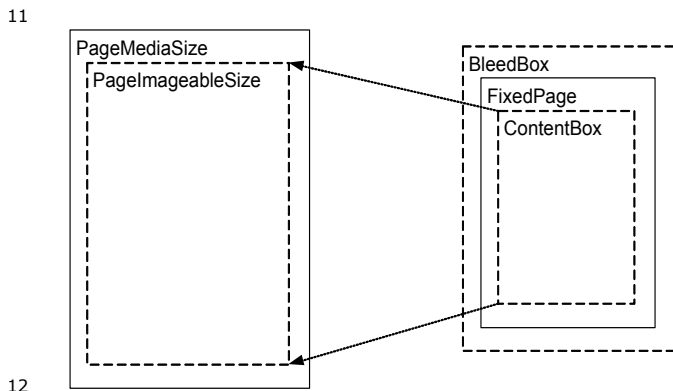
1 **10.3.4.1 FitApplicationBleedSizeToPageImageableSize**

2 Consumers MUST scale the bleed box (producer bleed size) to the  
3 PageImageableSize, preserving the aspect ratio [M3.16]. See the Print Schema  
4 PageScaling definition.



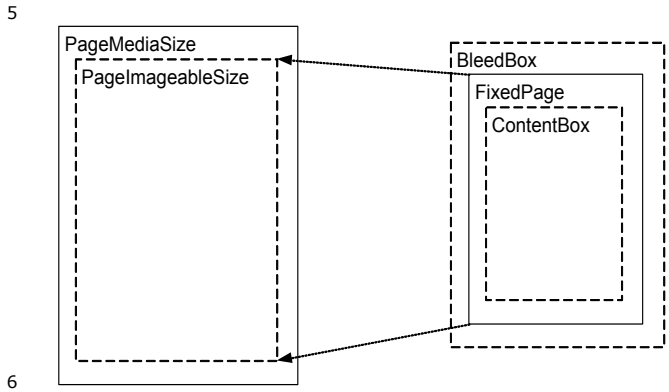
7 **10.3.4.2 FitApplicationContentSizeToPageImageableSize**

8 Consumers MUST scale the content box (producer content size) to the  
9 PageImageableSize, preserving the aspect ratio [M3.17]. See the Print Schema  
10 PageScaling definition.



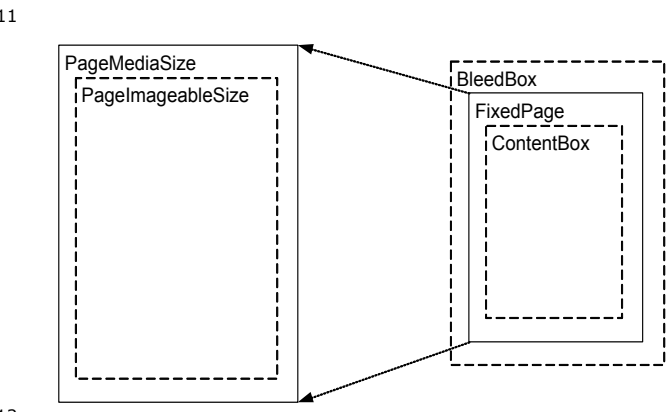
1 **10.3.4.3 FitApplicationMediaSizeToPageImageableSize**

2 Consumers MUST scale the height and width (producer media size) to the  
 3 PageImageableSize, preserving the aspect ratio [M3.18]. See the Print Schema  
 4 PageScaling definition.



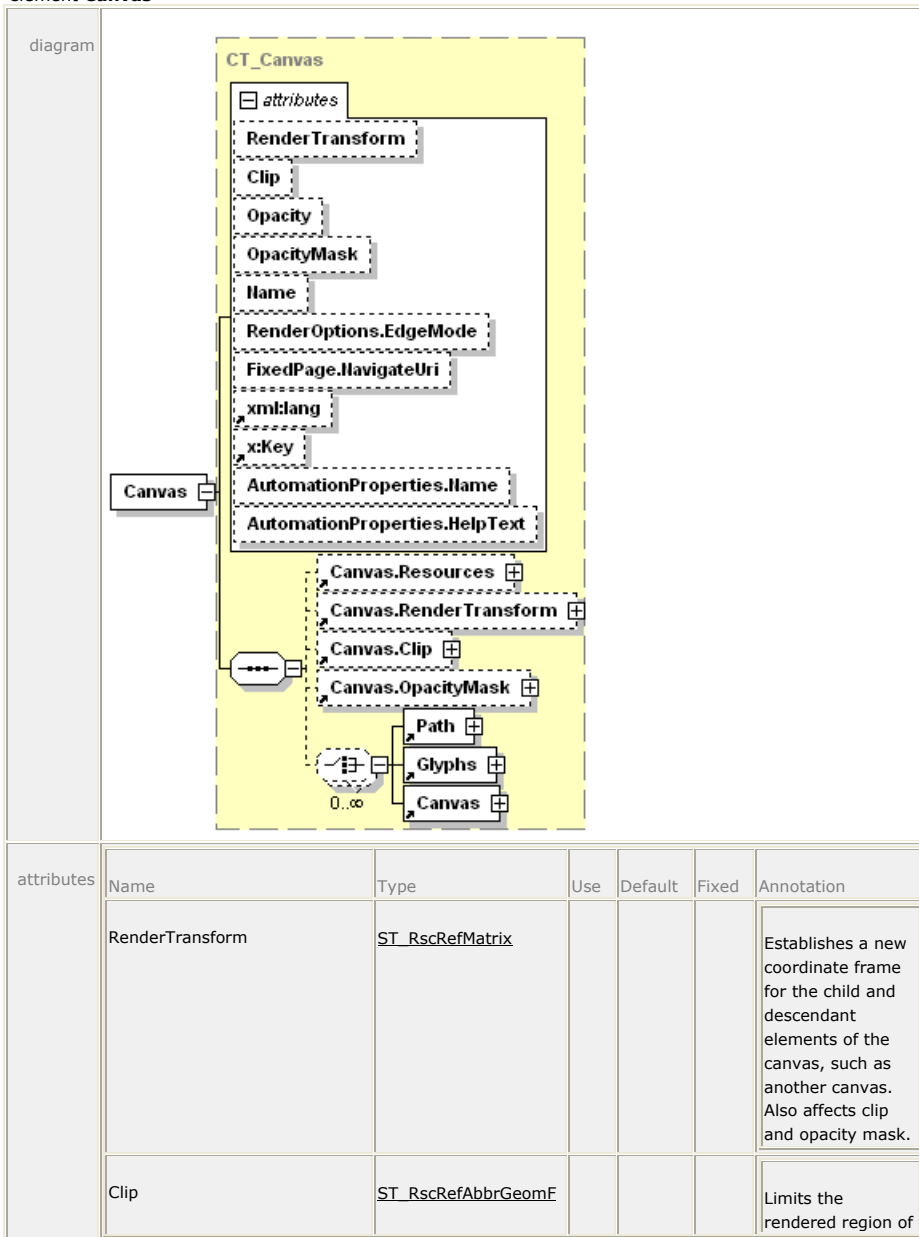
7 **10.3.4.4 FitApplicationMediaSizeToPageMediaSize**

8 Consumers MUST scale the height and width (producer media size) to the  
 9 PageMediaSize, preserving the aspect ratio [M3.19]. See the Print Schema  
 10 PageScaling definition.



1 **10.4 <Canvas> Element**

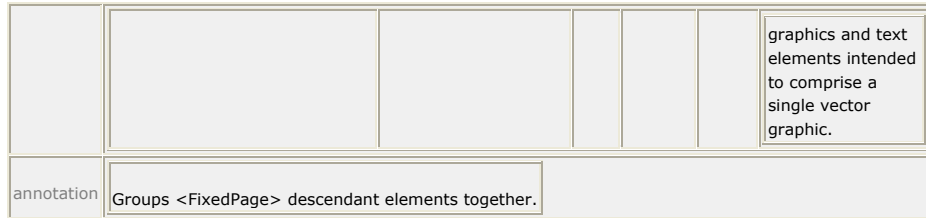
2 element **Canvas**



					the element.
Opacity	<a href="#">ST_ZeroOne</a>		1.0		Defines the uniform transparency of the canvas. Values range from 0 (fully transparent) to 1 (fully opaque), inclusive. Values outside of this range are invalid.
OpacityMask	<a href="#">ST_RscRef</a>				Specifies a mask of alpha values that is applied to the canvas in the same fashion as the Opacity attribute, but allowing different alpha values for different areas of the element.
Name	<a href="#">ST_Name</a>				Contains a string value that identifies the current element as a named, addressable point in the document for the purpose of hyperlinking.
RenderOptions.EdgeMode	<a href="#">ST_EdgeMode</a>				Controls how edges of paths within the canvas are rendered. The only valid value is Aliased. Omitting this attribute causes the edges to be rendered in the consumer's default manner.
FixedPage.NavigateUri	xs:anyURI				Associates a hyperlink URI with the element. May be a relative reference or a URI



					that addresses a resource that is internal to or external to the package.
xml:lang					Specifies the default language used for the current element and for any child or descendant elements. The language is specified according to RFC 3066.
x:Key					Specifies a name for a resource in a resource dictionary. x:Key MUST be present when the current element is defined in a resource dictionary. x:Key MUST NOT be specified outside of a resource dictionary [M3.20].
AutomationProperties.Name	xs:string				A brief description of the <Canvas> contents for accessibility purposes, particularly if filled with a set of vector graphics and text elements intended to comprise a single vector graphic.
AutomationProperties.HelpText	xs:string				A detailed description of the <Canvas> contents for accessibility purposes, particularly if filled with a set of



1 The <Canvas> element groups elements together. [*Example:* <Glyphs> and <Path>  
 2 elements can be grouped in a canvas in order to be identified as a unit (as a  
 3 hyperlink destination) or to apply a composed property value to each child and  
 4 ancestor element. *end example*]

5 Some properties of the <Canvas> element are composable and affect the rendering  
 6 of child elements. This includes the coordinate space of the canvas. For details, see  
 7 14, "Common Properties."

8 The `RenderOptions.EdgeMode` property can be set on the <Canvas> element to  
 9 instruct anti-aliasing consumers to render the contents of the <Canvas> and all child  
 10 and descendant elements without performing anti-aliasing, including child brushes  
 11 and their contents as well as contents included via resource dictionary references.

12 *Example 10-6. Canvas composition*

13 The following markup describes a path that provides the background. On top of this  
 14 is rendered a canvas with the composable `Opacity` and `RenderTransform` properties  
 15 specified.

16 The path inside the canvas has the same path geometry as the background path, but  
 17 since it is composing the <Canvas> element's `RenderTransform` property, it is  
 18 rendered differently. The path is partially transparent due to the composable `Opacity`  
 19 property of the parent <Canvas> element.

20 The <Glyphs> element inside the canvas specifies its own `RenderTransform`  
 21 property. This property is composed with the <Canvas> element's `RenderTransform`  
 22 property, such that the coordinate space of the <Glyphs> element is transformed  
 23 within the context of the coordinate space transformed by the <Canvas> element.

```

24 <Path>
25   <Path.Fill>
26     <SolidColorBrush Color="#808080" />
27   </Path.Fill>
28   <Path.Data>
29     <PathGeometry>
30       <PathFigure StartPoint="0,0" IsClosed="true">
31         <PolyLineSegment Points="200,0 200,100 0,100 0,0" />
32       </PathFigure>
33     </PathGeometry>
34   </Path.Data>
35 </Path>
36
37 <Canvas Opacity="0.5" RenderTransform="0.75,0,0,0.75,25,46">
38   <Path>
39     <Path.Fill>
40       <SolidColorBrush Color="#0000FF" />
41     </Path.Fill>
42     <Path.Data>

```

```
1      <PathGeometry>
2          <PathFigure StartPoint="0,0" IsClosed="true">
3              <PolyLineSegment Points="200,0 200,100 0,100 0,0" />
4          </PathFigure>
5      </PathGeometry>
6  </Path.Data>
7 </Path>
8 <Glyphs
9     FontUri="..\Resources/Fonts/times.ttf"
10    OriginX="1"
11    OriginY="100"
12    UnicodeString="EXAMPLE"
13    FontRenderingEmSize="42"
14    RenderTransform="1.0,0,0,2.0,0,-100">
15    <Glyphs.Fill>
16        <SolidColorBrush Color="#FFFFFF" />
17    </Glyphs.Fill>
18 </Glyphs>
19 </Canvas>
```

20 This markup is rendered as follows:



21  
22 *end example]*

---

### 23 10.5 <Path> Element

24 The <Path> element specifies a geometry that can be filled with a brush. For more  
25 information, see §11.1, "<Path> Element," on page 68.

---

### 26 10.6 <Glyphs> Element

27 The <Glyphs> element is used to represent a run of uniformly-formatted text from a  
28 single font. The <Glyphs> element provides information for accurate rendering and  
29 supports search and selection features in XPS Document consumers. For more  
30 information, see §12.1, "<Glyphs> Element," on page 98.

31

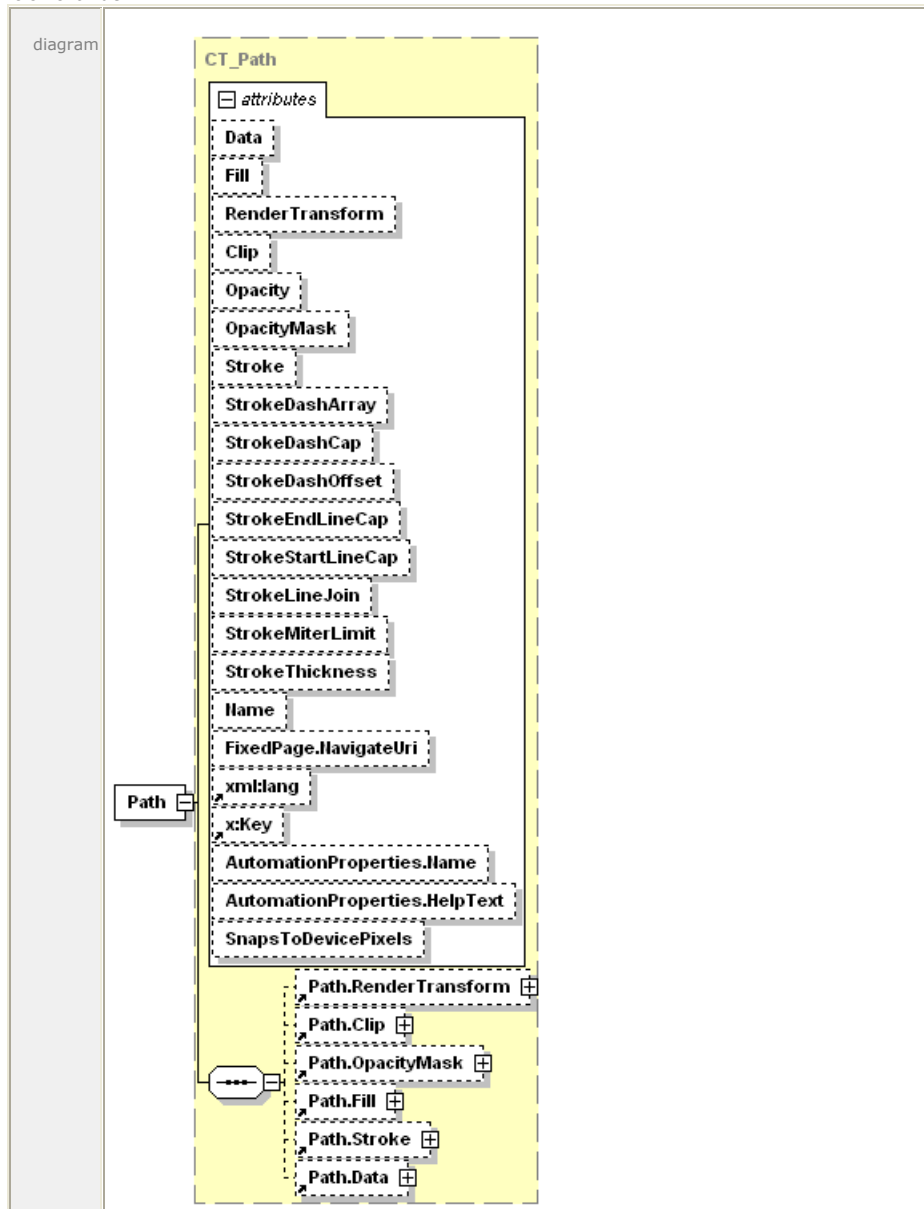


## 1 **11. Graphics**

2 Vector graphics are created using the <Path> element. A full set of properties is  
3 available to describe the visual characteristics of the graphic. These characteristics  
4 include the fill, opacity, clipping, rendering transformation, and various stroke details  
5 including thickness, fill, line join style, line miter limit, line cap style, dash style, and  
6 dash cap style. The description of the geometric area of the path (the geometry) is  
7 described by the Data property. Raster images are included in fixed page markup by  
8 specifying a <Path> element filled with an <ImageBrush>.

1 **11.1 <Path> Element**

2 element **Path**



attributes	Name	Type	Use	Default	Fixed	Annotation
	Data	<a href="#">ST_RscRefAbbrGeomF</a>				Describes the geometry of the path.
	Fill	<a href="#">ST_RscRefColor</a>				Describes the brush used to paint the geometry specified by the Data property of the path.
	RenderTransform	<a href="#">ST_RscRefMatrix</a>				Establishes a new coordinate frame for all attributes of the path and for all child elements of the path, such as the geometry defined by the <Path.Data> property element.
	Clip	<a href="#">ST_RscRefAbbrGeomF</a>				Limits the rendered region of the element.
	Opacity	<a href="#">ST_ZeroOne</a>		1.0		Defines the uniform transparency of the path element. Values range from 0 (fully transparent) to 1 (fully opaque), inclusive. Values outside of this range are invalid.

OpacityMask	<a href="#">ST_RscRef</a>				Specifies a mask of alpha values that is applied to the path in the same fashion as the Opacity attribute, but allowing different alpha values for different areas of the element.
Stroke	<a href="#">ST_RscRefColor</a>				Specifies the brush used to draw the stroke.
StrokeDashArray	<a href="#">ST_EvenArrayPos</a>				Specifies the length of dashes and gaps of the outline stroke. These values are specified as multiples of the stroke thickness as a space-separated list with an even number of non-negative values. When a stroke is drawn, the dashes and gaps specified by these values are repeated to cover the length of the stroke. If this attribute is omitted, the stroke is drawn solid, without any gaps.
StrokeDashCap	<a href="#">ST_DashCap</a>		Flat		Specifies how the ends of each dash are drawn. Valid values are Flat, Round, Square, and Triangle.
StrokeDashOffset	<a href="#">ST_Double</a>		0.0		Adjusts the start point for repeating the dash array



					pattern. If this value is omitted, the dash array aligns with the origin of the stroke. Values are specified as multiples of the stroke thickness.
StrokeEndLineCap	<a href="#">ST_LineCap</a>		Flat		Defines the shape of the end of the last dash in a stroke. Valid values are Flat, Square, Round, and Triangle.
StrokeStartLineCap	<a href="#">ST_LineCap</a>		Flat		Defines the shape of the beginning of the first dash in a stroke. Valid values are Flat, Square, Round, and Triangle.
StrokeLineJoin	<a href="#">ST_LineJoin</a>		Miter		Specifies how a stroke is drawn at a corner of a path. Valid values are Miter, Bevel, and Round. If Miter is selected, the value of StrokeMiterLimit is used in drawing the stroke.
StrokeMiterLimit	<a href="#">ST_GEOne</a>		10.0		The ratio between the maximum miter length and half of the stroke thickness. This value is significant only if the StrokeLineJoin attribute specifies Miter.
StrokeThickness	<a href="#">ST_GEZero</a>		1.0		Specifies the thickness of a stroke, in units of

					the effective coordinate space (includes the path's render transform). The stroke is drawn on top of the boundary of the geometry specified by the <Path> element's Data property. Half of the StrokeThickness extends outside of the geometry specified by the Data property and the other half extends inside of the geometry.
Name	<u>ST_Name</u>				Contains a string value that identifies the current element as a named, addressable point in the document for the purpose of hyperlinking.
FixedPage.NavigateUri	xs:anyURI				Associates a hyperlink URI with the element. Can be a relative reference or a URI that addresses a resource that is internal to or external to the package.
xml:lang					Specifies the default language used for the current element and for any child or descendant elements. The language is specified according to RFC 3066.

	x:Key			Specifies a name for a resource in a resource dictionary. x:Key MUST be present when the current element is defined in a resource dictionary. x:Key MUST NOT be specified outside of a resource dictionary [M4.1].
	AutomationProperties.Name	xs:string		A brief description of the <Path> for accessibility purposes, particularly if filled with an <ImageBrush>.
	AutomationProperties.HelpText	xs:string		A detailed description of the <Path> for accessibility purposes, particularly if filled with an <ImageBrush>.
	SnapsToDevicePixels	<u>ST_Boolean</u>		On Anti-aliasing consumers controls if control points snap to the nearest device pixels. Valid values are 'false' and 'true'. Consumers MAY ignore this attribute [O4.1].
annotation	Defines a single graphical effect to be rendered to the page. It paints a geometry with a brush and draws a stroke around it.			

- 1 The <Path> element is the sole means of adding vector graphics and images to a
- 2 fixed page. It defines a single vector graphic to be rendered on a page. Some
- 3 properties of the <Path> element are composable, meaning that the markings
- 4 rendered to the page are determined by a combination of the property and all of the
- 5 like-named properties of its parent and ancestor elements.

1 The Data property contains a geometric description of the area on which to apply a  
 2 given effect. This description can take one of two forms: verbose or abbreviated. In  
 3 the verbose form, the geometry is described in the <Path.Data> property element  
 4 using the elements described in §11.2, “Geometries and Figures,” on page 77. In  
 5 abbreviated form, it is described using abbreviated syntax in the **Data** attribute. For  
 6 more information, see §11.2.3, “Abbreviated Geometry Syntax,” on page 90.

7 The <Path.Fill> property element describes the appearance of the area specified by  
 8 the Data property. It contains a brush (see 13, “Brushes”) that is used to fill the  
 9 described areas. These can include a solid color, an image, a gradient, or a vector  
 10 drawing pattern.

11 The <Path.Stroke> property element describes the appearance of the borders of the  
 12 shape specified by the Data property. It also contains a <Brush> element, which is  
 13 used to fill the borders according to the stroke properties (such as StrokeThickness).  
 14 See 18, “Rendering Rules,” for detailed rendering rules of strokes, line caps, and  
 15 dash caps.

16 If neither Stroke nor Fill properties are specified, the <Path> element has no visible  
 17 effect.

18 The transparency of the rendered <Path> element is controlled by the **Opacity**  
 19 attribute. More complex transparency descriptions may be defined using the  
 20 **OpacityMask** attribute to control the transparency of the brush described by the Fill  
 21 property.

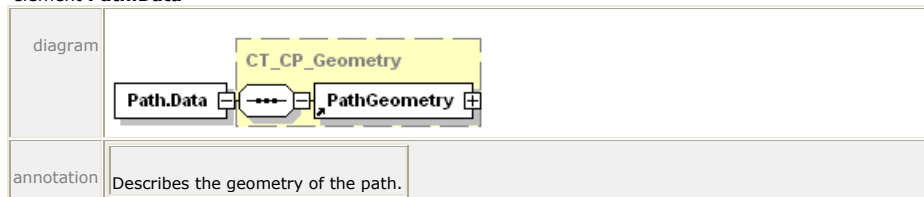
22 Consumers or viewers that perform anti-aliasing MAY “snap” those control points of  
 23 the path that are situated on the path bounding box to whole device pixels if the  
 24 ignorable **SnapsToDevicePixels** attribute is specified as true [O4.1].

25 Finally, the path may be cropped by specifying a clipping region in the Clip property,  
 26 which describes the geometric area to be preserved. The remainder is not rendered.  
 27 See §11.2.1, “Geometries,” on page 78 for how geometries are defined.

28 For details on the Clip, Opacity, OpacityMask, and RenderTransform properties, see  
 29 14, “Common Properties.”

### 30 11.1.1 <Path.Data> Element

31 element **Path.Data**



32 The <Path.Data> property element describes the geometric area of a path. It  
 33 contains a single geometry.

34 *Example 11-1. <Path.Data> usage*

```

35 <Path Stroke="#000000" StrokeThickness="1">
36   <Path.Data>
37     <PathGeometry>

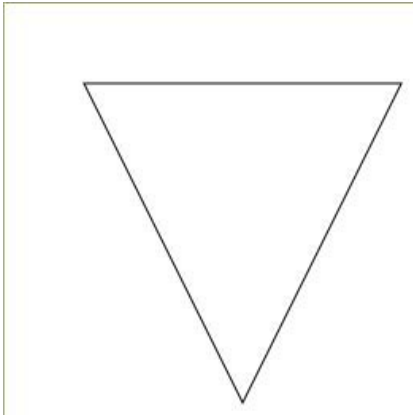
```

```

1 <PathFigure StartPoint="50,50" IsClosed="true">
2 <PolyLineSegment Points="250,50 150,250" />
3 </PathFigure>
4 </PathGeometry>
5 </Path.Data>
6 </Path>

```

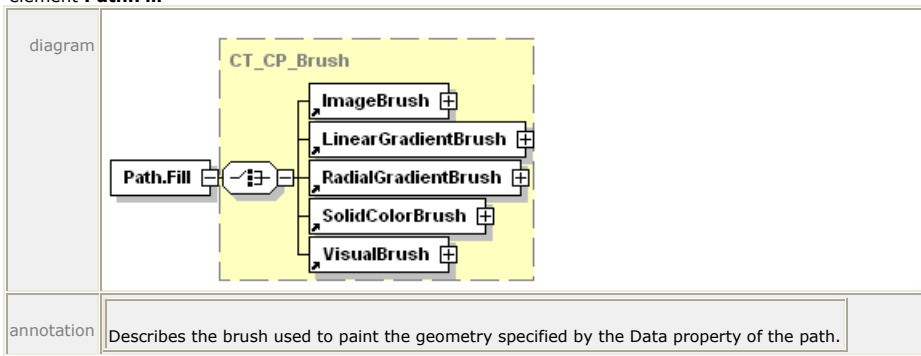
7 This markup produces the following results:



8  
9 *end example]*

10 **11.1.2 <Path.Fill> Element**

11 element **Path.Fill**



12 The <Path.Fill> property element specifies the brush that is used to fill the region  
13 described by the Data property. This can be a solid color, an image, a gradient, or a  
14 vector drawing pattern.

1 *Example 11-2. <Path.Fill> usage*

2 In the following markup, the geometry is filled with a solid color:

```

3 <Path>
4 <Path.Fill>
5 <SolidColorBrush Color="#0000FF" />
6 </Path.Fill>
7 <Path.Data>
8 <PathGeometry>
9 <PathFigure StartPoint="10,10" IsClosed="true">
10 <PolyLineSegment Points="50,200 100,40 150,200
11 200,10 100,105" />
12 </PathFigure>
13 </PathGeometry>
14 </Path.Data>
15 </Path>
    
```

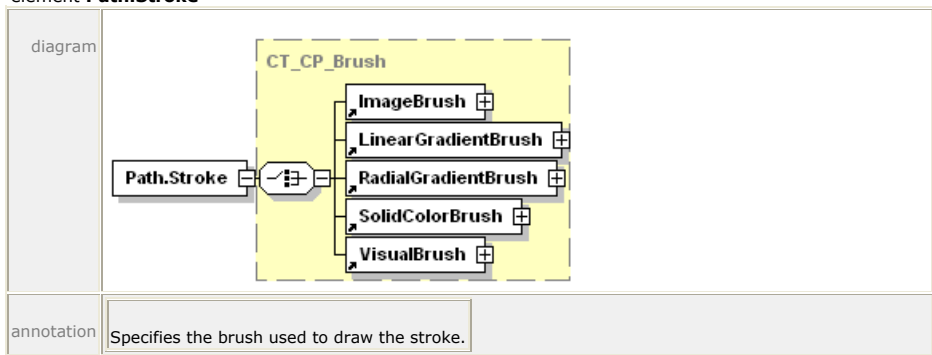
16 This markup produces the following result:



17  
18 *end example]*

19 **11.1.3 <Path.Stroke> Element**

20 element **Path.Stroke**



21 The <Path.Stroke> property element describes the border of the path's geometry.  
22 <Path.Stroke> contains a brush. Only those segments of the path figure in the

1 <Path.Data> element that set the **IsStroked** attribute to true (the default value if  
 2 omitted) are stroked. If **IsClosed** is set to true, an extra segment will be stroked,  
 3 connecting the last point in the path figure with the first point in the path figure.

4 The <Path.Stroke> property element is then used to describe the appearance of the  
 5 borders of the shape defined by the Data property. It also contains a brush, which is  
 6 used to fill the borders according to the stroke properties (such as StrokeThickness).

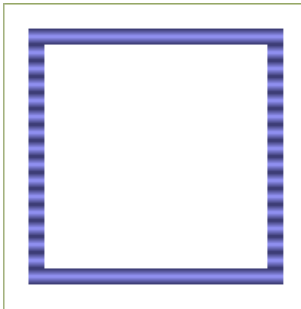
7 For more information, see §18.6, "Stroke Rendering," on page 288.

8 *Example 11-3. <Path.Stroke> usage*

9 The following <Path.Stroke> element uses a gradient brush to fill the border of a  
 10 box:

```
11 <Path StrokeThickness="10" Data="M 20,20 L 170,20 L 170,170 L 20,170 Z">
12 <Path.Stroke>
13 <LinearGradientBrush MappingMode="Absolute"
14 StartPoint="0,0" EndPoint="0,5" SpreadMethod="Reflect">
15 <LinearGradientBrush.GradientStops>
16 <GradientStop Color="#9999FF" Offset="0.0" />
17 <GradientStop Color="#333366" Offset="1.0" />
18 </LinearGradientBrush.GradientStops>
19 </LinearGradientBrush>
20 </Path.Stroke>
21 </Path>
```

22 This markup produces the following results:



23

24 *end example]*

---

## 25 11.2 Geometries and Figures

26 Geometries are used to build visual representations of geometric shapes.

27 The smallest atomic unit in a geometry is a segment. Segments may be lines or  
 28 curves. One or more segments are combined into a path figure definition. A path  
 29 figure is a single shape comprised of continuous segments. One or more path figures  
 30 collectively define an entire path geometry. A path geometry MAY define the fill  
 31 algorithm to be used on the component path figures [M2.72].

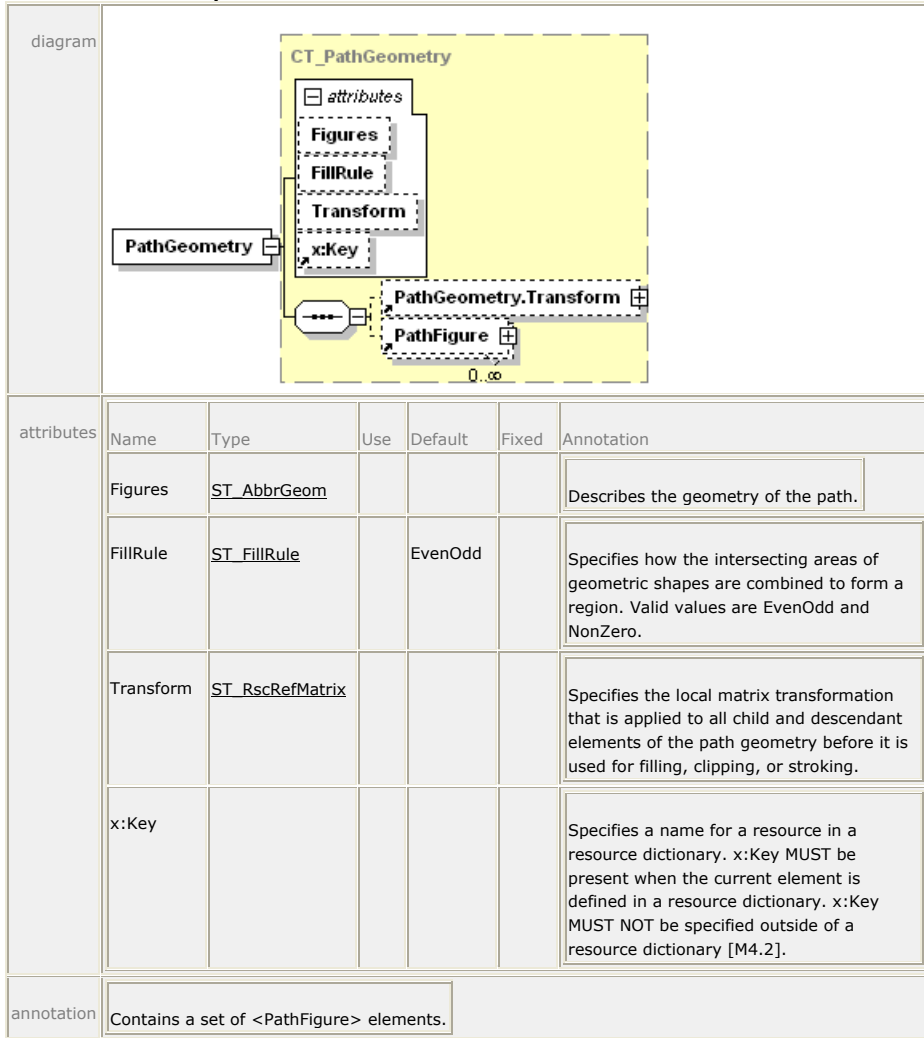
32 A single path geometry may be used in the Data property of the <Path> element to  
 33 describe its overall geometry. A path geometry may also be used in the Clip property  
 34 of the <Canvas>, <Path>, or <Glyphs> elements to describe a clipping region.

1 **11.2.1 Geometries**

2 A <PathGeometry> element constitutes a complete geometry definition.

3 **11.2.1.1 <PathGeometry> Element**

4 element **PathGeometry**



5 A <PathGeometry> element contains a set of path figures specified either with the  
 6 **Figures** attribute or with a child <PathFigure> element. Producers MUST NOT  
 7 specify the path figures of a geometry with both the **Figures** attribute and a child  
 8 <PathFigure> element [M4.3].



1 The union of the path figures defines the interior of the path geometry according to  
 2 the **FillRule** attribute as described in §11.2.1.2, "FillRule Attribute," on page 79.

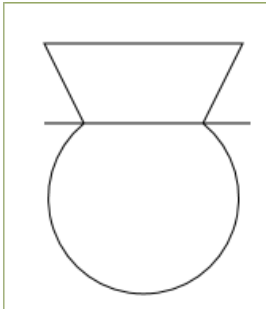
3 *Example 11-4. <PathGeometry> usage*

```

4 <Path Stroke="#000000">
5 <Path.Data>
6 <PathGeometry>
7 <PathFigure StartPoint="25,75">
8 <PolyLineSegment Points="150,75 50,75" />
9 </PathFigure>
10 <PathFigure StartPoint="50,75" IsClosed="true">
11 <ArcSegment
12 Size="60,60"
13 RotationAngle="0"
14 IsLargeArc="true"
15 SweepDirection="Counterclockwise"
16 Point="125,75" />
17 </PathFigure>
18 <PathFigure StartPoint="50,75" IsClosed="true">
19 <PolyLineSegment Points="25,25 150,25 125,75" />
20 </PathFigure>
21 </PathGeometry>
22 </Path.Data>
23 </Path>

```

24 This markup produces the following results:



25

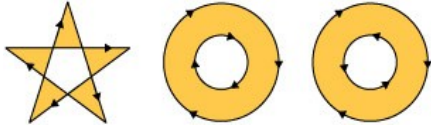
26 *end example]*

### 27 11.2.1.2 FillRule Attribute

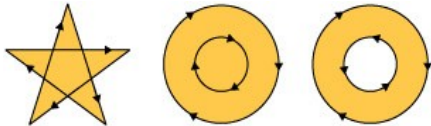
28 The **FillRule** attribute specifies a fill algorithm. The fillable area of a geometry is  
 29 defined by taking all of the contained path figures and applying the fill algorithm to  
 30 determine the enclosed area. Fill algorithms determine how the intersecting areas of  
 31 geometric shapes are combined to form a region.

#### 32 11.2.1.2.1 EvenOdd Fill Algorithm

33 This rule determines the "insideness" of a point on the canvas by drawing a ray from  
 34 the point to infinity in any direction and counting the number of segments from the  
 35 given shape that the ray crosses. If this number is odd, the point is inside; if it is  
 36 even, the point is outside. This is the default rule used throughout XPS Document  
 37 markup.

1 *Figure 11-1. Fill using EvenOdd algorithm*2  
3 **11.2.1.2.2 NonZero Fill Algorithm**

4 This rule determines the “insideness” of a point on the canvas by drawing a ray from  
 5 the point to infinity in any direction and then examining the places where a segment  
 6 of the shape crosses the ray. Starting with a count of zero, add one each time a  
 7 segment crosses the ray from left to right and subtract one each time a path  
 8 segment crosses the ray from right to left. After counting the crossings, if the result  
 9 is zero then the point is outside the path; otherwise, it is inside.

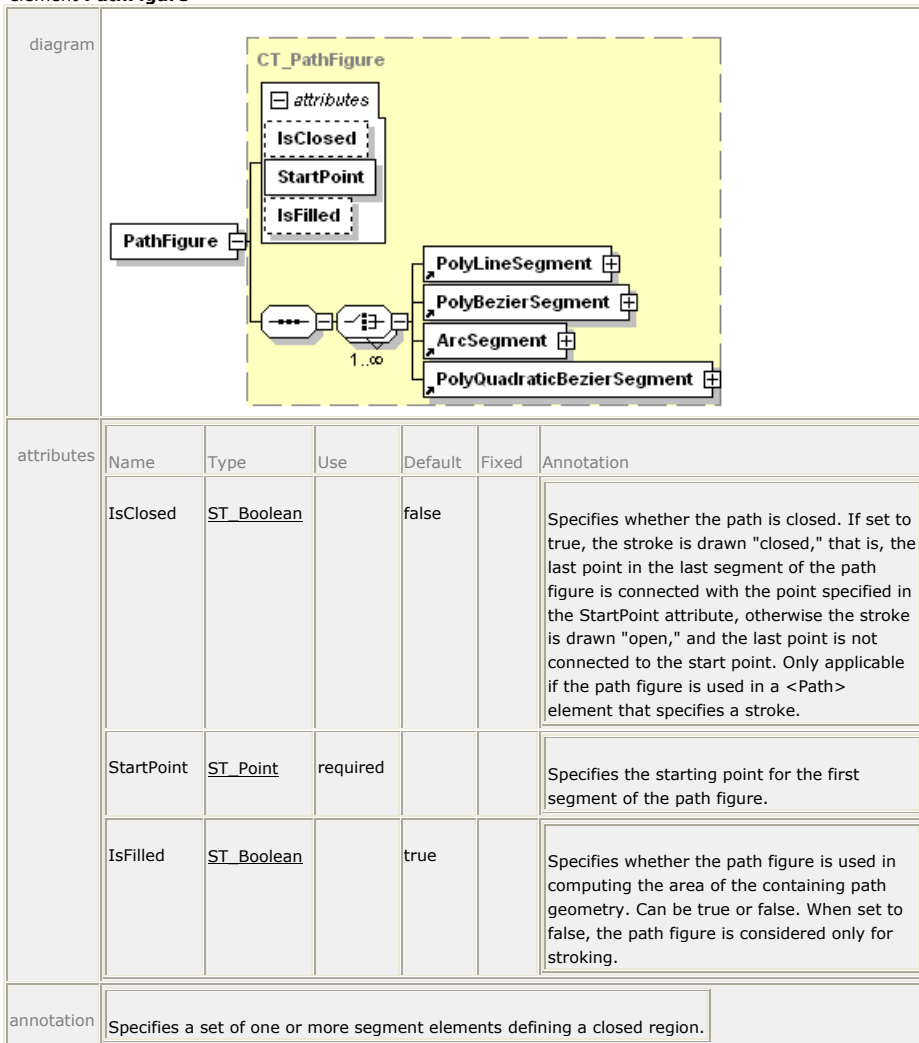
10 *Figure 11-2. Fill using NonZero algorithm*11  
12 **11.2.1.3 Figures Attribute**

13 The <PathGeometry> element’s **Figures** attribute can be used to describe the path  
 14 figures the geometry contains using abbreviated syntax (see §11.2.3, “Abbreviated  
 15 Geometry Syntax,” on page 90) with the exception that the FillRule command MUST  
 16 NOT be used [M2.72].

1 **11.2.2 Figures**

2 **11.2.2.1 <PathFigure> Element**

3 element **PathFigure**



4 A <PathFigure> element is composed of a set of one or more line or curve segments.  
 5 The segment elements define the shape of the path figure. The initial point of the  
 6 first segment element is specified as the **StartPoint** attribute of the path figure. The  
 7 last point of each segment element is the first point of the following segment  
 8 element.

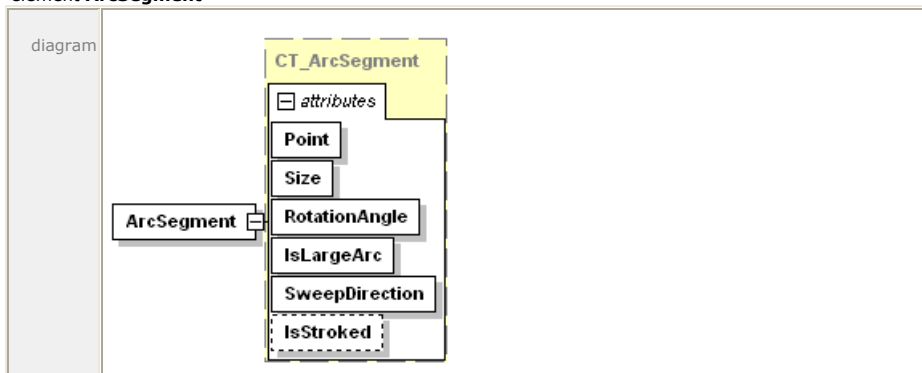
1 Segment elements are:

- 2 • <ArcSegment>
- 3 • <PolyBezierSegment>
- 4 • <PolyLineSegment>
- 5 • <PolyQuadraticBezierSegment>

6 Line segments and curve segments SHOULD NOT be specified as zero-length [S4.1].  
 7 If they are specified as zero-length, nothing is drawn, even if line caps would  
 8 normally be visible. For further details and exceptions, see 18, "Rendering Rules."

9 **11.2.2.2 <ArcSegment> Element**

10 element **ArcSegment**



attributes						
Name	Type	Use	Default	Fixed	Annotation	
Point	<a href="#">ST_Point</a>	required			Specifies the endpoint of the elliptical arc.	
Size	<a href="#">ST_PointGE0</a>	required			Specifies the x and y radius of the elliptical arc as an x,y pair.	
RotationAngle	<a href="#">ST_Double</a>	required			Indicates how the ellipse is rotated relative to the current coordinate system.	
IsLargeArc	<a href="#">ST_Boolean</a>	required			Determines whether the arc is drawn with a sweep of 180 or greater. Can be true or false.	
SweepDirection	<a href="#">ST_SweepDirection</a>	required			Specifies the direction in which the arc is drawn. Valid values are Clockwise and Counterclockwise.	

	IsStroked	ST_Boolean	true	Specifies whether the stroke for this segment of the path is drawn. Can be true or false.
annotation	Represents an elliptical arc between two points.			

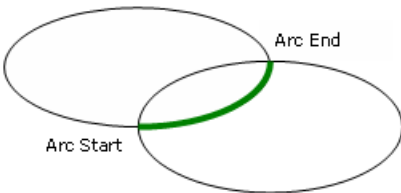
- 1 The <ArcSegment> element describes an elliptical arc. It is geometrically defined by  
 2 the intersection of two ellipses that have the same x radius and y radius. The ellipses  
 3 intersect at the starting and ending points of the arc.

4 Table 11-9. Arc segment definition

Term	Description
Starting Point	Implicitly defined by the previous point in the path figure definition.
Ending Point	Specified by the Point attribute.
Arc Size	Defined by the Size attribute. This value consists of the comma-delimited x and y radii of the ellipses that will be used to define the arc. [Example: "100,50" end example]
Rotation Angle	Specified by the RotationAngle attribute, this determines how the ellipses defining the arc are rotated with respect to the x axis, in degrees. Positive values are clockwise and negative values are counter-clockwise.
Large Arc Flag	Specified by the IsLargeArc attribute, this flag indicates which of the arc pairs created by the intersecting ellipses to use. When the flag is true, it uses the larger arc (arc length $\geq 180^\circ$ ), and when it is false it uses the smaller arcs (arc length $< 180^\circ$ ).
Sweep Direction	Specified by the SweepDirection attribute, this flag determines which of the two possible arcs (selected by the Large Arc Flag) is used. Beginning at the starting point, one arc proceeds in the positive (clockwise) direction, while the other proceeds in the negative (counter-clockwise) direction.

1 *Figure 11-3. Arc choice A*

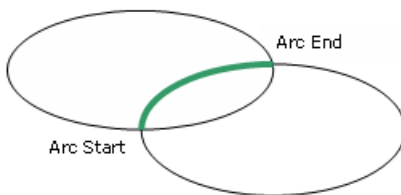
2 `IsLargeArc = false; SweepDirection = Counterclockwise`



3

4 *Figure 11-4. Arc choice B*

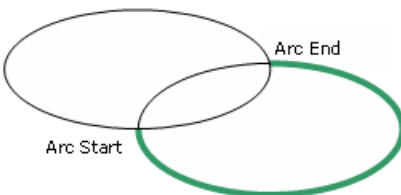
5 `IsLargeArc = false; SweepDirection = Clockwise`



6

7 *Figure 11-5. Arc choice C*

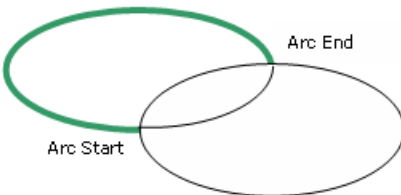
8 `IsLargeArc = true; SweepDirection = Counterclockwise`



9

10 *Figure 11-6. Arc choice D*

11 `IsLargeArc = true; SweepDirection = Clockwise`



12

13 *Example 11-5. <ArcSegment> usage*

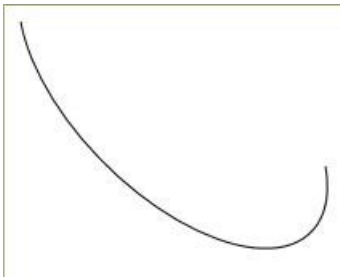
```
14 <Path Stroke="#000000" StrokeThickness="1">
15 <Path.Data>
```

```

1  <PathGeometry>
2  <PathFigure StartPoint="10,10">
3  <ArcSegment
4  Size="100,50"
5  RotationAngle="45"
6  IsLargeArc="true"
7  SweepDirection="Counterclockwise"
8  Point="200,100" />
9  </PathFigure>
10 </PathGeometry>
11 </Path.Data>
12 </Path>

```

13 This markup generates the following arc:



14

15 *end example]*

#### 16 11.2.2.2.1 Out-of-Range Attributes

17 The following guidelines are followed when encountering incompatible attribute  
 18 values on an <ArcSegment> element:

- 19 • If the arc is impossible to render given the combination of radii specified in the  
 20 **Size** attribute and the angle of rotation specified in the **RotationAngle**  
 21 attribute, the ellipses are scaled equally until there is exactly one solution that  
 22 satisfies the arc requirements to pass through the specified **Point** attribute.
- 23 • If the **Point** attribute is the same as the previous point in the path figure, the  
 24 segment is omitted.
- 25 • If either the x or y radius in the **Size** attribute is 0, the segment is rendered as  
 26 a poly line segment with a single line segment to the x,y coordinates specified  
 27 by the **Point** attribute.
- 28 • The x or y radius in the **Size** attribute MUST NOT be negative [M2.72].
- 29 • If the **RotationAngle** value is greater than 360, it is replaced by the value of  
 30 the **RotationAngle** modulo 360. If it is less than 0, it is replaced with a value  
 31 normalized to the range 0–360.

1 **11.2.2.3 <PolyBezierSegment> Element**

2 element **PolyBezierSegment**

diagram						
attributes	Name	Type	Use	Default	Fixed	Annotation
	Points	<a href="#">ST_Points</a>	required			Specifies control points for multiple Bézier segments. Coordinate values within each pair are comma-separated and additional whitespace may appear. Coordinate pairs are separated from other coordinate pairs by whitespace.
	IsStroked	<a href="#">ST_Boolean</a>		true		Specifies whether the stroke for this segment of the path is drawn. Can be true or false.
annotation	A series of Bézier segments.					

3 The <PolyBezierSegment> element describes a set of cubic Bézier curves. Bézier  
 4 curves are drawn from the previous point in the path figure or the previous Bézier  
 5 curve in the segment and terminate at the third point ( $x_{3n}, y_{3n}$ ) in the **Points**  
 6 attribute (where  $n$  is the curve being drawn). The tangents and curvature of each  
 7 Bézier curve are controlled by the first two control points ( $x_{3n-2}, y_{3n-2}$  and  $x_{3n-1}, y_{3n-1}$ ) in  
 8 the **Points** attribute. The **Points** attribute contains a multiple of three whitespace-  
 9 delimited pairs of comma-delimited  $x,y$  values.

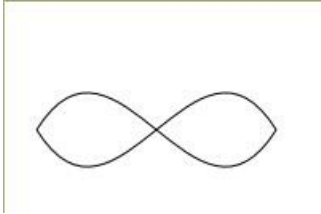
10 *Example 11-6. <PolyBezierSegment> usage*

```

11 <Path Stroke="#000000" StrokeThickness="1">
12 <Path.Data>
13 <PathGeometry>
14 <PathFigure StartPoint="20,80">
15 <PolyBezierSegment Points="70,0 120,160 170,80 120,0 70,160
16 20,80" />
17 </PathFigure>
18 </PathGeometry>
19 </Path.Data>
20 </Path>
    
```



1 This markup generates the following results:



2

3 *end example]*

#### 4 11.2.2.4 <PolyLineSegment> Element

5 element **PolyLineSegment**

diagram						
attributes	Name	Type	Use	Default	Fixed	Annotation
	Points	<a href="#">ST_Points</a>	required			Specifies a set of coordinates for the multiple segments that define the poly line segment. Coordinate values within each pair are comma-separated and additional whitespace may appear. Coordinate pairs are separated from other coordinate pairs by whitespace.
	IsStroked	<a href="#">ST_Boolean</a>		true		Specifies whether the stroke for this segment of the path is drawn. Can be true or false.
annotation	Specifies a set of points between which lines are drawn.					

6 The <PolyLineSegment> element describes a polygonal drawing containing an  
 7 arbitrary number of individual vertices. The **Points** attribute defines the vertices and  
 8 contains whitespace-delimited pairs of comma-delimited x,y values.

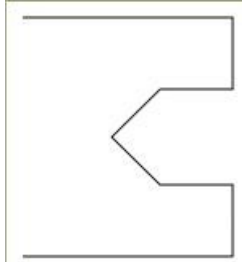
9 *Example 11-7. <PolyLineSegment> usage*

```

10 <Path Stroke="#000000" StrokeThickness="1">
11   <Path.Data>
12     <PathGeometry>
13       <PathFigure StartPoint="10,10">
14         <PolyLineSegment Points="140,10 140,55 95,55 65,85 95,115
15           140,115 140,160 10,160" />
16       </PathFigure>
17     </PathGeometry>
18   </Path.Data>
19 </Path>

```

1 This markup produces the following figure:



2  
3 *end example]*

4 **11.2.2.5 <PolyQuadraticBezierSegment> Element**

5 element **PolyQuadraticBezierSegment**

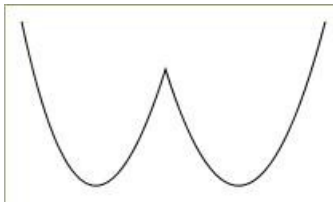
diagram						
attributes	Name	Type	Use	Default	Fixed	Annotation
	Points	<a href="#">ST_Points</a>	required			Specifies control points for multiple quadratic Bézier segments. Coordinate values within each pair are comma-separated and additional whitespace may appear. Coordinate pairs are separated from other coordinate pairs by whitespace.
	IsStroked	<a href="#">ST_Boolean</a>		true		Specifies whether the stroke for this segment of the path is drawn. Can be true or false.
annotation	A series of quadratic Bézier segments.					

6 The <PolyQuadraticBezierSegment> element describes a set of quadratic Bézier  
 7 curves from the previous point in the path figure through a set of vertices, using  
 8 specified control points. The **Points** attribute defines an off-curve control point ( $x_{2n-1}, y_{2n-1}$ )  
 9 followed by the end point ( $x_{2n}, y_{2n}$ ) for each quadratic Bézier curve (where  $n$   
 10 represents the quadratic Bézier curve). The **Points** attribute contains a multiple of  
 11 two whitespace-delimited pairs of comma-delimited  $x,y$  values.

1 *Example 11-8. <PolyQuadraticBezierSegment> usage*

```
2 <Path Stroke="#000000" StrokeThickness="1">
3 <Path.Data>
4 <PathGeometry>
5 <PathFigure StartPoint="10,10">
6 <PolyQuadraticBezierSegment Points="50,200 100,40 150,200
7 200,10" />
8 </PathFigure>
9 </PathGeometry>
10 </Path.Data>
11 </Path>
```

12 This markup produces the following curve:



13

14 *end example]*

#### 15 11.2.2.6 Closed <PathFigure>

16 If the **IsClosed** attribute of the <PathFigure> element is set to true, a straight line  
17 is drawn from the last point in the last segment of the <PathFigure> element to the  
18 **StartPoint** attribute of the <PathFigure> element. If the **IsClosed** attribute is  
19 omitted, its default setting is "false".

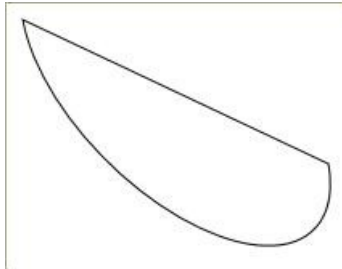
20 <PathFigure> elements used in filled <Path> elements or as **Clip** attributes are  
21 implicitly closed.

22 *Example 11-9. Closed <PathFigure> usage*

23 The following markup shows the arc segment as shown in Example 11-5 with the  
24 **IsClosed** attribute of the <PathFigure> element set to true.

```
25 <Path Stroke="#000000" StrokeThickness="1">
26 <Path.Data>
27 <PathGeometry>
28 <PathFigure StartPoint="10,10" IsClosed="true">
29 <ArcSegment
30 Size="100,50"
31 RotationAngle="45"
32 IsLargeArc="true"
33 SweepDirection="Counterclockwise"
34 Point="200,100" />
35 </PathFigure>
36 </PathGeometry>
37 </Path.Data>
38 </Path>
```

1 This markup generates the following figure:



2  
3 *end example]*

#### 4 **11.2.3 Abbreviated Geometry Syntax**

5 Abbreviated geometry syntax MAY be used to specify a geometry of one or more  
6 figures comprised of multiple segments [M2.72]. A geometry is specified with an  
7 optional FillRule command (not allowed in the **Figures** attribute of a  
8 <PathGeometry> element) followed by one or more figure definitions. Figure  
9 definitions are specified with a Move command, a set of drawing commands to create  
10 segments, and an optional Close command to create a closing segment. Drawing  
11 commands include:

- 12 • Line
- 13 • Horizontal Line
- 14 • Vertical Line
- 15 • Cubic Bézier Curve
- 16 • Quadratic Bézier Curve
- 17 • Smooth Cubic Bézier Curve
- 18 • Elliptical Arc

19 A command is represented by a single letter and is followed by zero or more  
20 whitespace characters, which are followed by command parameters. Parameters are  
21 whitespace-delimited. Points are specified as a comma-delimited pair with zero or  
22 more whitespace characters.

23 Uppercase letters denote absolute values and lowercase letters denote relative  
24 values. When relative coordinate values are specified, each coordinate pair expresses  
25 an offset relative to the current endpoint (the previous command's terminating  
26 coordinate pair). If a relative value is used for the first Move command, the current  
27 endpoint is, by definition, 0,0.

28 If a relative value is used following a Close command, the current endpoint is the  
29 first point of the previous figure.

30 If entering more than one drawing command of the same type sequentially, the  
31 duplicate command entry MAY be omitted [M2.72]. [*Example*: "L 100,200 300,400"  
32 is equivalent to "L 100,200 L 300,400". *end example*] The current endpoint is  
33 determined as though each command appeared individually.

- 1 Values specifying coordinates can be real numbers.
- 2 For more information, see F, "Abbreviated Geometry Syntax Algorithm."
- 3 *Table 11-10. Commands*

Name	Syntax	Description	Non-Abbreviated Equivalent
FillRule	F fFillRule	Establishes the fill rule that should be used for this geometry. A value of 0 is equivalent to a FillRule value of EvenOdd; a value of 1 is equivalent to a FillRule value of NonZero. The default value if this command is omitted is 0.  This command MUST appear only as the first command in the abbreviated geometry syntax [M2.72]. This command MUST NOT be specified in the value of the Figures attribute of the <PathGeometry> element [M2.72]. [ <i>Example: F 0 end example</i> ]	<PathGeometry> FillRule attribute
Move	M x,y or m x,y	Establishes a new current endpoint. Every geometry MAY specify one or more figures, and MAY be preceded by a FillRule command where allowed [M2.72]. The first figure in a geometry MUST begin with a Move command [M2.72]. Subsequent Move commands indicate the start of a new figure but MAY be omitted, indicating the current endpoint for the subsequent figure is the same as the endpoint of the previous figure [M2.72]. [ <i>Example: M 1.0,1.5 end example</i> ]	<PathFigure> StartPoint attribute
Line	L x,y or l x,y	Draws a straight line from the current point to the specified point. [ <i>Example: L 20,30 end example</i> ]	<PolyLineSegment> element
Horizontal Line	H x or h x	Draws a horizontal line from the current endpoint to the specified x coordinate. [ <i>Example: H 90 end example</i> ]	<PolyLineSegment> element

Name	Syntax	Description	Non-Abbreviated Equivalent
Vertical Line	v y or v y	Draws a vertical line from the current endpoint to the specified y coordinate. [Example: v 90 end example]	<PolyLineSegment> element
Cubic Bézier Curve	C x <sub>1</sub> ,y <sub>1</sub> x <sub>2</sub> ,y <sub>2</sub> x <sub>3</sub> ,y <sub>3</sub> or c x <sub>1</sub> ,y <sub>1</sub> x <sub>2</sub> ,y <sub>2</sub> x <sub>3</sub> ,y <sub>3</sub>	Draws a cubic Bézier curve from the current endpoint to the specified point (x <sub>3</sub> ,y <sub>3</sub> ) using the two specified control points (x <sub>1</sub> ,y <sub>1</sub> and x <sub>2</sub> ,y <sub>2</sub> ). The first control point determines the initial direction (tangent) of the curve, and the second determines the terminating direction (tangent) of the curve. [Example: c 100,200 200,400 300,200 end example]	<PolyBezierSegment> element
Quadratic Bézier Curve	Q x <sub>1</sub> ,y <sub>1</sub> x <sub>2</sub> ,y <sub>2</sub> or q x <sub>1</sub> ,y <sub>1</sub> x <sub>2</sub> ,y <sub>2</sub>	Draws a quadratic Bézier curve from the current endpoint to the specified point (x <sub>2</sub> ,y <sub>2</sub> ) using the specified control point (x <sub>1</sub> ,y <sub>1</sub> ). [Example: q 100,200 300,200 end example]	<PolyQuadraticBezierSegment> element
Smooth Cubic Bézier Curve	S x <sub>1</sub> ,y <sub>1</sub> x <sub>2</sub> ,y <sub>2</sub> or s x <sub>1</sub> ,y <sub>1</sub> x <sub>2</sub> ,y <sub>2</sub>	Draws a cubic Bézier curve from the current endpoint to the specified point (x <sub>2</sub> ,y <sub>2</sub> ). The first control point is assumed to be the reflection of the second control point of the previous command, relative to the current endpoint. If there is no previous command or if the previous command was not a Cubic Bézier Curve command or Smooth Cubic Bézier Curve command, the first control point is assumed to be coincident with the current endpoint. The second control point is specified by x <sub>1</sub> ,y <sub>1</sub> . [Example: s 100,200 200,300 end example]	<PolyBezierSegment> element
Elliptical Arc	A x <sub>r</sub> ,y <sub>r</sub> r <sub>x</sub> fArc fSweep x,y or a x <sub>r</sub> ,y <sub>r</sub> r <sub>x</sub> fArc fSweep x,y	Draws an elliptical arc from the current endpoint to the specified point (x,y). The size and orientation of the ellipse are defined by x <sub>r</sub> ,y <sub>r</sub> . r <sub>x</sub> ,r <sub>y</sub> defines the x radius, y <sub>r</sub> defines	<ArcSegment> element

Name	Syntax	Description	Non-Abbreviated Equivalent
		<p>the <math>y</math> radius, and <math>r_x</math> defines the <math>x</math>-axis rotation in degrees, which indicates how the ellipse is rotated relative to the current coordinate system. The center of the ellipse is calculated automatically.</p> <p>In most situations, four different arcs satisfy the specified constraints. <math>f_{Arc}</math> and <math>f_{Sweep}</math> indicate which arc to use.</p> <p>Of the four candidate arc sweeps, two represent large arcs with sweeps of <math>180^\circ</math> or greater, and two represent smaller arcs with sweeps less than <math>180^\circ</math>.</p> <p>If <math>f_{Arc}</math> is 1, one of the two larger arc sweeps is chosen. If <math>f_{Arc}</math> is 0, one of the smaller arc sweeps is chosen. No other values of <math>f_{Arc}</math> are valid.</p> <p>If <math>f_{Sweep}</math> is 1, the arc is drawn in a positive-angle (clockwise) direction. If <math>f_{Sweep}</math> is 0, the arc is drawn in a negative-angle (counter-clockwise) direction. No other values of <math>f_{Sweep}</math> are valid. [<i>Example: a 200,70 10 0 1 100,100 end example</i>]</p>	
Close	Z or z	<p>Draws a straight line from the current endpoint to the first point of the current figure and then ends the figure.</p> <p>If the command following a Close command is a Move command, the Move command specifies the initial point of the next figure. Otherwise, the next figure starts at the same initial point as the current figure.</p>	<PathFigure> IsClosed attribute

1 *Example 11-10. A path described using abbreviated syntax*

2 The following markup demonstrates a simple path, which is drawn using the  
3 abbreviated syntax:

```
4 <Path Stroke="#000000" Data="M 100,100 L 300,100 L 200,300 z" />
```

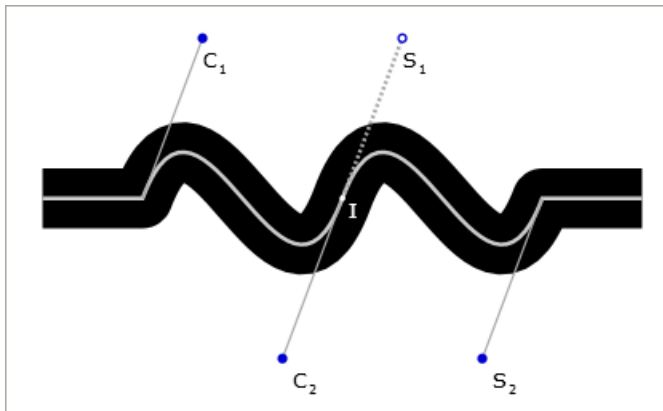
5 *end example]*

### 6 11.2.3.1 Smooth Bézier Curve Abbreviated Syntax

7 Smooth Bézier curves specified with the abbreviated geometry syntax are basic cubic  
8 Bézier curves with an implied first control point. This control point is coincident with  
9 the endpoint of the previous segment unless the previous segment is also a Bézier  
10 curve. In this case, the first control point of the smooth Bézier curve is a reflection of  
11 the second control point of the previous curve segment around the start point of the  
12 smooth Bézier curve segment, as shown below.

13 *Example 11-11. Smooth Bézier curve*

14 In the following example,  $C_1$  and  $C_2$  represent the first and second control points of  
15 the first cubic Bézier curve segment, respectively.  $S_1$  represents the implied first  
16 control point of the smooth Bézier curve segment.  $S_2$  represents the specified control  
17 point of the smooth Bézier curve segment.  $I$  represents the inflection point around  
18 which control point  $S_1$  is derived from control point  $C_2$ .



19

20 The above diagram is generated with the following markup:

```
21 <Canvas RenderTransform="1.25,0,0,1.25,-40,20" >
22 <!-- Main Path -->
23 <Path Stroke="#000000" StrokeThickness="30" StrokeLineJoin="Round"
24 Data="M50,80 L100,80 C130,0 170,160 200,80 S270,160 300,80 L350,80"/>
25 <Path Stroke="#CCCCCC" StrokeThickness="2"
26 Data="M50,80 L100,80 C130,0 170,160 200,80 S270,160 300,80 L350,80"/>
27 <!-- C1 -->
28 <Path Stroke="#AAAAAA" StrokeThickness="1" Data="M 100,80 L 130,0" />
29 <Path Stroke="#0000CC" StrokeThickness="5" StrokeStartLineCap="Round"
30 StrokeEndLineCap="Round" Data="M 130,0 L 130,0" />
31 <Glyphs Fill="#000000" UnicodeString="C" OriginX="130" OriginY="15"
32 FontUri="..\Resources/Fonts/Verdana.ttf" FontRenderingEmSize="10"
33 />
34 <Glyphs Fill="#000000" UnicodeString="1" OriginX="138" OriginY="18"
```



```

1      FontUri="..\Resources/Fonts/Verdana.ttf" FontRenderingEmSize="6"
2      />
3      <!-- C2 -->
4      <Path Stroke="#AAAAAA" Data="M 200,80 L 170,160" />
5      <Path Stroke="#0000CC" StrokeThickness="5" StrokeStartLineCap="Round"
6          StrokeEndLineCap="Round" Data="M 170,160 L 170,160" />
7      <Glyphs Fill="#000000" UnicodeString="C" OriginX="175"
8          OriginY="175" FontUri="..\Resources/Fonts/Verdana.ttf"
9          FontRenderingEmSize="10" />
10     <Glyphs Fill="#000000" UnicodeString="2" OriginX="183"
11         OriginY="178" FontUri="..\Resources/Fonts/Verdana.ttf"
12         FontRenderingEmSize="6" />
13     <!-- S1 -->
14     <Path Stroke="#AAAAAA" StrokeThickness="2"
15         StrokeDashArray="0.75 0.75" Data="M 200,80 L 230,0" />
16     <Path Stroke="#0000CC" StrokeThickness="5" StrokeStartLineCap="Round"
17         StrokeEndLineCap="Round" Data="M 230,0 L 230,0" />
18     <Path Stroke="#FFFFFF" StrokeThickness="3" StrokeStartLineCap="Round"
19         StrokeEndLineCap="Round" Data="M 230,0 L 230,0" />
20     <Glyphs Fill="#000000" UnicodeString="S" OriginX="230" OriginY="15"
21         FontUri="..\Resources/Fonts/Verdana.ttf" FontRenderingEmSize="10"
22         />
23     <Glyphs Fill="#000000" UnicodeString="1" OriginX="238" OriginY="18"
24         FontUri="..\Resources/Fonts/Verdana.ttf" FontRenderingEmSize="6"
25         />
26     <!-- S2 -->
27     <Path Stroke="#AAAAAA" StrokeThickness="1" Data="M 300,80 L 270,160"
28         />
29     <Path Stroke="#0000CC" StrokeThickness="5" StrokeStartLineCap="Round"
30         StrokeEndLineCap="Round" Data="M 270,160 L 270,160" />
31     <Glyphs Fill="#000000" UnicodeString="S" OriginX="275" OriginY="175"
32         FontUri="..\Resources/Fonts/Verdana.ttf" FontRenderingEmSize="10"
33         />
34     <Glyphs Fill="#000000" UnicodeString="2" OriginX="283" OriginY="178"
35         FontUri="..\Resources/Fonts/Verdana.ttf" FontRenderingEmSize="6"
36         />
37     <!-- Inflection -->
38     <Path Stroke="#FFFFFF" StrokeThickness="3" StrokeStartLineCap="Round"
39         StrokeEndLineCap="Round" Data="M 200,80 L 200,80" />
40     <Glyphs Fill="#FFFFFF" UnicodeString="I" OriginX="203" OriginY="90"
41         FontUri="..\Resources/Fonts/Verdana.ttf" FontRenderingEmSize="10"
42         />
43     </Canvas>

```

44 *end example]*

#### 45 11.2.3.2 Relative Commands and Curve Control Points

46 When using relative (lowercase) commands with the abbreviated geometry syntax,  
 47 each control point and end point are individually specified relative to the start point  
 48 of that segment.

49 *Example 11-12. Relative commands and curves*

50 The following markup describes a simple shape using cubic Bézier curves:

```

51     <Path Stroke="#000000" Data="M 50,20 L 150,20 C 250,75 170,130 120,100
52         C 70,70 90,110 130,160 Q 0,150 50,20" />

```

53 This markup describes the same shape, using relative commands:

```

54     <Path Stroke="#000000" Data="M 50,20 l 100,0 c 100,55 20,110 -30,80
55         c -50,-30 -30,10 10,60 q -130,-10 -80,-140" />

```

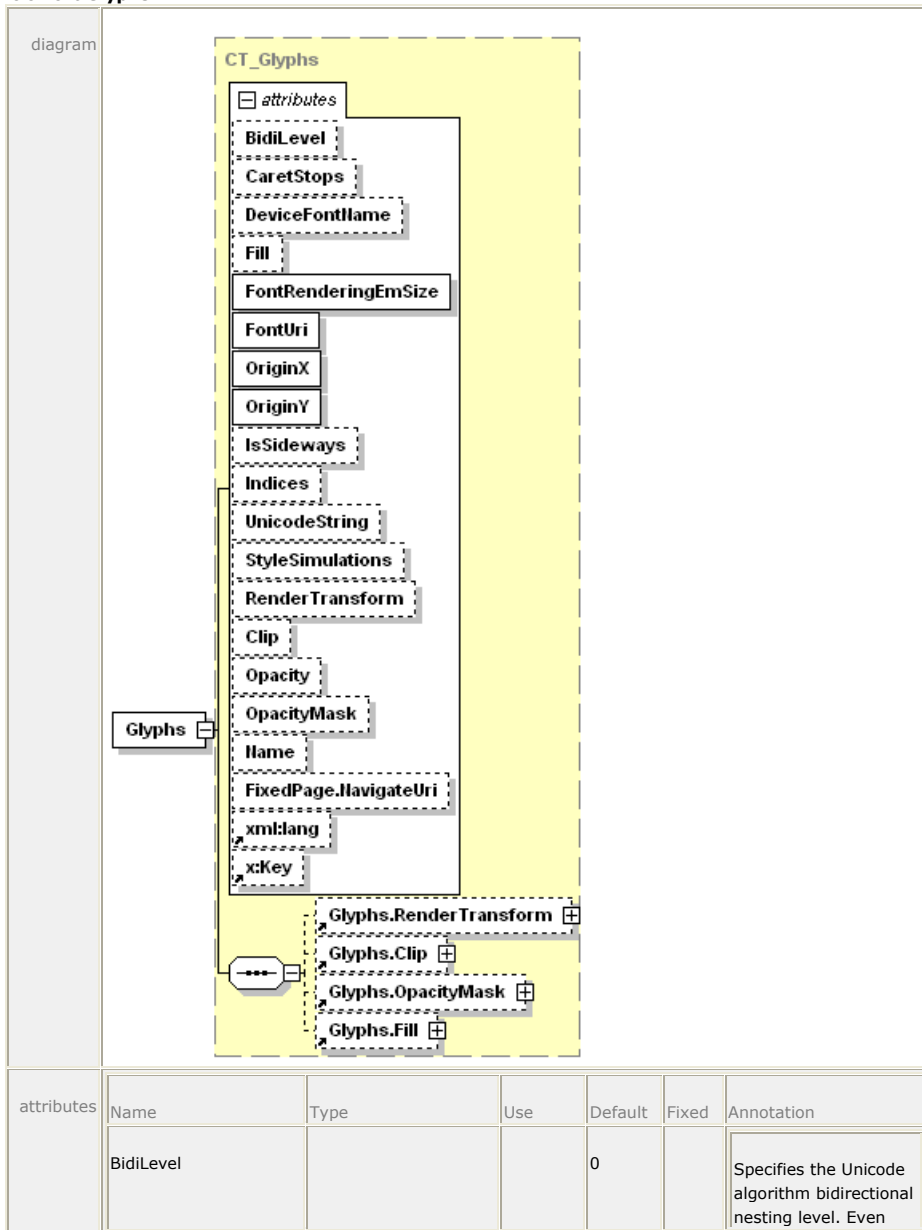
1 *end example]*

1 **12. Text**

2 A run of text sharing the same characteristics is represented by a <Glyphs>  
3 element. Text runs are broken by line advances and formatting changes. The set of  
4 properties on the <Glyphs> element allows for a complete description of the glyph  
5 characteristics, such as the fill and opacity, as well as clipping information. The  
6 <Glyphs> element allows specification of a Unicode string and supports bidirectional  
7 and vertical text.

1 **12.1 <Glyphs> Element**

2 element **Glyphs**



					<p>values imply left-to-right layout, odd values imply right-to-left layout. Right-to-left layout places the run origin at the right side of the first glyph, with positive advance widths (representing advances to the left) placing subsequent glyphs to the left of the previous glyph. Valid values range from 0 to 61, inclusive.</p>
CaretStops	<u>ST_CaretStops</u>				<p>Identifies the positions within the sequence of Unicode characters at which a text-selection tool may place a text-editing caret. Potential caret-stop positions are identified by their indices into the UTF-16 code units represented by the UnicodeString attribute value. When this attribute is missing, the text in the UnicodeString attribute value MUST be interpreted as having a caret stop between every Unicode UTF-16 code unit and at the beginning and end of the text [M5.1]. The value SHOULD indicate that the caret cannot stop in front of most combining marks or in front of the second UTF-16 code unit of UTF-16 surrogate pairs [S5.1].</p>
DeviceFontName	<u>ST_UnicodeString</u>				<p>Uniquely identifies a specific device font. The identifier is typically defined by a</p>

12.1 <Glyphs> Element

Text

					hardware vendor or font vendor.
Fill	<a href="#">ST_RscRefColor</a>				Describes the brush used to fill the shape of the rendered glyphs.
FontRenderingEmSize	<a href="#">ST_GEZero</a>	required			Specifies the font size in drawing surface units, expressed as a float in units of the effective coordinate space. A value of 0 results in no visible text.
FontUri	xs:anyURI	required			The URI of the physical font from which all glyphs in the run are drawn. The URI MUST reference a font contained in the package [M2.1]. If the physical font referenced is a TrueType Collection (containing multiple font faces), the fragment portion of the URI is a 0-based index indicating which font face of the TrueType Collection should be used.
OriginX	<a href="#">ST_Double</a>	required			Specifies the x coordinate of the first glyph in the run, in units of the effective coordinate space. The glyph is placed so that the leading edge of its advance vector and its baseline intersect with the point defined by the OriginX and OriginY attributes.
OriginY	<a href="#">ST_Double</a>	required			Specifies the y coordinate of the first glyph in the run, in

					units of the effective coordinate space. The glyph is placed so that the leading edge of its advance vector and its baseline intersect with the point defined by the OriginX and OriginY attributes.
IsSideways	<a href="#">ST_Boolean</a>		false		Indicates that a glyph is turned on its side, with the origin being defined as the top center of the unturned glyph.
Indices	<a href="#">ST_Indices</a>				Specifies a series of glyph indices and their attributes used for rendering the glyph run. If the UnicodeString attribute specifies an empty string (“” or “{}”) and the Indices attribute is not specified or is also empty, a consumer MUST generate an error [M5.2].
UnicodeString	<a href="#">ST_UnicodeString</a>				Contains the string of text rendered by the <Glyphs> element. The text is specified as Unicode code points.
StyleSimulations	<a href="#">ST_StyleSimulations</a>		None		Specifies a style simulation. Valid values are None, ItalicSimulation, BoldSimulation, and BoldItalicSimulation.
RenderTransform	<a href="#">ST_RscRefMatrix</a>				Establishes a new coordinate frame for the glyph run specified by the <Glyphs> element. The render transform affects clip, opacity mask, fill, x

12.1 <Glyphs> Element

Text

					origin, y origin, the actual shape of individual glyphs, and the advance widths. The render transform also affects the font size and values specified in the Indices attribute.
Clip	<a href="#">ST_RscRefAbbrGeomF</a>				Limits the rendered region of the element. Only portions of the <Glyphs> element that fall within the clip region (even partially clipped characters) produce marks on the page.
Opacity	<a href="#">ST_ZeroOne</a>		1.0		Defines the uniform transparency of the glyph element. Values range from 0 (fully transparent) to 1 (fully opaque), inclusive. Values outside of this range are invalid.
OpacityMask	<a href="#">ST_RscRef</a>				Specifies a mask of alpha values that is applied to the glyphs in the same fashion as the Opacity attribute, but allowing different alpha values for different areas of the element.
Name	<a href="#">ST_Name</a>				Contains a string value that identifies the current element as a named, addressable point in the document for the purpose of hyperlinking.
FixedPage.NavigateUri	xs:anyURI				Associates a hyperlink URI with the element. May be a relative reference or a URI that addresses a



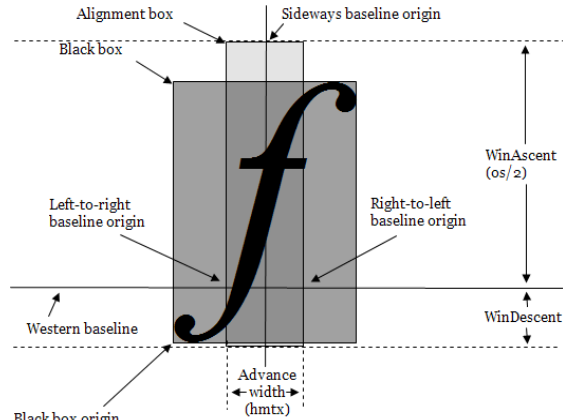
					resource that is internal to or external to the package.
	xml:lang				Specifies the default language used for the current element. The language is specified according to RFC 3066.
	x:Key				Specifies a name for a resource in a resource dictionary. x:Key MUST be present when the current element is defined in a resource dictionary. x:Key MUST NOT be specified outside of a resource dictionary [M5.3].
annotation	Represents a run of text from a single font.				

- 1 The <Glyphs> element represents a run of uniformly-formatted text from a single
- 2 font. It provides information necessary for accurate rendering and supports search
- 3 and selection features in viewing consumers.
- 4 If the Fill property is not specified, the <Glyphs> element has no visible effect.
- 5 Some properties of the <Glyphs> element are composable, meaning that the
- 6 markings rendered to the page are determined by a combination of the property and
- 7 all the like-named properties of the <Glyphs> element’s parent and ancestor
- 8 elements. For details, see 14, “Common Properties.”

1 **12.1.1 Glyph Metrics**

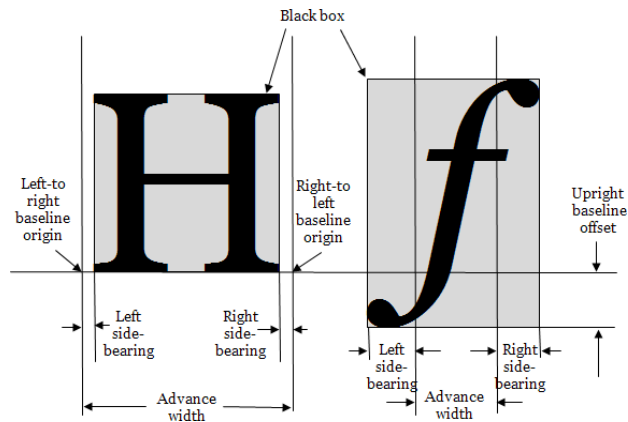
2 Each glyph defines metrics that specify how it aligns with other glyphs. The metrics  
 3 are illustrated below.

4 *Figure 12-1. Glyph metrics*



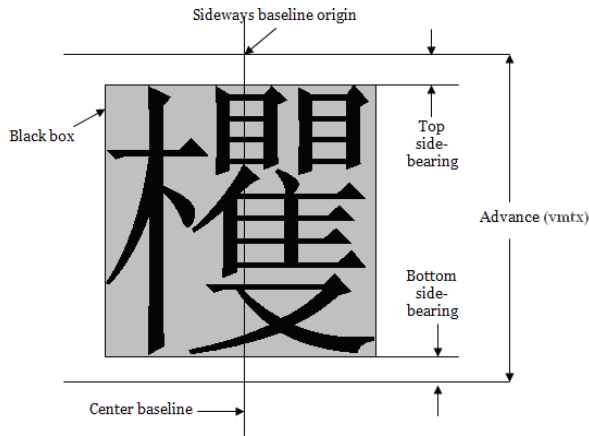
5 Black box origin

6 *Figure 12-2. Upright (usually horizontal) glyph metrics*



7

1 Figure 12-3. Sideways (usually vertical) glyph metrics



2

3 In general, glyphs within a font are either base glyphs or combining marks that may  
 4 be attached to base glyphs. Base glyphs usually have an advance width that is non-  
 5 zero, and a 0,0 offset vector. Combining marks usually have a zero advance width.  
 6 The offset vector can be used to adjust the position of a combining mark and,  
 7 therefore, may have a non-0,0 value for combining marks.

8 The position of each glyph in the glyph run is controlled by the following values:

- 9 • *Origin*. Each glyph is assumed to be given a nominal origin. For the first glyph  
 10 in the run, this is the origin of the run.
- 11 • *Advance Width*. The advance width for each glyph provides the origin of the  
 12 next glyph relative to the origin of the current glyph. The advance vector is  
 13 drawn in the direction of the run progression.
- 14 • *Glyph Offset (Base or Mark)*. The glyph offset vector adjusts the position of  
 15 this glyph relative to its nominal origin. The orientation of the glyph offset  
 16 vector is not affected by the value of the **IsSideways** attribute, but is affected  
 17 by the value of the **BidiLevel** attribute.

### 18 12.1.2 Mapping Code Units to Glyphs

19 A Unicode scalar value in a **UnicodeString** attribute is typically represented by a  
 20 single UTF-16 code unit and has a single corresponding glyph representation in the  
 21 font. More complex mapping scenarios are common in non-Latin scripts: a single  
 22 Unicode scalar value may map to two UTF-16 code units, multiple UTF-16 code units  
 23 may map to a single glyph, single UTF-16 code units may map to multiple glyphs  
 24 based on context, and multiple UTF-16 code units may map indivisibly to multiple  
 25 glyphs. In these cases, the clusters of UTF-16 code units are mapped using a cluster  
 26 map.

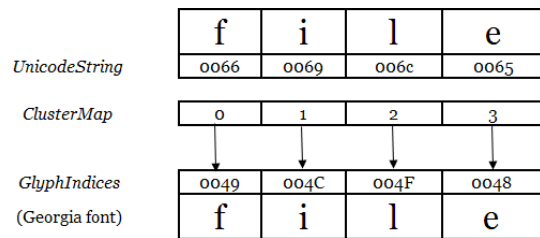
27 The cluster map contains one entry for each UTF-16 code unit in the **UnicodeString**  
 28 attribute. Each entry specifies the offset of the first glyph that represents the cluster  
 29 of UTF-16 code units.

1 **12.1.2.1 One-to-One Mappings**

2 When each UTF-16 code unit is represented by exactly one glyph, the cluster map  
 3 entries are 0, 1, 2, and so on.

4 *Example 12-1. One-to-one cluster map*

5 Each character in the word "file" is represented by a single glyph.



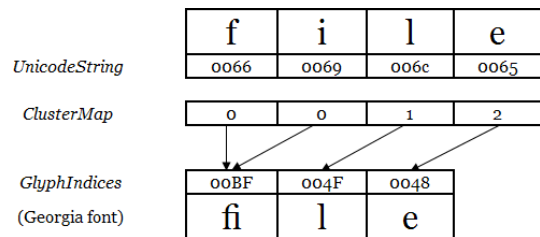
6  
 7 *end example]*

8 **12.1.2.2 Many-to-One Mappings**

9 When two or more UTF-16 code units map to a single glyph, the entries for those  
 10 UTF-16 code units specify the offset of that glyph in the glyph index buffer.

11 *Example 12-2. Many-to-one cluster map*

12 In the following mapping, the *f* and *i* characters are replaced by a ligature.



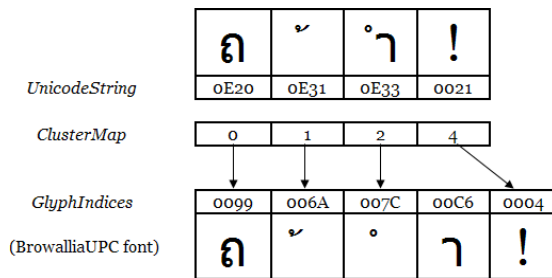
13  
 14 *end example]*

1 **12.1.2.3 One-to-Many Mappings**

2 When one UTF-16 code unit maps to two or more glyphs, the value in the cluster  
 3 map for that UTF-16 code unit references the first glyph in the **Indices** attribute that  
 4 represents that UTF-16 code unit.

5 *Example 12-3. One-to-many cluster map*

6 The Thai *Sara Am* character contains a part that sits on top of the previous base  
 7 character (the ring), and a part that sits to the right of the base character (the  
 8 hook). When Thai text is micro-justified, the hook is spaced apart from the base  
 9 character, while the ring remains on top of the base character. Many fonts encode  
 10 the ring and the hook as separate glyphs.

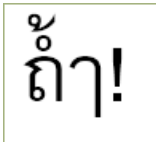


11

12 The markup appears as follows:

```
13 <Glyphs
14   FontUri="./Resources/Fonts/browau.ttf"
15   UnicodeString="&#xe20;&#xe49;&#xe33;&#x21;"
16   Indices="153;106,,16;(1:2)124;198;4"
17   OriginX="10" OriginY="60"
18   FontRenderingEmSize="70"
19   Fill="#000000"/>
```

20 The markup above is rendered as follows:



21

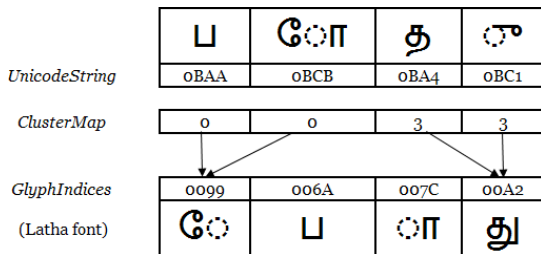
22 *end example]*

1 **12.1.2.4 Many-to-Many Mappings**

2 In some fonts, an indivisible group of UTF-16 code units for a character maps to  
 3 more than one glyph. This is common in fonts that support Indic scripts. When an  
 4 indivisible group of UTF-16 code units maps to one or more glyphs, the value in the  
 5 cluster map for each of the UTF-16 code units references the first glyph in the  
 6 **Indices** attribute representing that codepoint.

7 *Example 12-4. Many-to-many cluster map*

8 The following mapping shows the Unicode and glyph representations of a Tamil word  
 9 that has two glyph clusters. Each cluster has a base character and a combining  
 10 mark. The first pair of UTF-16 code units generates three glyphs because the  
 11 combining mark splits both sides of the base character. The second pair of UTF-16  
 12 code units is represented by a single glyph that incorporates the effect of the  
 13 combining mark.



14

15 The markup appears as follows:

```

16 <Glyphs
17   FontUri="../../Resources/Fonts/latha.ttf"
18   UnicodeString="#xbaa;#xbcb;#xba4;#xbc1;"
19   Indices="(2:3)94;76;88;(2:1)162"
20   OriginX="10" OriginY="120"
21   FontRenderingEmSize="40"
22   Fill="#000000"/>
    
```

23 The markup above is rendered as follows:



24

25 *end example]*

### 12.1.3 Indices Attribute

The <Glyphs> element MAY have an **Indices** attribute [M2.72]. The glyph specifications within the **Indices** attribute are OPTIONAL [M2.72]. The GlyphIndex portion of the **Indices** attribute MAY be used to specify a series of glyphs, complex character-to-glyph cluster mappings, or a combination of both [M2.72]. The **Indices** attribute MAY also include glyph placement information [M2.72].

Within the **Indices** attribute, each glyph specification is separated by a semicolon. The Indices attribute MUST adhere to the glyph specification syntax as follows [M2.72]:

```

GlyphIndices = *1GlyphMapping *( ";" *1GlyphMapping )
GlyphMapping = *1([ClusterMapping] GlyphIndex) [GlyphMetrics]
ClusterMapping = "(" ClusterCodeUnitCount [":" ClusterGlyphCount] ")"
ClusterCodeUnitCount = 1*DIGIT
ClusterGlyphCount = 1*DIGIT
GlyphIndex = *DIGIT
GlyphMetrics = "," *1AdvanceWidth ["," *1uOffset ["," vOffset]]
AdvanceWidth = ["+"] RealNum
uOffset = ["+" | "-"] RealNum
vOffset = ["+" | "-"] RealNum
RealNum = ((1*DIGIT ["." 1*DIGIT]) | ( "." 1*DIGIT)) [Exponent]
Exponent = *1( ("E"|"e") ("+"|"-") 1*DIGIT )

```

The sum of the code unit counts for all the GlyphMapping entries in the **Indices** attribute MUST NOT exceed the number of UTF-16 code units in the **UnicodeString** attribute if the **UnicodeString** attribute is specified and does not contain an empty value (" " or "{}"). If a ClusterMapping is not specified within a GlyphMapping entry, the code unit count is 1. If the **Indices** attribute specifies a GlyphIndex that does not exist in the font, the consumer MUST generate an error [M5.4]. If the **Indices** attribute is specified, the values provided MUST be used in preference to values determined from the **UnicodeString** attribute alone [M5.23].

Table 12-1. Glyph specifications

Name	Description
GlyphIndex	<p>Index of the glyph (16-bit) in the physical font. The entry MAY be empty [M2.72], in which case the glyph index is determined by looking up the UTF-16 code unit in the font character map table. If there is not a one-to-one mapping between code units and the glyph indices, this entry MUST be specified [M5.5].</p> <p>In cases where character-to-glyph mappings are not one-to-one, a cluster mapping specification precedes the glyph index (further described below).</p>
AdvanceWidth	<p>Advance width indicating placement for the subsequent glyph, relative to the origin of the current glyph. Measured in direction of advance as defined by the IsSideways and BidiLevel attributes. Base glyphs generally have a non-zero advance width and combining glyphs have a zero advance width.</p> <p>Advance width is measured in hundredths of the font em size. The default value is defined in the horizontal metrics font table (hmtx) if the IsSideways attribute is specified as false or the vertical metrics font table (vmtx) if the IsSideways attribute is specified as true. Advance width is a real number with units specified in</p>

---

hundredths of an em.

So that rounding errors do not accumulate, the advance MUST be calculated as the exact unrounded origin of the subsequent glyph minus the sum of the calculated (that is, rounded) advance widths of the preceding glyphs [M5.6].

The advance MUST be 0 or greater [M2.72]. The right-to-left writing direction can be specified using the BidilLevel attribute.

---

**uOffset, vOffset** Offset in the effective coordinate space relative to glyph origin to move this glyph ( $x$  offset for uOffset and  $-y$  offset for vOffset. The sign of vOffset is reversed from the direction of the  $y$  axis. A positive vOffset value shifts the glyph by a negative  $y$  offset and vice versa.). Used to attach marks to base characters. The value is added to the nominal glyph origin calculated using the advance width to generate the actual origin for the glyph. The setting of the IsSideways attribute does not change the interpretation of uOffset and vOffset.

Measured in hundredths of the font em size. The default offset values are 0.0,0.0. uOffset and vOffset are real numbers.

Base glyphs generally have a glyph offset of 0.0,0.0. Combining glyphs generally have an offset that places them correctly on top of the nearest preceding base glyph.

For left-to-right text, a positive uOffset value points to the right; for right-to-left text, a positive uOffset value points to the left.

---

1 *Example 12-1. Using indices to specify advance width*

2 The following **Indices** attribute specifies that the seventh glyph in the Unicode string  
3 has an advance width of 40:

4 

```
Indices = ";;;;;,40"
```

5 *end example]*

#### 6 **12.1.3.1 Specifying Character-to-Glyph Mappings**

7 A cluster map specification MAY precede the glyph specification for the first glyph of  
8 the cluster [M2.72].

9 Empty **Indices** attribute values indicate that the corresponding UTF-16 code unit  
10 within the Unicode string has a one-to-one relationship with the glyph index as  
11 specified by the character mapping table within the font.

12 Cluster maps that specify 0:n or n:0 mappings are invalid.

13 See the glyph specification syntax above for details of how to specify cluster maps.

14 *Table 12-2. Portions of the cluster specification*

Name	Description
ClusterCodeUnitCount	Number of UTF-16 code units that combine to form this cluster. One or more code units may be specified. Default value is 1.
ClusterGlyphCount	Number of glyph indices that combine to form this cluster.

---



---

One or more indices may be specified. Default value is 1.

---

1 *Example 12-2. Using the Indices attribute to specify glyph replacement for a cluster*

2 The following **Indices** attribute specifies that the sixth and seventh UTF-16 code  
3 units in the Unicode string should be replaced by a single glyph having an index of  
4 191:

5 `Indices = ";;;;(2:1)191"`

6 *end example]*

#### 7 **12.1.4 UnicodeString Attribute**

8 The **UnicodeString** attribute holds the array of Unicode scalar values that are  
9 represented by the current <Glyphs> element. Specifying a Unicode string is  
10 RECOMMENDED, as it supports searching, selection, and accessibility [S5.5]. If the  
11 Unicode string contains Unicode scalar values that require two UTF-16 code units, a  
12 cluster map with a many-to-one or many-to-many mapping **MUST** be specified for  
13 the values [M5.5].

14 The standard XML escaping mechanisms are used to specify XML-reserved  
15 characters. An additional mechanism **MUST** be used to escape a **UnicodeString**  
16 attribute value that begins with an open brace (“{”) [M5.7].

17 In order to use an open brace at the beginning of the Unicode string, it **MUST** be  
18 escaped with a prefix of “{” [M5.7]. If the **UnicodeString** attribute value starts  
19 with “{”, consumers **MUST** ignore those first two characters in processing the  
20 Unicode string and in calculating index positions for the characters of the Unicode  
21 string [M5.7].

22 If the **UnicodeString** attribute specifies an empty string (“” or “{”), and the  
23 **Indices** attribute is missing or is also empty, it **MUST** be treated as an error [M5.2].  
24 If the **UnicodeString** attribute contains a Unicode code unit that cannot be mapped  
25 to a glyph index via a cmap table in the font and there is no corresponding  
26 GlyphIndex entry in the **Indices** attribute, the consumer **MUST** display the .notdef  
27 glyph [M5.9].

28 Producers **MAY** include Unicode control marks in the Unicode string [O5.1]. Such  
29 marks include control codes, layout controls, invisible operators, deprecated format  
30 characters, variation selectors, non-characters, and specials, according to their  
31 definition within the Unicode specification. If producers include control marks in the  
32 Unicode string, they **SHOULD** include an **Indices** attribute to specify glyph indices  
33 and/or character-to-glyph mapping information for the control marks [S5.2]. In the  
34 absence of such information, consumers **MUST** treat Unicode control marks like  
35 ordinary characters and render the glyphs to which the Unicode control marks are  
36 mapped in the CMAP table [M5.10]. The resulting glyphs might produce an  
37 inappropriate rendering of the original Unicode string.

38 Producers **MAY** choose to generate **UnicodeString** attribute values that are not  
39 normalized by any Unicode-defined algorithm [O5.2]. Because advance-widths, glyph  
40 indices, and caret-stops are associated with the generated Unicode string, consumers  
41 **MUST NOT** normalize the **UnicodeString** attribute value to produce an internal  
42 representation [M5.11]. See §9.1.7.5, “Non-Standard Font Compatibility Encoding,”  
43 on page 35 for details and exceptions.

### 1 **12.1.5 StyleSimulations Attribute**

2 Synthetic style simulations can be applied to the shape of the glyphs by using the  
3 **StyleSimulations** attribute. Style simulations can be applied in addition to the  
4 designed style of a font. The default value for the **StyleSimulations** attribute is  
5 None, in which case the shapes of glyphs are not modified from their original design.

6 When the **StyleSimulations** value is specified as BoldSimulation, synthetic  
7 emboldening is applied by geometrically widening the strokes of glyphs by 1% of the  
8 em size, so that the centers of strokes remain at the same position. This leaves the  
9 baseline origin unmodified. The black box grows 1% all around for a total of 2%  
10 horizontal and 2% vertical. As a result, the character height and the advance width  
11 of each glyph are increased by 2% of the em size. Producers **MUST** lay out  
12 algorithmically emboldened glyphs using advance widths that are 2% of the em size  
13 larger than when not algorithmically emboldened [M5.12].

14 Consumers **MUST** implement the effect of algorithmic emboldening such that the  
15 black box of the glyph grows by 2% of the em size [M5.13]. When advance widths  
16 are omitted from the markup and the glyphs are algorithmically emboldened, the  
17 advance widths obtained from the horizontal metrics font table (if **IsSideways** is  
18 false) or the vertical metrics font table (if **IsSideways** is true) of the font **MUST** be  
19 increased by 2% of the em size [M5.13].

20 When **StyleSimulations** is specified as ItalicSimulation, synthetic italicizing is  
21 applied to glyphs with an **IsSideways** value of false by skewing the top edge of the  
22 alignment box of the character by 20° to the right, relative to the baseline of the  
23 character. Glyphs with an **IsSideways** value of true are italicized by skewing the  
24 right edge of the alignment box of the character by 20° down, relative to the  
25 baseline origin of the glyph. The character height and advance width are not  
26 modified. Producers **MUST** lay out algorithmically italicized glyphs using exactly the  
27 same advance widths as when not algorithmically italicized [M5.14].

28 When **StyleSimulations** is specified as BoldItalicSimulation, both BoldSimulation  
29 and ItalicSimulation are applied, in order.

### 30 **12.1.6 IsSideways Attribute**

31 Glyphs for text in vertical writing systems are normally represented by rotating the  
32 coordinate system and using the **IsSideways** attribute. <Glyphs> elements with the  
33 **IsSideways** attribute set to true will be rotated 90° counter-clockwise and placed so  
34 that the sideways baseline origin is coincident with the nominal origin of the  
35 character (within the glyph-local coordinate space), as modified by the offset vector  
36 in the **Indices** attribute. The advance vector places the nominal origin of the next  
37 character a distance along the direction of progression of the run. The direction of  
38 the advance vector is unaffected by **IsSideways**, however the method by which the  
39 size of the advance vector is chosen is different.

40 [Example: To represent a run of characters top to bottom on a page, a render  
41 transform can be used to rotate the <Glyphs> coordinate system 90° clockwise.  
42 **OriginX** and **OriginY** can be used to specify a position at the top of the column of  
43 text. Text from a vertical writing system can then be written using <Glyphs>  
44 elements with the **IsSideways** attribute set to true. The individual glyphs appear in  
45 the normal orientation because the rotation effected by the **IsSideway** attribute  
46 undoes the effect of the render transform. *end example*]

1 Text from horizontal writing systems can be included in the column by using  
 2 <Glyphs> elements without specifying **IsSideways**, or using a value of false for it.  
 3 The rotated coordinate system makes them appear top to bottom on the page, but  
 4 with the glyphs rotated to the right.

5 If alternate vertical character representations are available in the font, the producer  
 6 SHOULD use those and provide their glyph indices in the **Indices** attribute [S5.3].

#### 7 **12.1.6.1 Calculating Sideways Text Origin and Advance Width**

8 The formulas below describe the method used to calculate each glyph's nominal  
 9 origin, which is used for positioning the glyphs on the fixed page and for calculating  
 10 the default advance width for each glyph.

11 The origin is the top center of the unturned glyph. The x origin of the unturned glyph  
 12 is calculated to be exactly one-half the advance width of the glyph, as specified in  
 13 the horizontal metrics table of the font. This formula is expressed as follows (in  
 14 pseudocode):

```
15 TopOriginX = hmtx.advanceWidth[GlyphIndex] / 2
```

16 If the font is a CFF OpenType font, the y origin of the unturned glyph is determined  
 17 from the vertical origin (vorg) table for the font, which may be specified for a  
 18 particular glyph index but falls back to the default vertical origin if the glyph index is  
 19 not present in the vertical origin table. This formula is expressed as follows (in  
 20 pseudocode):

```
21 TopOriginY = vorg.vertOriginY[glyphIndex]
```

22 or:

```
23 TopOriginY = vorg.defaultVertOriginY
```

24 If the vertical origin table is not present, the glyph data (glyf) and vertical metrics  
 25 (vmtx) font tables are consulted. The glyph bounding box is retrieved from the glyph  
 26 data table and added to the top side-bearing for the glyph, specified in the vertical  
 27 metrics table. This formula is expressed as follows (in pseudocode):

```
28 TopOriginY = glyf.yMax[glyphIndex] + vmtx.topSideBearing[glyphIndex]
```

29 [Note: CFF fonts do not contain the glyf.yMax information; instead the yMax for each  
 30 glyph is computed by calculating the top of the glyph's bounding box from the CFF  
 31 charstring data. *end note*]

32 If the vertical metrics font table does not exist but the Windows-specific metrics  
 33 (OS/2) table does exist, the latter table is consulted and the sTypoAscender value is  
 34 used. This formula is expressed as follows (in pseudocode):

```
35 TopOriginY = os/2.sTypoAscender  

  36 Descender = abs(os/2.typoDescender)
```

37 In all other circumstances, the Ascender value from the horizontal header (hhea)  
 38 table is used. This formula is expressed as follows (in pseudocode):

```
39 TopOriginY = hhea.Ascender  

  40 Descender = abs(hhea.Descender)
```

41 Finally, the advance width for sideways text is computed as follows (in pseudocode),  
 42 unless specifically overridden by the **Indices** attribute:

1 `AdvanceWidth = TopOriginY + Descender`

2 **12.1.6.2 IsSideways and BidiLevel Effects on Glyph Positioning**

3 Right-to-left text (**BidiLevel** attribute value of 1) changes the direction of the  
4 AdvanceWidth and uOffset (horizontal offset) values of the **Indices** attribute, as well  
5 as the position of the glyph origin. Vertical text (**IsSideways** attribute set to true)  
6 changes the position of the glyph origin.

7 Producers MUST NOT specify text that is both right-to-left (**BidiLevel** attribute value  
8 of 1) and vertical (**IsSideways** attribute set to true) [M5.15].

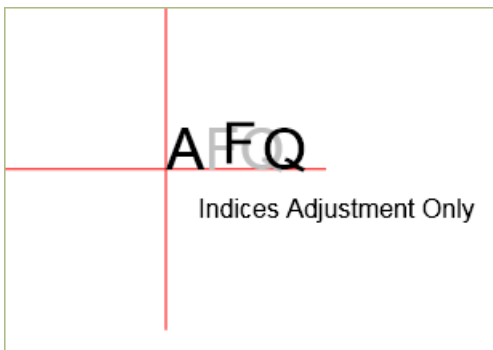
9 *Table 12-3. IsSideways and BidiLevel effects on origin placement*

<b>IsSideways</b>	<b>BidiLevel</b>	<b>Glyph origin</b>	<b>Direction of advance width and positive uOffset</b>
Horizontal (false)	Left-to-right (0)	Left end of horizontal advance vector along Latin baseline	To the right
Horizontal (false)	Right-to-left (1)	Right end of horizontal advance vector along Latin baseline	To the left
Vertical (true)	Left-to-right (0)	Top end of vertical advance vector through the glyph centerline	To the right
Vertical (true)	Right-to-left (1)	<i>Invalid combination</i>	

1 *Example 12-3. Text with positive uOffset and vOffset Indices values*

2 In this example, the position of the glyphs is shown relative to the origin shown at  
 3 the crossed lines centered at 100,100. The text in gray shows where this text would  
 4 be rendered without modification of the uOffset and vOffset value of the **Indices**  
 5 attributes.

```
6 <Glyphs Fill="#000000" FontRenderingEmSize="48"  
7 OriginX="100" OriginY="100"  
8 UnicodeString="AFQ"  
9 Indices=";,100,30,10;"  
10 FontUri=" ../Resources/Fonts/Arial.ttf" />
```



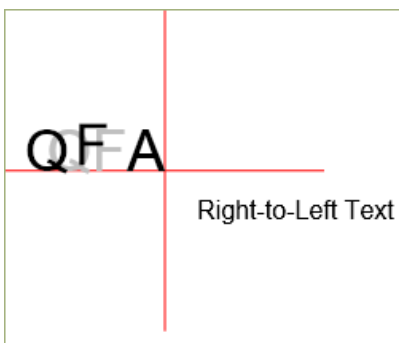
11

12 *end example]*

13 *Example 12-4. Right-to-left text (odd BidiLevel)*

14 The markup for this example matches the previous example, except the **BidiLevel**  
 15 attribute is set to 1. Note the change in the origin, and the reversal of the glyph  
 16 advance direction.

```
17 <Glyphs Fill="#000000" FontRenderingEmSize="48"  
18 OriginX="100" OriginY="100"  
19 UnicodeString="AFQ"  
20 Indices=";,100,30,10;"  
21 BidiLevel="1"  
22 FontUri=" ../Resources/Fonts/Arial.ttf" />
```



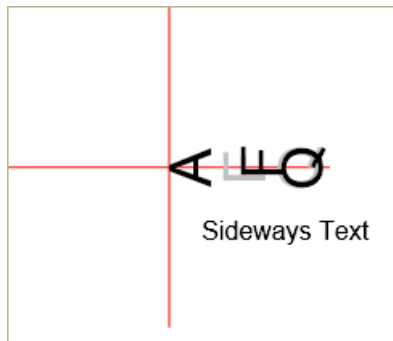
23

1 *end example]*

2 *Example 12-5. Sideways text (IsSideways set to true)*

3 This example shows the **IsSideways** attribute set to true. The **BidiLevel** must be  
 4 even when the **IsSideways** attribute is set to true [M5.15]. Note that the origin has  
 5 changed to be the top-center of the first glyph, with each glyph rotated 90° counter-  
 6 clockwise. The interpretation of the advance direction and **uOffset** and **vOffset** values  
 7 in the **Indices** attribute are otherwise unchanged.

```
8 <Glyphs Fill="#000000" FontRenderingEmSize="48"
9   OriginX="100" OriginY="100"
10  UnicodeString="AFQ"
11  Indices=";,100,30,10;"
12  IsSideways="true"
13  FontUri="../../Resources/Fonts/Arial.ttf" />
```



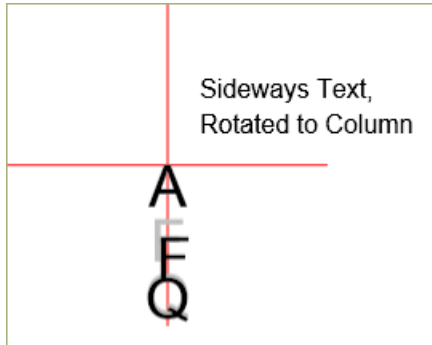
14

15 *end example]*

16 *Example 12-6. Vertical text*

17 The markup for this example matches the previous example, with the addition of a  
 18 render transformation to rotate and position the element as vertical text. For more  
 19 information on render transformations, see §14.4, "Positioning Content," on page  
 20 185.

```
21 <Glyphs Fill="#000000" FontRenderingEmSize="48"
22   OriginX="100" OriginY="100"
23   UnicodeString="AFQ"
24   Indices=";,100,30,10;"
25   IsSideways="true"
26   FontUri="../../Resources/Fonts/Arial.ttf"
27   RenderTransform="0,1,-1,0,200,0" />
```



1

2 *end example]*3 *Example 12-7. Japanese vertical text*

4 This example demonstrates a real-world usage of vertical text. Japanese text is  
 5 shown below where the text is read down each column, from right to left across the  
 6 page. The **IsSideways** attribute is set to true, thus rotating the each glyph 90°  
 7 counter-clockwise. Then, the **RenderTransform** attribute (see §14.4, "Positioning  
 8 Content," on page 185) rotates the overall block of text 90° clockwise to achieve the  
 9 final result of columns of text.

```

10 <Glyphs Fill="#000000" FontRenderingEmSize="24" OriginX="10" OriginY="10"
11     UnicodeString="これは、縦書きの日本語テキストが"
12     FontUri="..\Resources/Fonts/msmincho.ttf" IsSideways="true"
13     RenderTransform="0,1,-1,0,145,0"/>
14 <Glyphs Fill="#000000" FontRenderingEmSize="24" OriginX="10" OriginY="45"
15     UnicodeString="どのように列で書かれるかの例です。"
16     FontUri="..\Resources/Fonts/msmincho.ttf" IsSideways="true"
17     RenderTransform="0,1,-1,0,145,0"/>
18 <Glyphs Fill="#000000" FontRenderingEmSize="24" OriginX="10" OriginY="80"
19     UnicodeString="テキストは縦に読み、一行ずつ進みます。"
20     FontUri="..\Resources/Fonts/msmincho.ttf" IsSideways="true"
21     RenderTransform="0,1,-1,0,145,0"/>
22 <Glyphs Fill="#000000" FontRenderingEmSize="24" OriginX="10"
23     OriginY="115" UnicodeString="他の言語も縦書きで書かれます。"
24     FontUri="..\Resources/Fonts/msmincho.ttf" IsSideways="true"
25     RenderTransform="0,1,-1,0,145,0"/>

```

1 This markup is rendered as follows:

これは、縦書きの日本語テキストが  
どのよう列で書かれるかの例です。  
テキストは縦に読み、一行ずつ進みます。  
他の言語も縦書きで書かれます。

2

3 *end example*]

#### 4 **12.1.7 DeviceFontName Attribute**

5 Printer device fonts are specified by the **DeviceFontName** attribute. Device  
6 manufacturers define the values for this attribute. Producers SHOULD NOT produce  
7 markup that will result in different rendering between consumers using the  
8 embedded font to render and consumers using the device font to render [S5.4].

9 Consumers that understand the device font name MAY ignore the embedded font and  
10 use the device-resident version [O5.3]. By definition, a consumer “understands” a  
11 printer device font if it can unambiguously correlate the device font name to a set of  
12 font metrics resident on the device. If a consumer does not understand the specified  
13 device font name, it MUST render the embedded version of the font [M5.16].

14 When rendering a printer device font, consumers MUST use the **UnicodeString**  
15 attribute and ignore the glyph index components of the **Indices** attribute [M5.17].  
16 The consumer MUST still honor the advance width and x,y offset values present in  
17 the **Indices** attribute [M5.18].

18 For producers, a <Glyphs> element with a specified device font name MUST have  
19 exactly one **Indices** glyph per character in the **UnicodeString** attribute. Its **Indices**



1 attribute MUST NOT include any cluster specifications. If the **Indices** attribute  
 2 includes a cluster mapping, the consumer MUST NOT use the device font and MUST  
 3 render the embedded version of the font [M5.19]. This means that a device font  
 4 cannot be used for characters outside the basic multilingual plane (BMP).

5 If a device font name is specified, each of the <Glyphs> element's **Indices** glyphs  
 6 MUST include a specified advance width and MUST include specified x and y offset  
 7 values if they are non-zero [M5.20].

#### 8 **12.1.8 xml:lang Attribute**

9 XPS Document consumers may need to override the default language for a specific  
 10 run of glyphs, particularly in multilingual documents. The language defaults to the  
 11 value specified for the **xml:lang** attribute of the <FixedPage> element but MAY be  
 12 overridden by an **xml:lang** attribute on a <Glyphs> element [M2.72]. For larger  
 13 blocks of text, the producer MAY specify the **xml:lang** attribute on the <Canvas>  
 14 element [M2.72].

15 The language specified does not affect rendering of <Glyphs> elements, but it can  
 16 be used by consumers for searching or selecting text. For more information,  
 17 see §9.3.5, "Language," on page 47.

#### 18 **12.1.9 CaretStops Attribute**

19 The **CaretStops** attribute contains an array of Boolean bit-flags, which is  
 20 represented as a string of hexadecimal characters. The flags indicate whether it is  
 21 legal to place the caret before the corresponding UTF-16 code unit in the  
 22 **UnicodeString** attribute. ("Before" refers to a *logical* placement, not a *physical*  
 23 placement.) [*Example*: If the flag is set in right-to-left text, the caret can be placed  
 24 before (to the right of) that UTF-16 code unit. *end example*] The **CaretStops**  
 25 attribute includes a final flag for placement of the caret following the final UTF-16  
 26 code unit in the Unicode string.

27 Each hexadecimal character in the **CaretStops** value represents the flags for four  
 28 UTF-16 code units in the Unicode string, with the highest-order bit representing the  
 29 first UTF-16 code unit. Any unused bits in the last UTF-16 code unit must be 0.

30 If the **CaretStops** attribute is omitted, it is legal to place the caret before any of the  
 31 UTF-16 code units in the Unicode string. Therefore, omitting the **CaretStops**  
 32 attribute is equivalent to specifying a string that has all the bits set to 1. If there are  
 33 insufficient flags in the **CaretStops** string to correspond to all the UTF-16 code units  
 34 in the Unicode string, all remaining UTF-16 code units in the Unicode string MUST be  
 35 considered valid caret stops [M5.22].

36 *Example 12-8. Using the CaretStops attribute to determine a valid caret stop position*

37 Given the following attributes, the *m* in "example" is not a valid caret stop position:

```
38 UnicodeString = "This is an example string of text."  
39 CaretStops = "ffff"
```

40 *end example*]

1 **12.1.10 Optimizing Glyph Markup**

2 Markup details such as glyph indices and advance widths can be omitted from the  
3 markup under the circumstances described below. The following options allow  
4 optimization of commonly used simple scripts.

5 **12.1.10.1 Optimizing Glyph Indices Markup**

6 Glyph indices MAY be omitted from markup where *all* of the following are true  
7 [O5.4]:

- 8 • There is a one-to-one mapping between the positions of Unicode scalar values  
9 in the **UnicodeString** attribute and the positions of glyphs in the glyph string.
- 10 • The glyph index is the value in selected character mapping table of the font.

### 1 12.1.10.2 Optimizing Glyph Position Markup

2 Glyph advance width MAY be omitted from the markup in the following cases [O5.5]:

- 3 • For glyphs that have not been algorithmically emboldened, the desired  
4 advance width is the value listed in the horizontal metrics font table (if the  
5 **IsSideways** attribute value is false) or as calculated in section 12.1.6.1,  
6 "Calculating Sideways Text Origin and Advance Width" (if the **IsSideways**  
7 attribute value is true).
- 8 • For algorithmically emboldened glyphs, the desired advance width is exactly  
9 2% larger than the values in the horizontal metrics font table (if the  
10 **IsSideways** attribute value is false) or as calculated in section 12.1.6.1,  
11 "Calculating Sideways Text Origin and Advance Width" (if the **IsSideways**  
12 attribute value is true).

13 Glyph horizontal offset MAY be omitted from the markup when the offset is 0.0, and  
14 Glyph vertical offset MAY be omitted from the markup when the offset is 0.0 [O5.6].  
15 This is almost always true for base characters, and commonly true for combining  
16 marks in simple scripts. However, this is often false for combining marks in complex  
17 scripts such as Arabic and Indic.

### 18 12.1.11 Glyph Markup Examples

19 *Example 12–9. Basic italic font*

```
20 <Canvas>
21   <Glyphs
22     FontUri="../Resources/Fonts/Timesi.ttf"
23     FontRenderingEmSize="20"
24     OriginX="35"
25     OriginY="35"
26     UnicodeString="Basic italic font..."
27     Fill="#009900" />
28 </Canvas>
```

29 This text is rendered as follows:



*Basic italic font...*

30

31 *end example]*

32 *Example 12–10. Italic font using StyleSimulations attribute*

```
33 <Canvas>
34   <Glyphs
35     FontUri="../Resources/Fonts/Times.ttf"
36     FontRenderingEmSize="20"
37     StyleSimulations="ItalicSimulation"
38     OriginX="35"
39     OriginY="35"
40     UnicodeString="Simulated italic font..."
41     Fill="#009900" />
42 </Canvas>
```

43 This text is rendered as follows:

*Simulated italic font...*

1  
2 *end example]*

3 *Example 12-11. Kerning*

```

4 <Canvas>
5
6 <!-- "WAVE" without kerning -->
7
8 <Glyphs
9   OriginX="35"
10  OriginY="35"
11  UnicodeString="WAVE (no kerning)"
12  FontUri="../../Resources/Fonts/Times.ttf"
13  FontRenderingEmSize="20"
14  Fill="#009900" />
15
16 <!-- "WAVE" with kerning -->
17
18 <Glyphs
19   OriginX="35"
20   OriginY="70"
21   UnicodeString="WAVE (with kerning)"
22   Indices=",88;,59"
23   FontUri="../../Resources/Fonts/Times.ttf"
24   FontRenderingEmSize="20"
25   Fill="#009900" />
26
27 </Canvas>

```

28 This text is rendered as follows:

**WAVE (no kerning)**  
**WAVE (with kerning)**

29  
30 *end example]*

31 *Example 12-12. Ligatures*

```

32 <Canvas>
33
34 <!-- "Open file" without "fi" ligature -->
35
36 <Glyphs
37   OriginX="35"
38   OriginY="35"
39   UnicodeString="Open file (no ligature)"
40   FontUri="../../Resources/Fonts/Times.ttf"
41   FontRenderingEmSize="20"
42   Fill="#009900" />
43
44 <!-- "Open file" with "fi" ligature -->
45
46 <Glyphs
47   OriginX="35"

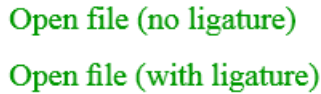
```

```

1      OriginY="70"
2      UnicodeString="Open file (with ligature)"
3      Indices=";;; (2:1)191"
4      FontUri="../Resources/Fonts/Times.ttf"
5      FontRenderingEmSize="20"
6      Fill="#009900" />
7
8 </Canvas>

```

9 This text is rendered as follows:



Open file (no ligature)  
Open file (with ligature)

10

11 *end example]*

12 *Example 12-13. Cluster maps*

```

13 <Canvas>
14
15 <!-- "ёжик в тумане" using pre-composed "ё" -->
16
17 <Glyphs
18   OriginX="35"
19   OriginY="35"
20   xml:lang="ru-RU"
21   UnicodeString="ёжик в тумане"
22   FontUri="../Resources/Fonts/Times.ttf"
23   FontRenderingEmSize="20"
24   Fill="#009900" />
25
26 <!-- "ёжик в тумане" using composition of "e" and diaeresis -->
27
28 <Glyphs
29   OriginX="35"
30   OriginY="70"
31   xml:lang="ru-RU"
32   UnicodeString="ёжик в тумане"
33   Indices="(1:2)72;142,0,-40"
34   FontUri="../Resources/Fonts/Times.ttf"
35   FontRenderingEmSize="20"
36   Fill="#009900" />
37
38 <!-- "ёжик в тумане" Forced rendering right-to-left showing
39   combining mark in logical order -->
40
41 <Glyphs
42   OriginX="155"
43   OriginY="105"
44   BidiLevel="1"
45   xml:lang="ru-RU"
46   UnicodeString="ёжик в тумане"
47   Indices="(1:2)72;142,0,-40"
48   FontUri="../Resources/Fonts/Times.ttf"
49   FontRenderingEmSize="20"
50   Fill="#009900" />
51
52 </Canvas>

```

1 This text is rendered as follows:

ЁЖИК В ТУМАНЕ

ЁЖИК В ТУМАНЕ

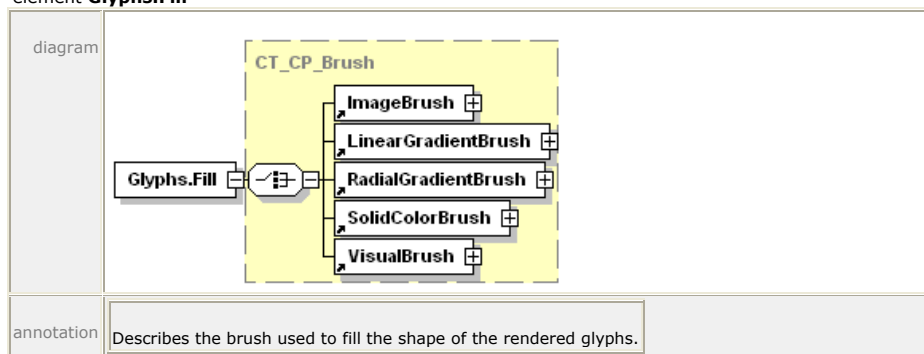
ЕНАМУТ В КИЖЁ

2

3 *end example]*

## 4 12.2 <Glyphs.Fill> Element

5 element **Glyphs.Fill**



6 The Fill property specifies the brush that fills a glyph. Any brush may be used.

7

## 1 13. Brushes

2 Brushes are used to paint the interior of the geometric shapes defined by a <Path>  
 3 element and the characters rendered with a <Glyphs> element. They are also used  
 4 to define the alpha-transparency mask in the <Canvas.OpacityMask>, <Path.OpacityMask>, and <Glyphs.OpacityMask> property elements.

6 All brushes are defined relative to a coordinate space. Most brushes (including image  
 7 brushes, visual brushes, linear gradient brushes, and radial gradient brushes) may  
 8 specify a coordinate-space transform, in which the Transform property is  
 9 concatenated with the current effective coordinate space to yield an effective  
 10 coordinate space local to the brush. For image brushes and visual brushes, the  
 11 viewport is transformed using the local effective render transform. For linear gradient  
 12 brushes, the start point and end point are transformed. For radial gradient brushes,  
 13 the ellipse defined by the center,  $x$  radius,  $y$  radius, and gradient origin is  
 14 transformed.

15 *Table 13–1. Brush types*

<b>Name</b>	<b>Description</b>
Solid color brush	Fills a region with a solid color
Image brush	Fills a region with an image
Visual brush	Fills a region with a drawing
Linear gradient brush	Fills a region with a linear gradient
Radial gradient brush	Fills a region with a radial gradient

1 **13.1 <SolidColorBrush> Element**

2 element **SolidColorBrush**

diagram	<p>The diagram shows a class <code>SolidColorBrush</code> with three attributes: <code>Opacity</code>, <code>x:Key</code>, and <code>Color</code>. A dashed box labeled <code>CT_SolidColorf</code> encloses the <code>Opacity</code>, <code>x:Key</code>, and <code>Color</code> attributes, indicating they are part of a specific configuration or context.</p>					
attributes	Name	Type	Use	Default	Fixed	Annotation
	Opacity	<a href="#">ST_ZeroOne</a>		1.0		Defines the uniform transparency of the brush fill. Values range from 0 (fully transparent) to 1 (fully opaque), inclusive. Values outside of this range are invalid.
	x:Key					Specifies a name for a resource in a resource dictionary. x:Key MUST be present when the current element is defined in a resource dictionary. x:Key MUST NOT be specified outside of a resource dictionary [M6.1].
	Color	<a href="#">ST_Color</a>	required			Specifies the color for filled elements. An sRGB color value specified as a 6-digit hexadecimal number (#RRGGBB) or an extended color.
annotation	Fills defined geometric regions with a solid color.					

3 The <SolidColorBrush> element is used to fill defined geometric regions with a solid  
 4 color. If there is an alpha component of the color, it is combined in a multiplicative  
 5 way with the corresponding **Opacity** attribute.

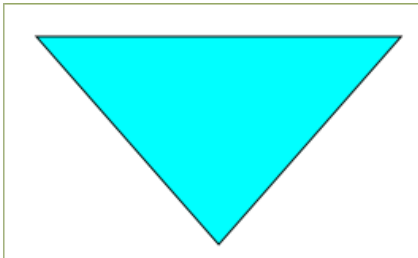


1 *Example 13-1. <SolidColorBrush> usage*

2 The following markup illustrates how a solid color brush fills a path.

```
3 <Path Stroke="#000000">  
4 <Path.Fill>  
5 <SolidColorBrush Color="#00FFFF" />  
6 </Path.Fill>  
7 <Path.Data>  
8 <PathGeometry>  
9 <PathFigure StartPoint="20,20" IsClosed="true">  
10 <PolyLineSegment Points="250,20 135,150" />  
11 </PathFigure>  
12 </PathGeometry>  
13 </Path.Data>  
14 </Path>
```

15 This markup is rendered as follows:



16

17 *end example]*

---

## 18 **13.2 <ImageBrush> Element**

19 element **ImageBrush**

<p>diagram</p>																																			
<p>attributes</p>	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> <th>Annotation</th> </tr> </thead> <tbody> <tr> <td>Opacity</td> <td><a href="#">ST_ZeroOne</a></td> <td></td> <td>1.0</td> <td></td> <td>Defines the uniform transparency of the brush fill. Values range from 0 (fully transparent) to 1 (fully opaque), inclusive. Values outside of this range are invalid.</td> </tr> <tr> <td>x:Key</td> <td></td> <td></td> <td></td> <td></td> <td>Specifies a name for a resource in a resource dictionary. x:Key MUST be present when the current element is defined in a resource dictionary. x:Key MUST NOT be specified outside of a resource dictionary [M6.2].</td> </tr> <tr> <td>Transform</td> <td><a href="#">ST_RscRefMatrix</a></td> <td></td> <td></td> <td></td> <td>Describes the matrix transformation applied to the coordinate space of the brush. The Transform property is concatenated with the current effective render transform to yield an effective render transform local to the brush. The viewport for the brush is transformed using the local effective render transform.</td> </tr> <tr> <td>Viewbox</td> <td><a href="#">ST_ViewBox</a></td> <td>required</td> <td></td> <td></td> <td>Specifies the position and dimensions of the brush's source content. Specifies four comma-</td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	Annotation	Opacity	<a href="#">ST_ZeroOne</a>		1.0		Defines the uniform transparency of the brush fill. Values range from 0 (fully transparent) to 1 (fully opaque), inclusive. Values outside of this range are invalid.	x:Key					Specifies a name for a resource in a resource dictionary. x:Key MUST be present when the current element is defined in a resource dictionary. x:Key MUST NOT be specified outside of a resource dictionary [M6.2].	Transform	<a href="#">ST_RscRefMatrix</a>				Describes the matrix transformation applied to the coordinate space of the brush. The Transform property is concatenated with the current effective render transform to yield an effective render transform local to the brush. The viewport for the brush is transformed using the local effective render transform.	Viewbox	<a href="#">ST_ViewBox</a>	required			Specifies the position and dimensions of the brush's source content. Specifies four comma-				
Name	Type	Use	Default	Fixed	Annotation																														
Opacity	<a href="#">ST_ZeroOne</a>		1.0		Defines the uniform transparency of the brush fill. Values range from 0 (fully transparent) to 1 (fully opaque), inclusive. Values outside of this range are invalid.																														
x:Key					Specifies a name for a resource in a resource dictionary. x:Key MUST be present when the current element is defined in a resource dictionary. x:Key MUST NOT be specified outside of a resource dictionary [M6.2].																														
Transform	<a href="#">ST_RscRefMatrix</a>				Describes the matrix transformation applied to the coordinate space of the brush. The Transform property is concatenated with the current effective render transform to yield an effective render transform local to the brush. The viewport for the brush is transformed using the local effective render transform.																														
Viewbox	<a href="#">ST_ViewBox</a>	required			Specifies the position and dimensions of the brush's source content. Specifies four comma-																														

					separated real numbers (x, y, Width, Height), where width and height are non-negative. The dimensions specified are relative to the image's physical dimensions expressed in units of 1/96". The corners of the viewbox are mapped to the corners of the viewport, thereby providing the default clipping and transform for the brush's source content.
Viewport	<a href="#">ST_ViewBox</a>	required			Specifies the region in the containing coordinate space of the prime brush tile that is (possibly repeatedly) applied to fill the region to which the brush is applied. Specifies four comma-separated real numbers (x, y, Width, Height), where width and height are non-negative. The alignment of the brush pattern is controlled by adjusting the x and y values.
TileMode	<a href="#">ST_TileMode</a>		None		Specifies how contents will be tiled in the filled region. Valid values are None, Tile, FlipX, FlipY, and FlipXY.
ViewboxUnits	<a href="#">ST_ViewUnits</a>	required		Absolute	Specifies the relationship of the viewbox coordinates to the containing coordinate space.
ViewportUnits	<a href="#">ST_ViewUnits</a>	required		Absolute	Specifies the relationship of the viewport coordinates to the containing coordinate space.
ImageSource	<a href="#">ST_UriCtxBmp</a>	required			Specifies the URI of an image resource or a combination of the URI of an image resource a color profile resource. See the Color clause for important details. The URI MUST refer to parts in the package [M2.1].
annotation	Fills a region with an image.				

- 1 The <ImageBrush> element is used to fill a region with an image. The image is
- 2 defined in a coordinate space specified by the resolution of the image. The image
- 3 MUST refer to a JPEG, PNG, TIFF 6.0, or Windows Media Photo image part within the

1 XPS Document package [M6.3]. For more information, see §9.1.5, "Image Parts," on  
2 page 24. A URI part name for the image is specified using the **ImageSource**  
3 attribute.

4 Image brushes share a number of tile-related properties with visual brushes. For  
5 details, see §13.4, "Common Attributes for Tiling Brushes," on page 136.

6 *Example 13-2. <ImageBrush> usage*

7 The following markup describes an image on a canvas.

```
8 <Canvas>  
9   <Path Stroke="#008000">  
10     <Path.Fill>  
11       <ImageBrush  
12         ImageSource="dog.jpg"  
13         TileMode="None"  
14         Viewbox="0,0,270,423"  
15         ViewboxUnits="Absolute"  
16         Viewport="25,25,125,185"  
17         ViewportUnits="Absolute" />  
18     </Path.Fill>  
19     <Path.Data>  
20       <PathGeometry>  
21         <PathFigure StartPoint="25,25" IsClosed="true">  
22           <PolyLineSegment Points="150,25 150,210 25,210" />  
23         </PathFigure>  
24       </PathGeometry>  
25     </Path.Data>  
26   </Path>  
27 </Canvas>
```

28 This markup produces the following results:



29

30 *end example]*

1 **13.3 <VisualBrush> Element**

2 element **VisualBrush**

diagram

attributes

Name	Type	Use	Default	Fixed	Annotation
Opacity	ST_ZeroOne		1.0		Defines the uniform transparency of the brush fill. Values range from 0 (fully transparent) to 1 (fully opaque), inclusive. Values outside of this range are invalid.
x:Key					Specifies a name for a resource in a resource dictionary. x:Key MUST be present when the current element is defined in a resource dictionary. x:Key MUST NOT be specified outside of a resource dictionary [M6.4].
Transform	ST_RscRefMatrix				Describes the matrix transformation applied to the coordinate space of the brush. The Transform property is concatenated with the current effective render transform to yield an effective render transform local to the brush. The viewport for the brush is

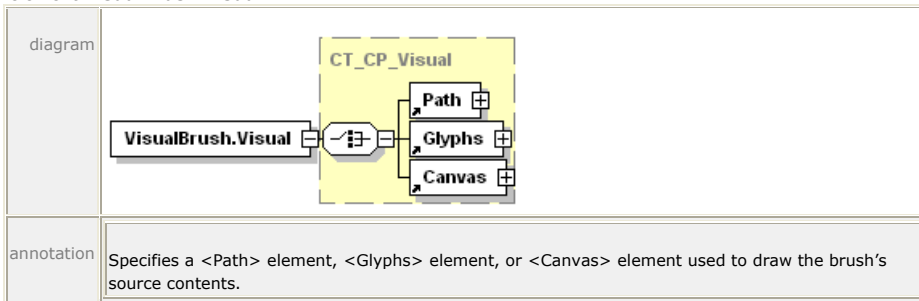
					transformed using that local effective render transform.
Viewbox	<a href="#">ST_ViewBox</a>	required			Specifies the position and dimensions of the brush's source content. Specifies four comma-separated real numbers (x, y, Width, Height), where width and height are non-negative. The viewbox defines the default coordinate system for the element specified in the <VisualBrush.Visual> property element. The corners of the viewbox are mapped to the corners of the viewport, thereby providing the default clipping and transform for the brush's source content.
Viewport	<a href="#">ST_ViewBox</a>	required			Specifies the region in the containing coordinate space of the prime brush tile that is (possibly repeatedly) applied to fill the region to which the brush is applied. Specifies four comma-separated real numbers (x, y, Width, Height), where width and height are non-negative. The alignment of the brush pattern is controlled by adjusting the x and y values.
TileMode	<a href="#">ST_TileMode</a>		None		Specifies how contents will be tiled in the filled region. Valid values are None, Tile, FlipX, FlipY, and FlipXY.
ViewboxUnits	<a href="#">ST_ViewUnits</a>	required		Absolute	Specifies the relationship of the viewbox coordinates to the containing coordinate space.
ViewportUnits	<a href="#">ST_ViewUnits</a>	required		Absolute	Specifies the relationship of the viewport coordinates to the containing coordinate space.
Visual	<a href="#">ST_RscRef</a>				Specifies resource reference to a <Path>, <Glyphs>, or <Canvas> element defined in a resource dictionary and used to draw the brush's source content.

annotation	Fills a region with a drawing. The drawing may be specified as either a child of the <VisualBrush> element, or as a resource reference. Drawing content is expressed using <Canvas>, <Path>, and <Glyphs> elements.
------------	---

- 1 The <VisualBrush> element is used to fill a region with a drawing. The drawing may  
 2 be specified as either a <VisualBrush.Visual> property element or as a resource  
 3 reference. Drawing content may include exactly one <Canvas>, <Path>, or  
 4 <Glyphs> element and that element's child and descendant elements.
- 5 Visual brushes share a number of tile-related properties with image brushes. For  
 6 details, see §13.4, "Common Attributes for Tiling Brushes," on page 136.

### 7 13.3.1 <VisualBrush.Visual> Element

8 element **VisualBrush.Visual**



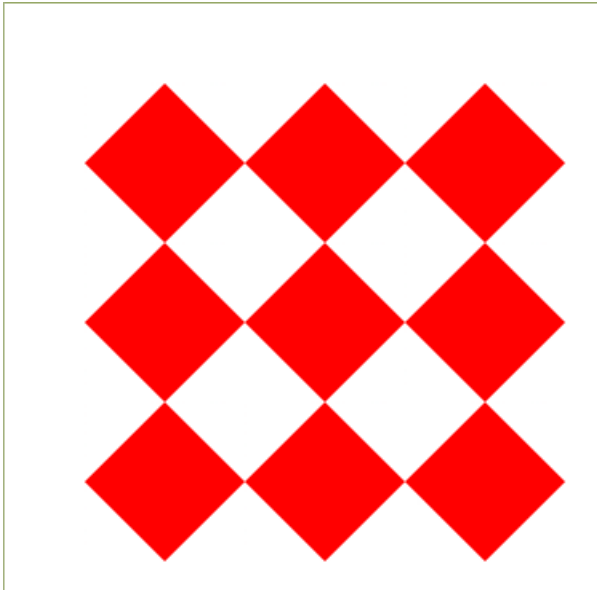
- 9 The <VisualBrush.Visual> property element contains markup that defines the  
 10 contents of a single visual brush tile. The tile can be used to fill the geometric region  
 11 to which the visual brush is applied. The <VisualBrush.Visual> property element  
 12 contains a single child element. For simple tiles, this can be a single <Path> or  
 13 <Glyphs> element. More complex visuals containing multiple <Path> and <Glyphs>  
 14 elements can be grouped within a <Canvas> child element.

1 *Example 13-3. <VisualBrush.Visual> usage*

```
2 <Path>
3   <Path.Fill>
4     <VisualBrush
5       Viewbox="0,0,1,1"
6       Viewport="50,50,100,100"
7       ViewportUnits="Absolute"
8       ViewboxUnits="Absolute"
9       TileMode="Tile">
10    <VisualBrush.Visual>
11      <Path>
12        <Path.Fill>
13          <SolidColorBrush Color="#FF0000" />
14        </Path.Fill>
15        <Path.Data>
16          <PathGeometry>
17            <PathFigure StartPoint="0,0.5" IsClosed="true">
18              <PolyLineSegment Points="0.5,0 1.0,0.5
19                0.5,1.0" />
20            </PathFigure>
21          </PathGeometry>
22        </Path.Data>
23      </Path>
24    </VisualBrush.Visual>
25  </VisualBrush>
26 </Path.Fill>
27 <Path.Data>
28   <PathGeometry>
29     <PathFigure StartPoint="50,50" IsClosed="true">
30       <PolyLineSegment Points="350,50 350,350 50,350" />
31     </PathFigure>
32   </PathGeometry>
33 </Path.Data>
34 </Path>
```



1 This markup produces the following result:



2

3 *end example]*

---

## 1 13.4 Common Attributes for Tiling Brushes

2 Image brushes and Visual brushes share certain tiling characteristics in common.  
 3 These characteristics are controlled by a common set of attributes described in the  
 4 table below.

5 *Table 13-2. Common attributes for <ImageBrush> and <VisualBrush> elements*

---

Name	Description
Viewbox	Specifies the region of the source content of the brush that is to be mapped to the viewport.
Viewport	Specifies the position and dimensions of the first brush tile. Subsequent tiles are positioned relative to this tile, as specified by the tile mode.
ViewboxUnits	Specifies the unit type for the Viewbox attribute. MUST have the value "Absolute" [M2.72].
ViewportUnits	Specifies the unit type for the Viewport attribute. MUST have the value "Absolute" [M2.72].
TileMode	Specifies how tiling is performed in the filled geometry. The value is optional, and defaults to "None" if no value is specified.

---

6 Both image brushes and visual brushes assume that the background of the brush  
 7 itself is initially transparent.

### 8 13.4.1 Viewbox, Viewport, ViewboxUnits, and ViewportUnits Attributes

9 The **Viewbox** attribute specifies the portion of a source image or visual to be  
 10 rendered to the page as a tile. The **Viewport** attribute specifies the dimensions and  
 11 location, in the effective coordinate space, of the initial tile that will be filled with the  
 12 specified image or visual fragment. In other words, the **Viewport** attribute defines  
 13 the initial tile whose origin (x and y values of the top left corner of the tile relative to  
 14 the current effective render transform) is specified by the first two parameters and  
 15 whose size (width and height values) is specified by the last two parameters. The tile  
 16 is then used to fill the geometry specified by the parent element according to the  
 17 **TileMode** attribute relative to the initial tile.

18 For images, the dimensions specified by the viewbox are expressed in units of 1/96".  
 19 The pixel coordinates in the source image are calculated as follows:

```
20 SourceLeft = HorizontalImageResolution * Viewbox.Left / 96
21 SourceTop = VerticalImageResolution * Viewbox.Top / 96
22 SourceWidth = HorizontalImageResolution * Viewbox.Width / 96
23 SourceHeight = VerticalImageResolution * Viewbox.Height / 96
```

24 The image resolution used is that specified in the header or tag information of the  
 25 image. If no resolution is specified, a default resolution of 96 dpi is assumed. The  
 26 coordinates of the upper-left corner of the image are 0,0.

27 The viewbox can specify a region larger than the image itself, including negative  
 28 values.

1 *Example 13-4. ViewboxUnits and ViewportUnits attribute usage*

2 The following markup contains an image brush:

```
3 <ImageBrush
4   ImageSource=" ../Resources/Images/tiger.jpg"
5   Viewbox="24,24,48,48"
6   ViewboxUnits="Absolute"
7   Viewport="96,96,192,192"
8   ViewportUnits="Absolute"
9   TileMode="None" />
```

10 Assuming the default fixed page coordinate system and that tiger.jpg specifies a  
11 resolution of 50 dpi and measures 100 pixels horizontally and 50 pixels vertically, the  
12 physical dimensions of the image are expressed (in units of 1/96") as 96 \* 100 / 50  
13 = 192 horizontal and 96 \* 50 / 50 = 96 vertical.

14 The viewbox uses a square starting at 24,24 (a quarter-inch from left and a quarter-  
15 inch from top) in the image, and extending for 48,48 (a half-inch to the right and a  
16 half-inch down) and scales it to a square starting at one inch from the left edge of  
17 the physical page and one inch from the top of the physical page and extending two  
18 inches to the right and two inches down. *end example*]

#### 19 **13.4.1.1 Viewbox and Viewport Examples**

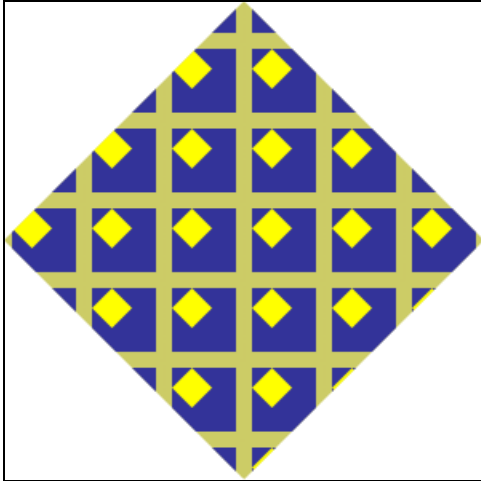
20 The following examples demonstrate how adjusting the viewbox and viewport can  
21 affect output.

22 *Example 13-5. Tiling brush base image and rendering*

23 The following markup describes a base image.

```
24 <!-- Draw background diamond to show where fill affects background -->
25 <Path Fill="#CCCC66" Data="M 150,0 L 300,150 L 150,300 L 0,150 Z" />
26 <Path Data="M 150,0 L 300,150 L 150,300 L 0,150 Z">
27   <Path.Fill>
28     <VisualBrush
29       Viewbox="0,0,1,1"
30       Viewport="150,75,50,50"
31       ViewboxUnits="Absolute"
32       ViewportUnits="Absolute"
33       TileMode="Tile">
34     <VisualBrush.Visual>
35       <Canvas>
36         <Path Fill="#333399" Data="M 0.1,0.1 L 0.9,0.1 L 0.9,0.9
37           L 0.1,0.9 Z" />
38         <Path Fill="#FFFF00" Data="M 0.1,0.35 L 0.35,0.1
39           L 0.6,0.35 L 0.35,0.6 Z" />
40       </Canvas>
41     </VisualBrush.Visual>
42   </VisualBrush>
43 </Path.Fill>
44 </Path>
```

1 This markup is rendered as follows:



2  
3 *end example]*

4 *Example 13-6. Tiling brush Viewport adjustments*

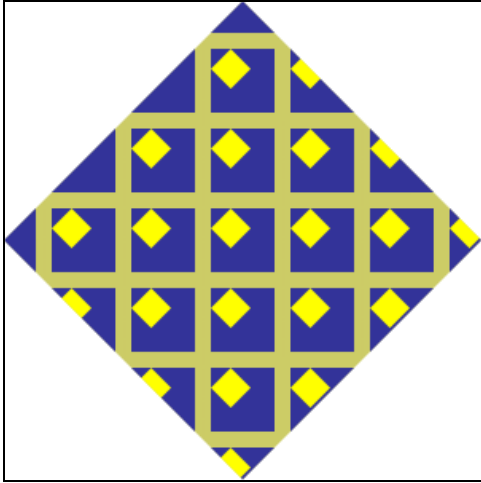
5 By adjusting the viewport, the position of the tiles within the image can be changed:

```

6 <!-- Draw background diamond to show where fill affects background -->
7 <Path Fill="#CCCC66" Data="M 150,0 L 300,150 L 150,300 L 0,150 Z" />
8 <Path Data="M 150,0 L 300,150 L 150,300 L 0,150 Z">
9   <Path.Fill>
10     <VisualBrush
11       Viewbox="0,0,1,1"
12       Viewport="125,125,50,50"
13       ViewboxUnits="Absolute"
14       ViewportUnits="Absolute"
15       TileMode="Tile">
16     <VisualBrush.Visual>
17       <Canvas>
18         <Path Fill="#333399" Data="M 0.1,0.1 L 0.9,0.1 L 0.9,0.9
19           L 0.1,0.9 Z" />
20         <Path Fill="#FFFF00" Data="M 0.1,0.35 L 0.35,0.1
21           L 0.6,0.35 L 0.35,0.6 Z" />
22       </Canvas>
23     </VisualBrush.Visual>
24   </VisualBrush>
25 </Path.Fill>
26 </Path>

```

1 This markup is rendered as follows:



2

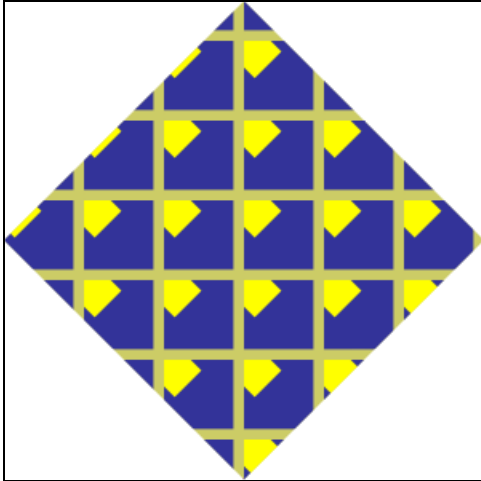
3 *end example]*

4 *Example 13-7. Tiling brush viewbox adjustments*

5 The following markup uses a smaller window on the viewbox to zoom in on each tile:

```
6 <!-- Draw background diamond to show where fill affects background -->
7 <Path Fill="#CCCC66" Data="M 150,0 L 300,150 L 150,300 L 0,150 Z" />
8 <Path Data="M 150,0 L 300,150 L 150,300 L 0,150 Z">
9   <Path.Fill>
10     <VisualBrush
11       Viewbox="0.25,0.25,0.75,0.75"
12       Viewport="150,75,50,50"
13       ViewboxUnits="Absolute"
14       ViewportUnits="Absolute"
15       TileMode="Tile">
16       <VisualBrush.Visual>
17         <Canvas>
18           <Path Fill="#333399" Data="M 0.1,0.1 L 0.9,0.1 L 0.9,0.9
19             L 0.1,0.9 Z" />
20           <Path Fill="#FFFF00" Data="M 0.1,0.35 L 0.35,0.1
21             L 0.6,0.35 L 0.35,0.6 Z" />
22         </Canvas>
23       </VisualBrush.Visual>
24     </VisualBrush>
25   </Path.Fill>
26 </Path>
```

1 This markup is rendered as follows:



2  
3 *end example]*

4 *Example 13-8. Image brush with a Viewbox larger than the image*

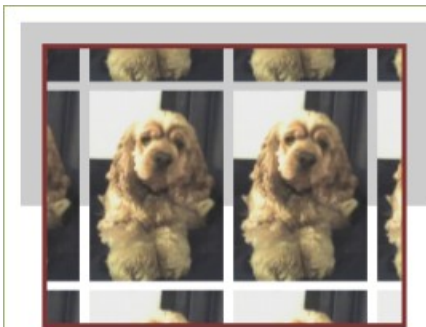
5 An image brush can specify a tile with the **Viewbox** attribute that exceeds the size  
6 of the image it uses, including negative values, as shown below.

```

7 <Path Fill="#CCCCCC" Data="M 10,10 L 265,10 L 265,125 L 10,125 Z" />
8 <Path Stroke="#803333" StrokeThickness="3"
9   Data="M 25,25 L 250,25 L 250,200 L 25,200 Z">
10   <Path.Fill>
11     <ImageBrush ImageSource="./Resources/Images/dog.jpg"
12       TileMode="Tile"
13       Viewbox="-10,-10,290,443" ViewboxUnits="Absolute"
14       Viewport="50,50,90,125" ViewportUnits="Absolute" />
15   </Path.Fill>
16 </Path>

```

17 This markup is rendered as follows. Note that the area around the image is  
18 transparent, revealing the underlying path between the tiles.



19

1 *end example]*

2 **13.4.2 TileMode Attribute**

3 Valid values for the **TileMode** attribute are None, Tile, FlipX, FlipY, and FlipXY.

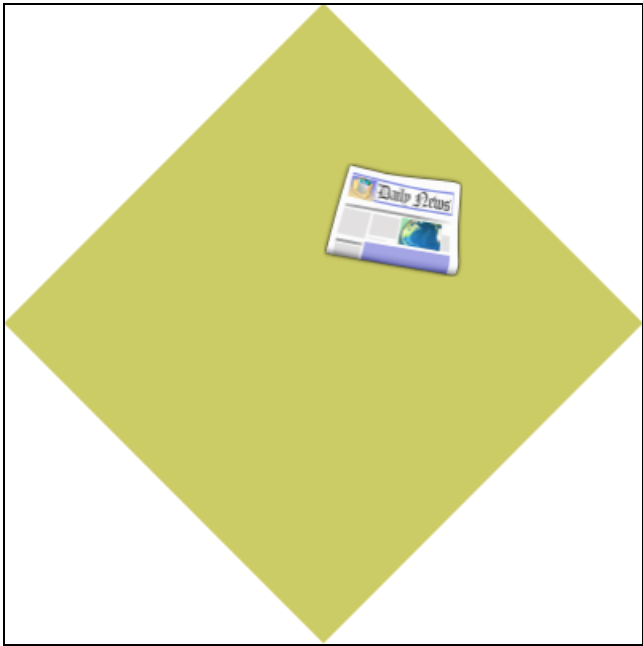
4 **13.4.2.1 None**

5 In this mode, only the single base tile is drawn. The remaining area is left  
6 transparent.

7 *Example 13-9. Image brush with TileMode value of None*

```
8 <!-- Draw background diamond to show where fill affects background -->  
9 <Path Fill="#CCCC66" Data="M 200,0 L 400,200 L 200,400 L 0,200 Z" />  
10 <Path Data="M 200,0 L 400,200 L 200,400 L 0,200 Z">  
11 <Path.Fill>  
12 <ImageBrush  
13 <ImageSource="newspaper.png"  
14 <Viewbox="0,0,350,284"  
15 <Viewport="200,100,87,71"  
16 <ViewportUnits="Absolute"  
17 <ViewboxUnits="Absolute"  
18 <TileMode="None" />  
19 </Path.Fill>  
20 </Path>
```

21 This markup is rendered as follows:



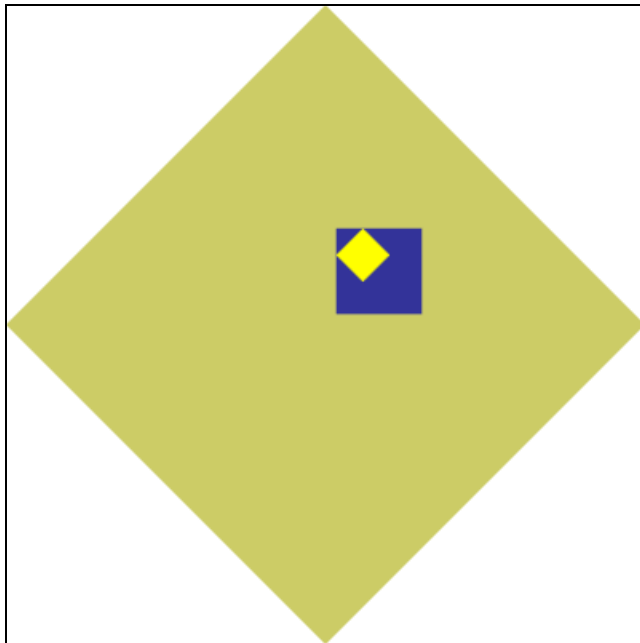
22

23 *end example]*

1 *Example 13-10. Visual brush with TileMode value of None*

```
2 <!-- Draw background diamond to show where fill affects background -->
3 <Path Fill="#CCCC66" Data="M 200,0 L 400,200 L 200,400 L 0,200 Z" />
4 <Path Data="M 200,0 L 400,200 L 200,400 L 0,200 Z">
5   <Path.Fill>
6     <VisualBrush
7       Viewbox="0,0,1,1"
8       Viewport="200,133,67,67"
9       ViewboxUnits="Absolute"
10      ViewportUnits="Absolute"
11      TileMode="None">
12       <VisualBrush.Visual>
13         <Canvas>
14           <Path Fill="#333399" Data="M 0.1,0.1 L 0.9,0.1 L 0.9,0.9
15             L 0.1,0.9 Z" />
16           <Path Fill="#FFFF00" Data="M 0.1,0.35 L 0.35,0.1
17             L 0.6,0.35 L 0.35,0.6 Z" />
18         </Canvas>
19       </VisualBrush.Visual>
20     </VisualBrush>
21   </Path.Fill>
22 </Path>
```

23 This markup is rendered as follows:



24

25 *end example]*



1 **13.4.2.2 Tile**

2 In this mode, the base tile is drawn and the remaining area is filled by repeating the  
3 base tile such that the right edge of each tile abuts the left edge of the next, and the  
4 bottom edge of each tile abuts the top edge of the next.

5 *Example 13-11. Image brush with a TileMode value of Tile*

```
6 <!-- Draw background diamond to show where fill affects background -->  
7 <Path Fill="#CCCC66" Data="M 200,0 L 400,200 L 200,400 L 0,200 Z" />  
8 <Path Data="M 200,0 L 400,200 L 200,400 L 0,200 Z">  
9 <Path.Fill>  
10 <ImageBrush  
11 <ImageSource="newspaper.png"  
12 Viewbox="0,0,350,284"  
13 Viewport="200,100,87,71"  
14 ViewportUnits="Absolute"  
15 ViewboxUnits="Absolute"  
16 TileMode="Tile" />  
17 </Path.Fill>  
18 </Path>
```

19 This markup is rendered as follows:



20

21 *end example]*

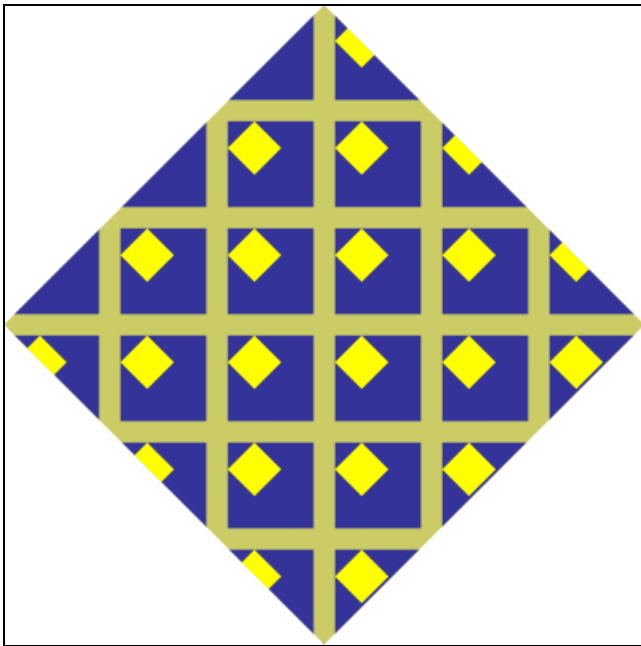
1 *Example 13-12. Visual brush with a TileMode value of Tile*

```

2 <!-- Draw background diamond to show where fill affects background -->
3 <Path Fill="#CCCC66" Data="M 200,0 L 400,200 L 200,400 L 0,200 Z" />
4 <Path Data="M 200,0 L 400,200 L 200,400 L 0,200 Z">
5   <Path.Fill>
6     <VisualBrush
7       Viewbox="0,0,1,1"
8       Viewport="200,133,67,67"
9       ViewboxUnits="Absolute"
10      ViewportUnits="Absolute"
11      TileMode="Tile">
12       <VisualBrush.Visual>
13         <Canvas>
14           <Path Fill="#333399" Data="M 0.1,0.1 L 0.9,0.1 L 0.9,0.9
15             L 0.1,0.9 Z" />
16           <Path Fill="#FFFF00" Data="M 0.1,0.35 L 0.35,0.1
17             L 0.6,0.35 L 0.35,0.6 Z" />
18         </Canvas>
19       </VisualBrush.Visual>
20     </VisualBrush>
21   </Path.Fill>
22 </Path>

```

23 This markup is rendered as follows:



24

25 *end example]*

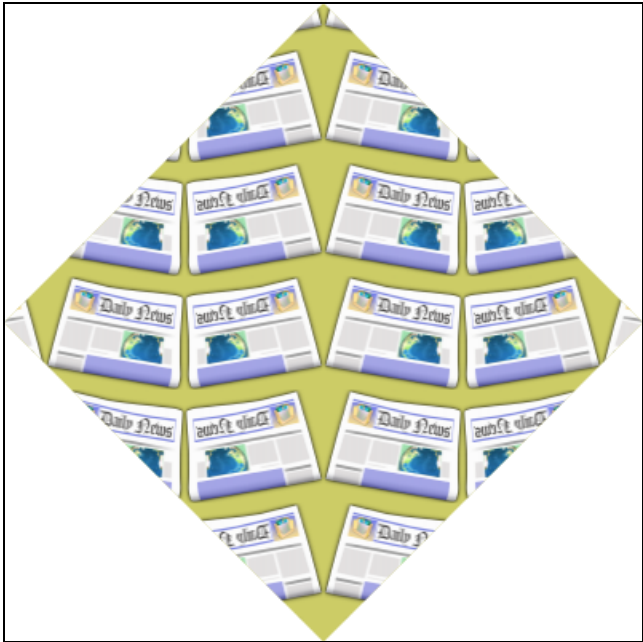
1 **13.4.2.3 FlipX**

2 The tile arrangement is similar to the Tile tile mode, but alternate columns of tiles  
3 are flipped horizontally. The base tile is positioned as specified by the viewport. Tiles  
4 in the columns to the left and right of this tile are flipped horizontally.

5 *Example 13-13. Image brush with a TileMode value of FlipX*

```
6 <!-- Draw background diamond to show where fill affects background -->  
7 <Path Fill="#CCCC66" Data="M 200,0 L 400,200 L 200,400 L 0,200 Z" />  
8 <Path Data="M 200,0 L 400,200 L 200,400 L 0,200 Z">  
9 <Path.Fill>  
10 <ImageBrush  
11 ImageSource="newspaper.png"  
12 Viewbox="0,0,350,284"  
13 Viewport="200,100,87,71"  
14 ViewportUnits="Absolute"  
15 ViewboxUnits="Absolute"  
16 TileMode="FlipX" />  
17 </Path.Fill>  
18 </Path>
```

19 This markup is rendered as follows:



20

21 *end example]*

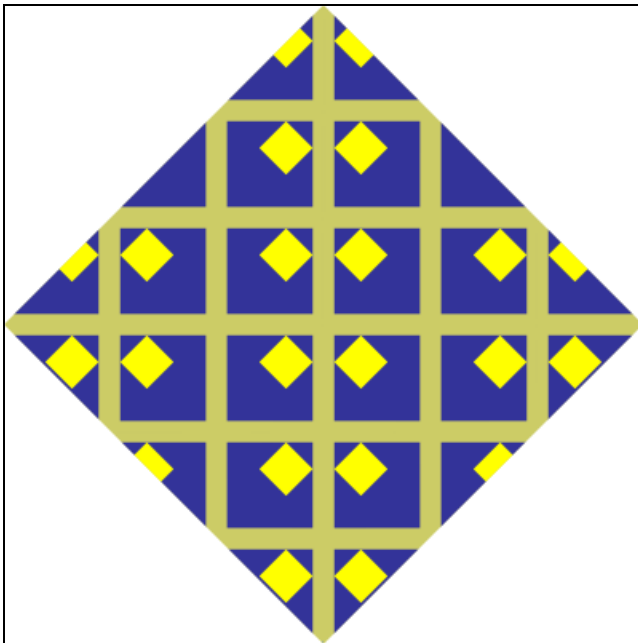
1 *Example 13-14. Visual brush with a TileMode value of FlipX*

```

2 <!-- Draw background diamond to show where fill affects background -->
3 <Path Fill="#CCCC66" Data="M 200,0 L 400,200 L 200,400 L 0,200 Z" />
4 <Path Data="M 200,0 L 400,200 L 200,400 L 0,200 Z">
5   <Path.Fill>
6     <VisualBrush
7       Viewbox="0,0,1,1"
8       Viewport="200,133,67,67"
9       ViewboxUnits="Absolute"
10      ViewportUnits="Absolute"
11      TileMode="FlipX">
12       <VisualBrush.Visual>
13         <Canvas>
14           <Path Fill="#333399" Data="M 0.1,0.1 L 0.9,0.1 L 0.9,0.9
15             L 0.1,0.9 Z" />
16           <Path Fill="#FFFF00" Data="M 0.1,0.35 L 0.35,0.1
17             L 0.6,0.35 L 0.35,0.6 Z" />
18         </Canvas>
19       </VisualBrush.Visual>
20     </VisualBrush>
21   </Path.Fill>
22 </Path>

```

23 This markup is rendered as follows:



24

25 *end example]*

1 **13.4.2.4 FlipY**

2 The tile arrangement is similar to the Tile tile mode, but alternate rows of tiles are  
3 flipped vertically. The base tile is positioned as specified by the viewport. Rows  
4 above and below are flipped vertically.

5 *Example 13-15. Image brush with a TileMode value of FlipY*

```
6 <!-- Draw background diamond to show where fill affects background -->  
7 <Path Fill="#CCCC66" Data="M 200,0 L 400,200 L 200,400 L 0,200 Z" />  
8 <Path Data="M 200,0 L 400,200 L 200,400 L 0,200 Z">  
9   <Path.Fill>  
10     <ImageBrush  
11       ImageSource="newspaper.png"  
12       Viewbox="0,0,350,284"  
13       Viewport="200,100,87,71"  
14       ViewportUnits="Absolute"  
15       ViewboxUnits="Absolute"  
16       TileMode="FlipY" />  
17   </Path.Fill>  
18 </Path>
```

19 This markup is rendered as follows:



20

21 *end example]*

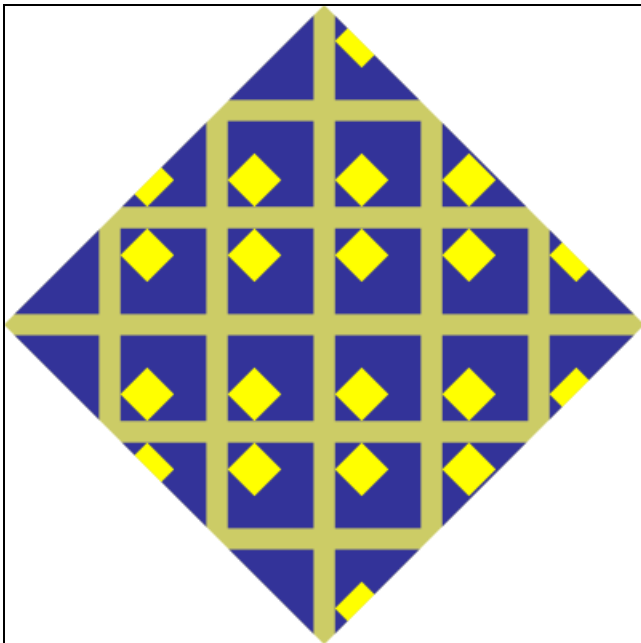
1 *Example 13-16. Visual Brush with a TileMode value of FlipY*

```

2 <!-- Draw background diamond to show where fill affects background -->
3 <Path Fill="#CCCC66" Data="M 200,0 L 400,200 L 200,400 L 0,200 Z" />
4 <Path Data="M 200,0 L 400,200 L 200,400 L 0,200 Z">
5   <Path.Fill>
6     <VisualBrush
7       Viewbox="0,0,1,1"
8       Viewport="200,133,67,67"
9       ViewboxUnits="Absolute"
10      ViewportUnits="Absolute"
11      TileMode="FlipY">
12       <VisualBrush.Visual>
13         <Canvas>
14           <Path Fill="#333399" Data="M 0.1,0.1 L 0.9,0.1 L 0.9,0.9
15             L 0.1,0.9 Z" />
16           <Path Fill="#FFFF00" Data="M 0.1,0.35 L 0.35,0.1
17             L 0.6,0.35 L 0.35,0.6 Z" />
18         </Canvas>
19       </VisualBrush.Visual>
20     </VisualBrush>
21   </Path.Fill>
22 </Path>

```

23 This markup is rendered as follows:



24

25 *end example]*

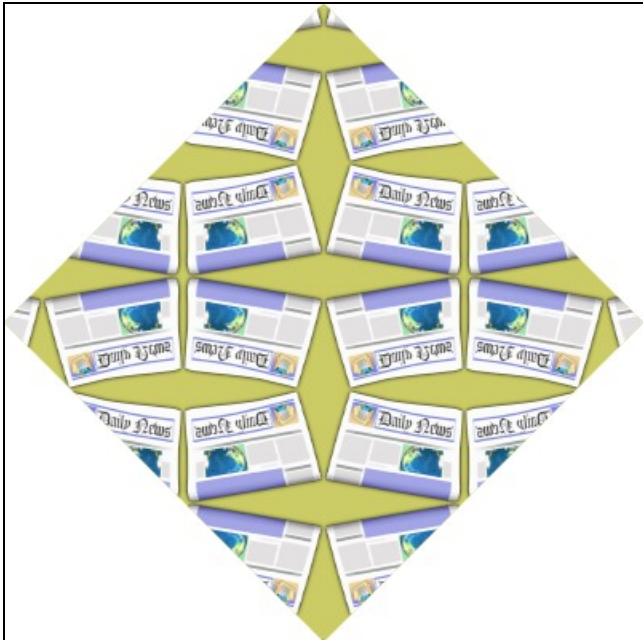
1 **13.4.2.5 FlipXY**

2 The tile arrangement is similar to the Tile mode, but alternate columns of tiles  
3 are flipped horizontally and alternate rows of tiles are flipped vertically. The base tile  
4 is positioned as specified by the viewport.

5 *Example 13-17. Image brush with a TileMode value of FlipXY*

```
6 <!-- Draw background diamond to show where fill affects background -->  
7 <Path Fill="#CCCC66" Data="M 200,0 L 400,200 L 200,400 L 0,200 Z" />  
8 <Path Data="M 200,0 L 400,200 L 200,400 L 0,200 Z">  
9 <Path.Fill>  
10 <ImageBrush  
11 <ImageSource="newspaper.png"  
12 Viewbox="0,0,350,284"  
13 Viewport="200,100,87,71"  
14 ViewportUnits="Absolute"  
15 ViewboxUnits="Absolute"  
16 TileMode="FlipXY" />  
17 </Path.Fill>  
18 </Path>
```

19 This markup is rendered as follows:



20

21 *end example]*

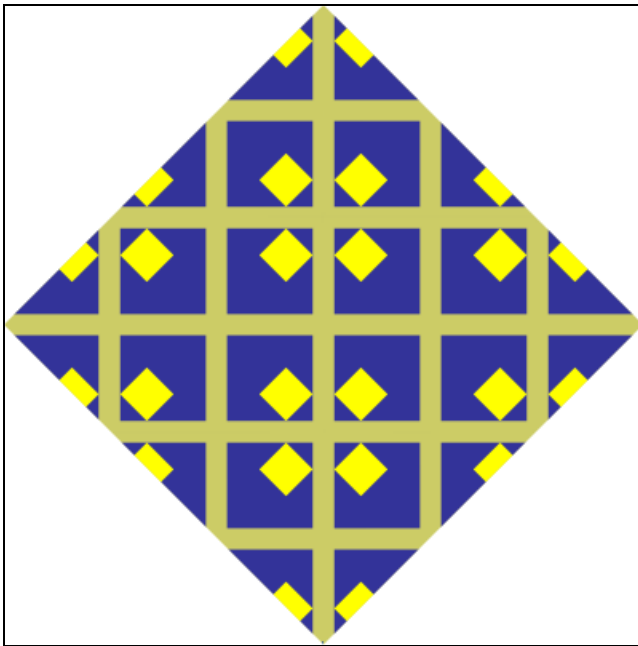
1 *Example 13-18. Visual brush with a TileMode value of FlipXY*

```

2 <!-- Draw background diamond to show where fill affects background -->
3 <Path Fill="#CCCC66" Data="M 200,0 L 400,200 L 200,400 L 0,200 Z" />
4 <Path Data="M 200,0 L 400,200 L 200,400 L 0,200 Z">
5   <Path.Fill>
6     <VisualBrush
7       Viewbox="0,0,1,1"
8       Viewport="200,133,67,67"
9       ViewboxUnits="Absolute"
10      ViewportUnits="Absolute"
11      TileMode="FlipXY">
12       <VisualBrush.Visual>
13         <Canvas>
14           <Path Fill="#333399" Data="M 0.1,0.1 L 0.9,0.1 L 0.9,0.9
15             L 0.1,0.9 Z" />
16           <Path Fill="#FFFF00" Data="M 0.1,0.35 L 0.35,0.1
17             L 0.6,0.35 L 0.35,0.6 Z" />
18         </Canvas>
19       </VisualBrush.Visual>
20     </VisualBrush>
21   </Path.Fill>
22 </Path>

```

23 This markup is rendered as follows:



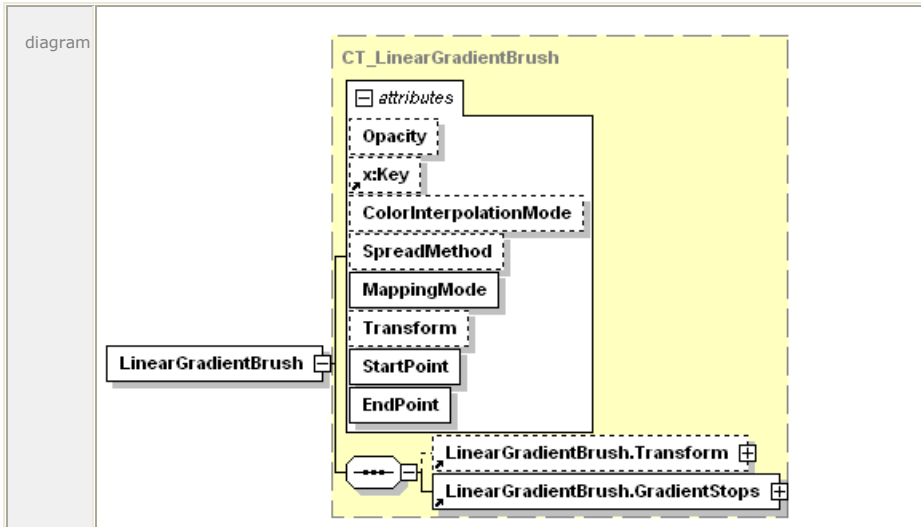
24

25 *end example]*



1 **13.5 <LinearGradientBrush> Element**

2 element **LinearGradientBrush**



Name	Type	Use	Default	Fixed	Annotation
Opacity	<a href="#">ST_ZeroOne</a>		1.0		Defines the uniform transparency of the linear gradient. Values range from 0 (fully transparent) to 1 (fully opaque), inclusive. Values outside of this range are invalid.
x:Key					Specifies a name for a resource in a resource dictionary. x:Key MUST be present when the current element is defined in a resource dictionary. x:Key MUST NOT be specified outside of a resource dictionary [M6.5].
ColorInterpolationMode	<a href="#">ST_ClrIntMode</a>		SRgbLinear Interpolation		Specifies the gamma function for color interpolation. The gamma adjustment

					should not be applied to the alpha component, if specified. Valid values are SRgbLinearInterpolation and ScRgbLinearInterpolation.
SpreadMethod	<a href="#">ST_SpreadMethod</a>		Pad		Describes how the brush should fill the content area outside of the primary, initial gradient area. Valid values are Pad, Reflect and Repeat.
MappingMode	<a href="#">ST_MappingMode</a>	required		Absolute	Specifies that the start point and end point are defined in the effective coordinate space (includes the Transform attribute of the brush).
Transform	<a href="#">ST_RscRefMatrix</a>				Describes the matrix transformation applied to the coordinate space of the brush. The Transform property on a brush is concatenated with the current effective render transform to yield an effective render transform local to the brush. The start point and end point are transformed using the local effective render transform.
StartPoint	<a href="#">ST_Point</a>	required			Specifies the starting point of the linear gradient.
EndPoint	<a href="#">ST_Point</a>	required			Specifies the end point of the linear gradient. The linear gradient brush interpolates the colors from the start point to the end point, where the start point represents an offset of

XPS Specification and Reference Guide 13.5 <LinearGradientBrush> Element

					<p>0, and the EndPoint represents an offset of 1. The Offset attribute value specified in a GradientStop element relates to the 0 and 1 offsets defined by the start point and end point.</p>
<p>annotation</p>	<p>Fills a region with a linear gradient.</p>				

1 The <LinearGradientBrush> element is used to specify a linear gradient brush along  
 2 a vector. For details about computing a linear gradient, see Section 18.3, "Gradient  
 3 Computations," on page 272.

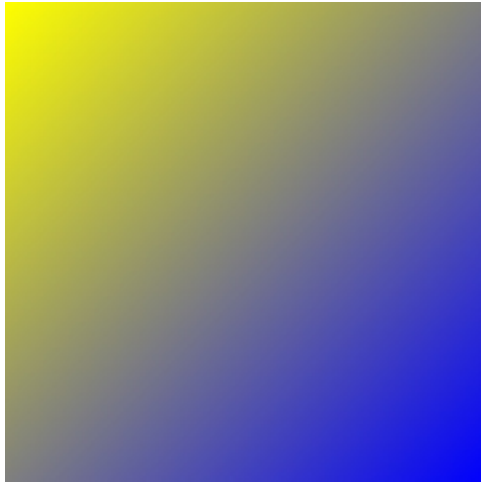
4 *Example 13-19. <LinearGradientBrush> usage*

5 The following markup describes a page with a rectangular path that is filled with a  
 6 linear gradient:

```

7 <Path>
8   <Path.Fill>
9     <LinearGradientBrush
10      MappingMode="Absolute"
11      StartPoint="0,0"
12      EndPoint="300,300">
13       <LinearGradientBrush.GradientStops>
14         <GradientStop Color="#FFFF00" Offset="0" />
15         <GradientStop Color="#0000FF" Offset="1" />
16       </LinearGradientBrush.GradientStops>
17     </LinearGradientBrush>
18   </Path.Fill>
19   <Path.Data>
20     <PathGeometry>
21       <PathFigure StartPoint="0,0">
22         <PolyLineSegment Points="300,0 300,300 0,300" />
23       </PathFigure>
24     </PathGeometry>
25   </Path.Data>
26 </Path>
  
```

1 This markup is rendered as follows:



2  
3 *end example]*

#### 4 **13.5.1 SpreadMethod Attribute**

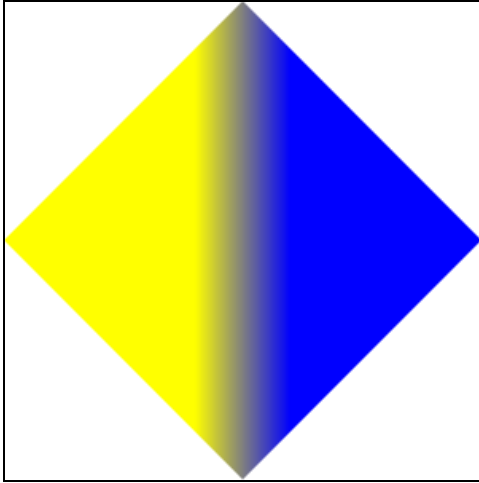
5 The **SpreadMethod** attribute describes the fill for areas beyond the start point and  
6 end point of the linear gradient brush. Valid values are Pad, Reflect, and Repeat.

7 *Example 13-20. Linear gradient brush with a SpreadMethod value of Pad*

8 In this method, the first color and the last color are used to fill the remaining fill area  
9 at the beginning and end.

```
10 <Path Data="M 150,0 L 300,150 L 150,300 L 0,150 z">  
11   <Path.Fill>  
12     <LinearGradientBrush  
13       MappingMode="Absolute"  
14       StartPoint="120,0"  
15       EndPoint="180,0"  
16       SpreadMethod="Pad">  
17       <LinearGradientBrush.GradientStops>  
18         <GradientStop Color="#FFFF00" Offset="0.0" />  
19         <GradientStop Color="#0000FF" Offset="1.0" />  
20       </LinearGradientBrush.GradientStops>  
21     </LinearGradientBrush>  
22   </Path.Fill>  
23 </Path>
```

1 This markup is rendered as follows:



2

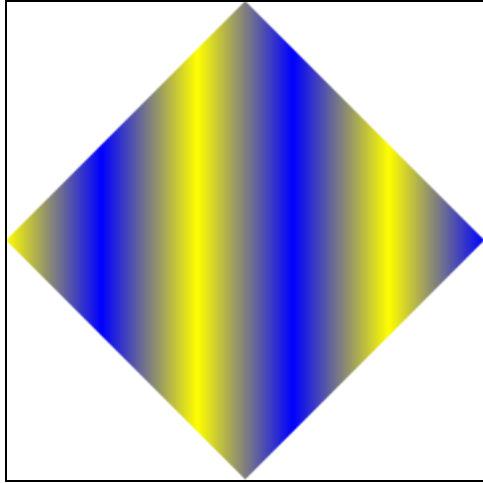
3 *end example]*

4 *Example 13-21. Linear gradient brush with a SpreadMethod value of Reflect*

5 In this method, the gradient stops are replayed in reverse order repeatedly to cover  
6 the fill area.

```
7 <Path Data="M 150,0 L 300,150 L 150,300 L 0,150 Z">  
8 <Path.Fill>  
9 <LinearGradientBrush  
10 MappingMode="Absolute"  
11 StartPoint="120,0"  
12 EndPoint="180,0"  
13 SpreadMethod="Reflect">  
14 <LinearGradientBrush.GradientStops>  
15 <GradientStop Color="#FFFF00" Offset="0.0" />  
16 <GradientStop Color="#0000FF" Offset="1.0" />  
17 </LinearGradientBrush.GradientStops>  
18 </LinearGradientBrush>  
19 </Path.Fill>  
20 </Path>
```

1 This markup is rendered as follows:



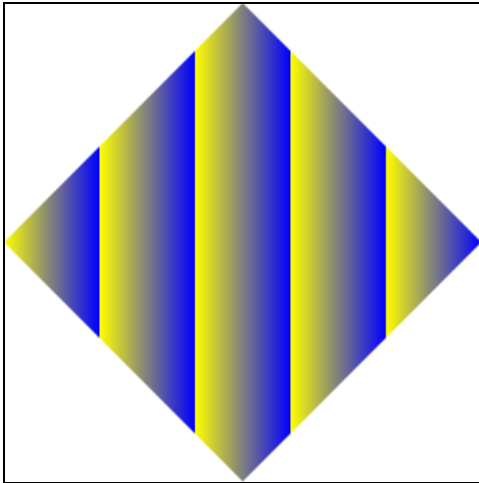
2  
3 *end example]*

4 *Example 13-22. Linear gradient brush with a SpreadMethod value of Repeat*

5 In this method, the gradient stops are repeated in order until the fill area is covered.

```
6 <Path Data="M 150,0 L 300,150 L 150,300 L 0,150 Z">  
7 <Path.Fill>  
8 <LinearGradientBrush  
9 MappingMode="Absolute"  
10 StartPoint="120,0"  
11 EndPoint="180,0"  
12 SpreadMethod="Repeat">  
13 <LinearGradientBrush.GradientStops>  
14 <GradientStop Color="#FFFF00" Offset="0.0" />  
15 <GradientStop Color="#0000FF" Offset="1.0" />  
16 </LinearGradientBrush.GradientStops>  
17 </LinearGradientBrush>  
18 </Path.Fill>  
19 </Path>
```

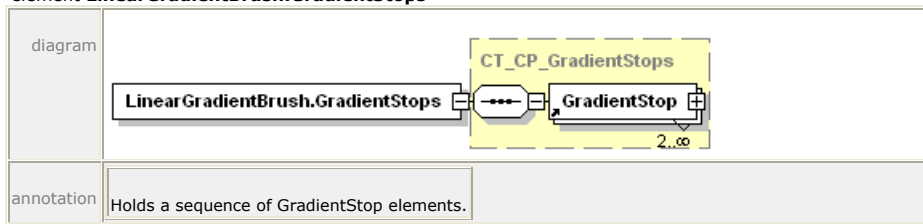
1 This markup is rendered as follows:



2  
3 *end example]*

4 **13.5.2 <LinearGradientBrush.GradientStops> Element**

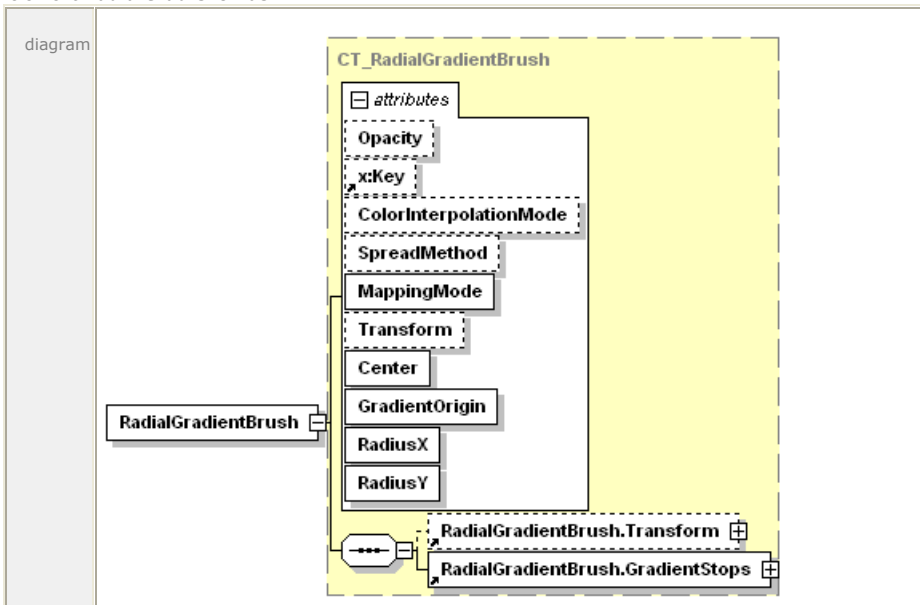
5 element **LinearGradientBrush.GradientStops**



6  
7 The <LinearGradientBrush.GradientStops> property element specifies a collection of  
8 gradient stops that comprise the linear gradient. For more information, see §13.7,  
9 “<GradientStop> Element,” on page 166.

1 **13.6 <RadialGradientBrush> Element**

2 element **RadialGradientBrush**



attributes	Name	Type	Use	Default	Fixed	Annotation
	Opacity	<a href="#">ST_ZeroOne</a>		1.0		Defines the uniform transparency of the radial gradient. Values range from 0 (fully transparent) to 1 (fully opaque), inclusive. Values outside of this range are invalid.
	x:Key					Specifies a name for a resource in a resource dictionary. x:Key MUST be present when the current element is defined in a resource dictionary. x:Key MUST NOT be specified outside of a resource dictionary [M6.6].
	ColorInterpolationMode	<a href="#">ST_ClrIntMode</a>		SRgbLinear		Specifies the gamma



XPS Specification and Reference Guide 13.6 <RadialGradientBrush> Element

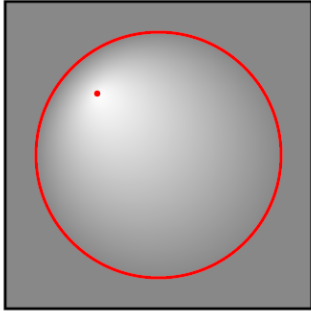
				Interpolation		function for color interpolation for sRGB colors. The gamma adjustment should not be applied to the alpha component, if specified. Valid values are SRgbLinearInterpolation and ScRgbLinearInterpolation.
SpreadMethod	<a href="#">ST_SpreadMethod</a>			Pad		Describes how the brush should fill the content area outside of the primary, initial gradient area. Valid values are Pad, Reflect and Repeat.
MappingMode	<a href="#">ST_MappingMode</a>	required			Absolute	Specifies that center, x radius, and y radius are defined in the effective coordinate space (includes the Transform attribute of the brush).
Transform	<a href="#">ST_RscRefMatrix</a>					Describes the matrix transformation applied to the coordinate space of the brush. The Transform property is concatenated with the current effective render transform to yield an effective render transform local to the brush. The ellipse defined by the center, gradient origin, x radius, and y radius values is transformed using the local effective render transform.
Center	<a href="#">ST_Point</a>	required				Specifies the center point of the radial gradient (that is, the center of the ellipse). The radial gradient brush interpolates the colors from the gradient origin to the circumference of the

						ellipse. The circumference is determined by the center and the radii.
	GradientOrigin	<u>ST_Point</u>	required			Specifies the origin point of the radial gradient.
	RadiusX	<u>ST_GZero</u>	required			Specifies the radius in the x dimension of the ellipse which defines the radial gradient.
	RadiusY	<u>ST_GZero</u>	required			Specifies the radius in the y dimension of the ellipse which defines the radial gradient.
annotation	Fills a region with a radial gradient.					

- 1 Radial gradient brushes are similar to linear gradient brushes. However, whereas a  
2 linear gradient brush has a start point and end point to define the gradient vector, a  
3 radial gradient brush has an ellipse (defined by the center, x radius, and y radius)  
4 and a gradient origin. The ellipse defines the end point of the gradient. In other  
5 words, a gradient stop with an offset at 1.0 defines the color at the circumference of  
6 the ellipse. The gradient origin defines the center of the gradient. A gradient stop  
7 with an offset at 0.0 defines the color at the gradient origin.
- 8 For details about computing a radial gradient, see §18.3.3, "Radial Gradients," on  
9 page 276.

1 *Example 13-23. A radial gradient brush*

2 The following figure is a radial gradient that transitions from white to gray. The  
 3 outside ellipse represents the gradient ellipse while the dot denotes the gradient  
 4 origin. This gradient has a **SpreadMethod** value of Pad. *end example]*



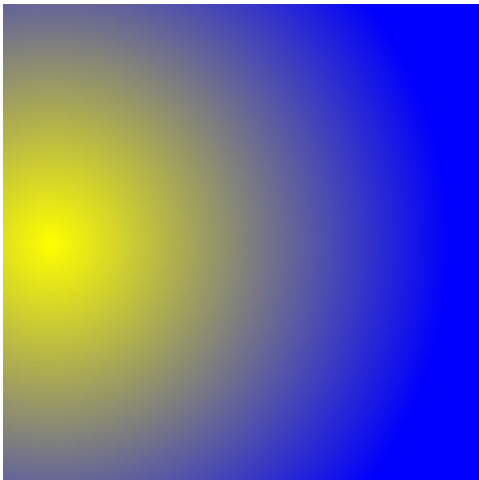
*Example 13-24. RadialGradientBrush usage*

The following markup describes a page with a rectangular path that is filled with a radial gradient:

```

20 <Path>
21   <Path.Fill>
22     <RadialGradientBrush
23       MappingMode="Absolute"
24       Center="30,150"
25       GradientOrigin="30,150"
26       RadiusX="250"
27       RadiusY="250">
28       <RadialGradientBrush.GradientStops>
29         <GradientStop Color="#FFFF00"
30           Offset="0" />
31         <GradientStop Color="#0000FF" Offset="1" />
32       </RadialGradientBrush.GradientStops>
33     </RadialGradientBrush>
34   </Path.Fill>
35   <Path.Data>
36     <PathGeometry>
37       <PathFigure StartPoint="0,0" IsClosed="true">
38         <PolyLineSegment Points="300,0 300,300 0,300" />
39       </PathFigure>
40     </PathGeometry>
41   </Path.Data>
42 </Path>
    
```

33 This markup is rendered as follows:



34

1 *end example]*

### 2 **13.6.1 SpreadMethod Attribute**

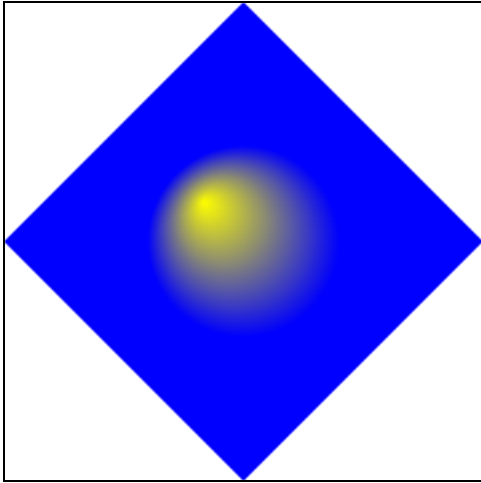
3 The **SpreadMethod** attribute describes the fill of areas beyond the ellipse described  
4 by the center, x radius, and y radius of the radial gradient brush. Valid values are  
5 Pad, Reflect, and Repeat.

6 *Example 13-25. Radial gradient brush with a SpreadMethod value of Pad*

7 In the following markup, the last color is used to cover the fill area outside the  
8 ellipse.

```
9 <Path Data="M 150,0 L 300,150 L 150,300 L 0,150 Z">  
10 <Path.Fill>  
11 <RadialGradientBrush  
12 MappingMode="Absolute"  
13 Center="150,150"  
14 GradientOrigin="125,125"  
15 RadiusX="60"  
16 RadiusY="60"  
17 SpreadMethod="Pad">  
18 <RadialGradientBrush.GradientStops>  
19 <GradientStop Color="#FFFF00" Offset="0.0" />  
20 <GradientStop Color="#0000FF" Offset="1.0" />  
21 </RadialGradientBrush.GradientStops>  
22 </RadialGradientBrush>  
23 </Path.Fill>  
24 </Path>
```

1 This markup is rendered as follows:



2

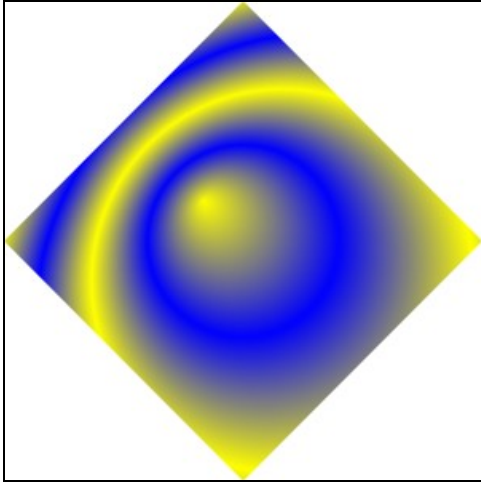
3 *end example]*

4 *Example 13-26. Radial gradient brush with a SpreadMethod value of Reflect*

5 In the following markup, the gradient stops are replayed in reverse order repeatedly  
6 to cover the fill area.

```
7 <Path Data="M 150,0 L 300,150 L 150,300 L 0,150 Z">  
8 <Path.Fill>  
9 <RadialGradientBrush  
10 MappingMode="Absolute"  
11 Center="150,150"  
12 GradientOrigin="125,125"  
13 RadiusX="60"  
14 RadiusY="60"  
15 SpreadMethod="Reflect">  
16 <RadialGradientBrush.GradientStops>  
17 <GradientStop Color="#FFFF00" Offset="0.0" />  
18 <GradientStop Color="#0000FF" Offset="1.0" />  
19 </RadialGradientBrush.GradientStops>  
20 </RadialGradientBrush>  
21 </Path.Fill>  
22 </Path>
```

1 This markup is rendered as follows:



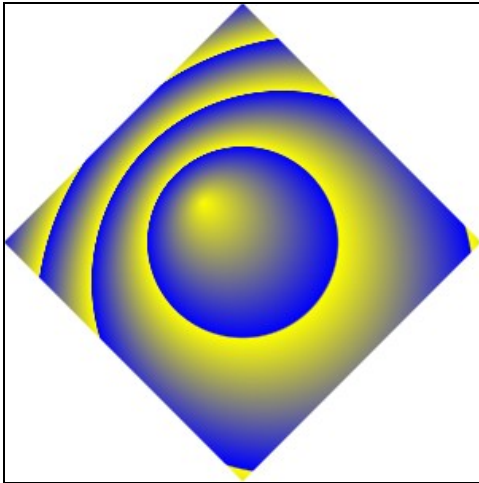
2  
3 *end example]*

4 *Example 13–27. Radial gradient brush with a SpreadMethod value of Repeat*

5 In the following markup, the gradient stops are repeated in order until the fill area is  
6 covered.

```
7 <Path Data="M 150,0 L 300,150 L 150,300 L 0,150 Z">  
8 <Path.Fill>  
9 <RadialGradientBrush  
10 MappingMode="Absolute"  
11 Center="150,150"  
12 GradientOrigin="125,125"  
13 RadiusX="60"  
14 RadiusY="60"  
15 SpreadMethod="Repeat">  
16 <RadialGradientBrush.GradientStops>  
17 <GradientStop Color="#FFFF00" Offset="0.0" />  
18 <GradientStop Color="#0000FF" Offset="1.0" />  
19 </RadialGradientBrush.GradientStops>  
20 </RadialGradientBrush>  
21 </Path.Fill>  
22 </Path>
```

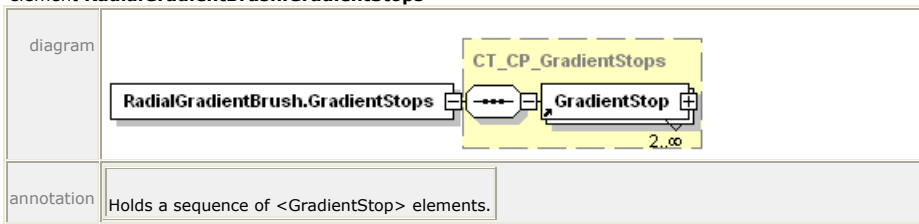
1 This markup is rendered as follows:



2  
3 *end example]*

4 **13.6.2 <RadialGradientBrush.GradientStops> Element**

5 element **RadialGradientBrush.GradientStops**



6 The <RadialGradientBrush.GradientStops> property element specifies a collection of  
7 gradient stops that comprise the radial gradient. For more information, see §13.7,  
8 “<GradientStop> Element,” on page 166.

## 1 13.7 <GradientStop> Element

2 element **GradientStop**

diagram						
attributes	Name	Type	Use	Default	Fixed	Annotation
	Color	<a href="#">ST_Color</a>	required			Specifies the gradient stop color. An sRGB color value specified as a 6-digit hexadecimal number (#RRGGBB) or an extended color.
	Offset	<a href="#">ST_Double</a>	required			Specifies the gradient offset. The offset indicates a point along the progression of the gradient at which a color is specified. Colors between gradient offsets in the progression are interpolated.
annotation	Indicates a location and range of color progression for rendering a gradient.					

3 The <GradientStop> element is used by both the <LinearGradientBrush> and  
 4 <RadialGradientBrush> elements to define the location and range of color  
 5 progression for rendering a gradient.

6 For linear gradient brushes, the offset value of 0.0 is mapped to the start point of the  
 7 gradient, and the offset value of 1.0 is mapped to the end point. Intermediate offset  
 8 values are interpolated between these two points to determine their location.

9 For radial gradient brushes, the offset value of 0.0 is mapped to the gradient origin  
 10 location. The offset value of 1.0 is mapped to the circumference of the ellipse as  
 11 determined by the center, x radius, and y radius. Offsets between 0.0 and 1.0 are  
 12 positioned at a location interpolated between these points.

13 For full details of rendering of gradient brushes, including handling of offsets, please  
 14 see §18.3, "Gradient Computations" on page 272.



## 13.8 Using a Brush as an Opacity Mask

Each pixel carries an alpha value ranging from 0.0 (fully transparent) to 1.0 (fully opaque). The alpha value is used when blending elements to achieve the visual effect of transparency. Each element can have an **Opacity** attribute by which the alpha value of each pixel is multiplied uniformly.

The OpacityMask property also allows the specification of per-pixel opacity, which controls how rendered content is blended with its destination. The opacity specified by the opacity mask is combined multiplicatively with any opacity that may already be present in the alpha channel of the contents. The per-pixel opacity specified by the opacity mask is determined by the alpha channel of each pixel in the mask. The color data is ignored.

The alpha value of the area not marked by the brush is 0.0. The required computations for transparently blending two elements when rendering, also known as [alpha blending](#), are described in §18.4, "Opacity Computations" on page 279.

An opacity mask always has a brush child element.

*Example 13-28. Opacity mask with linear gradient*

The following markup illustrates how an opacity mask is used to create a fade effect on a glyph. The opacity mask is a linear gradient that fades from opaque black to transparent black.

```
<FixedPage Height="1056" Width="816" xml:lang="en-US">
  <Glyphs
    OriginX="25"
    OriginY="50"
    UnicodeString="This is a fading text example."
    FontUri="./Resources/Fonts/Times.TTF"
    FontRenderingEmSize="32">
    <Glyphs.OpacityMask>
      <LinearGradientBrush
        StartPoint="25,0"
        EndPoint="450,0"
        MappingMode="Absolute">
        <LinearGradientBrush.GradientStops>
          <GradientStop Color="#FF000000" Offset="0" />
          <GradientStop Color="#00000000" Offset="1" />
        </LinearGradientBrush.GradientStops>
      </LinearGradientBrush>
    </Glyphs.OpacityMask>
    <Glyphs.Fill>
      <SolidColorBrush Color="#000000" />
    </Glyphs.Fill>
  </Glyphs>
</FixedPage>
```

This markup is rendered as follows:



1 *end example]*

1 *Example 13–29. Opacity mask with radial gradient*

2 In the following markup, the opacity mask is a radial gradient:

```

3 <FixedPage Width="816" Height="1056" xml:lang="en-US">
4 <Path>
5 <Path.OpacityMask>
6 <RadialGradientBrush
7 MappingMode="Absolute"
8 Center="200,300"
9 GradientOrigin="200,300"
10 RadiusX="200"
11 RadiusY="300">
12 <RadialGradientBrush.GradientStops>
13 <GradientStop Color="#FF000000" Offset="0" />
14 <GradientStop Color="#20000000" Offset="1" />
15 </RadialGradientBrush.GradientStops>
16 </RadialGradientBrush>
17 </Path.OpacityMask>
18 <Path.Fill>
19 <ImageBrush
20 Viewbox="0,0,400,600"
21 ViewboxUnits="Absolute"
22 Viewport="0,0,400,600"
23 ViewportUnits="Absolute"
24 TileMode="None"
25 ImageSource="images/jpeg3.jpg" />
26 </Path.Fill>
27 <Path.Data>
28 <PathGeometry>
29 <PathFigure StartPoint="0,0" IsClosed="true">
30 <PolyLineSegment Points="400,0 400,600 0,600" />
31 </PathFigure>
32 </PathGeometry>
33 </Path.Data>
34 </Path>
35 </FixedPage>

```

- 1 This markup is rendered as follows:



- 2
- 3 *end example]*

## 1 14. Common Properties

2 Several XPS Document elements share property attributes and elements as  
 3 summarized in Table 14–1 and Table 14–2 and detailed in the following sections.  
 4 Other than the Name, FixedPage.NavigateUri, and xml:lang attributes, these  
 5 properties compose their results from parent to child, as described in §18.4.1, “Pre-  
 6 Multiplied Alpha and Superluminous Colors,” on page 281.

7 *Table 14–1. Common property attributes*

Name	Applies to	Description
Clip	<Canvas> <Glyphs> <Path>	Restricts the region to which a brush can be applied.
Opacity	<Canvas> <Glyphs> <ImageBrush> <LinearGradientBrush> <Path> <RadialGradientBrush> <SolidColorBrush> <VisualBrush>	Defines the uniform transparency of the element.
OpacityMask	<Canvas> <Glyphs> <Path>	Specifies a mask of alpha values.
RenderTransform	<Canvas> <Glyphs> <Path>	Establishes a new coordinate space through the use of an affine matrix transformation. For more information, see §14.4, “Positioning Content,” on page 185.
Transform	<ImageBrush> <LinearGradientBrush> <PathGeometry> <RadialGradientBrush> <VisualBrush>	Establishes a new coordinate space through the use of an affine matrix transformation. Geometry transformations are applied before brushes. The results are concatenated with any containing effective render transformation specification.
Name	<Canvas> <FixedPage> <Glyphs> <Path>	Defines a hyperlink target or identifies an element uniquely for document structure markup to reference. For more information, see §16.2, “Hyperlinks,” on page 243.

FixedPage.NavigateUri	<Canvas> <Glyphs> <Path>	Defines a hyperlink source. For more information, see §16.2, "Hyperlinks," on page 243.
xml:lang	<Canvas> <FixedPage> <Glyphs> <Path>	Specifies a language.

1 Table 14-2. Common property elements

Name	Description
<Canvas.Resources> <FixedPage.Resources>	Contains elements that may be reused by reference throughout the markup of the <FixedPage> or <Canvas> child or descendant elements.
<Canvas.Clip> <Glyphs.Clip> <Path.Clip>	Restricts the region to which a brush can be applied.
<Canvas.RenderTransform> <Glyphs.RenderTransform> <Path.RenderTransform>	Establishes a new coordinate space through the use of an affine matrix transformation. For more information, see §14.4, "Positioning Content," on page 185.
<ImageBrush.Transform> <LinearGradientBrush.Transform> <PathGeometry.Transform> <RadialGradientBrush.Transform> <VisualBrush.Transform>	Establishes a new effective coordinate space through the use of an affine matrix transformation. Path geometry transformations (<PathGeometry.Transform>) are applied before brushes. The results are concatenated with any containing effective render transformation.
<Canvas.OpacityMask> <Glyphs.OpacityMask> <Path.OpacityMask>	Specifies a mask of alpha values that is applied in the same fashion as the Opacity attribute, but allows different alpha values on a pixel-by-pixel basis.

2 **14.1 Opacity**

3 The **Opacity** property attribute is used to transparently blend the current element  
 4 with previously specified elements, also known as alpha blending. The opacity value  
 5 MUST fall within the 0 (fully transparent) to 1 (fully opaque) range, inclusive  
 6 [M2.72].

7 For more information, see §18.4, "Opacity Computations," on page 279.

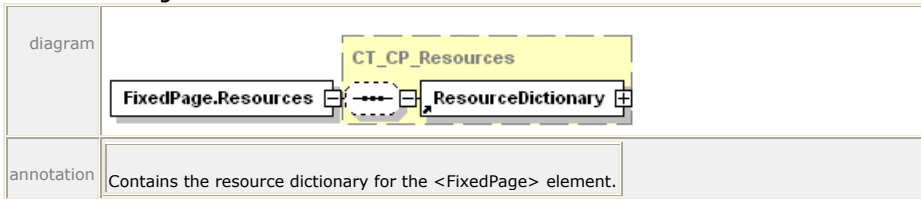
8 **14.2 Resources and Resource References**

9 Fixed page markup supports the concept of resources. A **resource** is a reusable  
 10 property value that is expressed in markup, identified by a key, and stored in a

- 1 **resource dictionary**. In general, any property value that can be expressed using  
 2 property element syntax may be held in a resource dictionary.
- 3 Each resource in a resource dictionary has a key. Any property that specifies its  
 4 value by referencing a resource key in a resource dictionary is called a **resource**  
 5 **reference**.
- 6 The <Canvas> and <FixedPage> elements can carry a resource dictionary. A  
 7 resource dictionary is expressed in markup by the <FixedPage.Resources> or  
 8 <Canvas.Resources> property element. Individual resource values MUST be  
 9 specified within a resource dictionary [M7.1].
- 10 The <Canvas.Resources> or <FixedPage.Resources> property elements MUST  
 11 precede any property elements of the <Canvas> or <FixedPage> elements [M2.72].  
 12 Likewise, they MUST precede any path, glyphs, or canvas children of the <Canvas>  
 13 or <FixedPage> elements [M2.72].
- 14 Alternatively, resource dictionaries MAY be specified in separate parts and referenced  
 15 from within the <FixedPage.Resources> or <Canvas.Resources> property element  
 16 [O7.1]. Such a **remote resource dictionary** can be shared across multiple pages.  
 17 [Example: By defining a brush in a remote resource dictionary, graphical elements  
 18 that are common to multiple pages can be reused. end example]
- 19 The <Path>, <Glyphs>, and <Canvas> elements can appear as a resource definition  
 20 solely for the purpose of using these elements in the **Visual** attribute of a  
 21 <VisualBrush> element. Brushes and geometries appear in resource dictionaries far  
 22 more frequently.

23 **14.2.1 <FixedPage.Resources> Element**

24 element **FixedPage.Resources**



25 *Example 14-1. <FixedPage.Resources> usage*

```

26 <FixedPage Width="816" Height="1056" xml:lang="en-US"
27 xmlns="http://schemas.microsoft.com/xps/2005/06"
28 xmlns:x="http://schemas.microsoft.com/xps/2005/06/resourcedictionary-
29 key">
30 <FixedPage.Resources>
31 <ResourceDictionary>
32 <PathGeometry x:Key="Rectangle">
33 <PathFigure StartPoint="20,20" IsClosed="true">
34 <PolyLineSegment Points="120,20 120,70 20,70" />
35 </PathFigure>
36 </PathGeometry>
37 </ResourceDictionary>
38 </FixedPage.Resources>
39 <Path Stroke="#000000"
40 StrokeThickness="1"
    
```

```
1     Data="{StaticResource Rectangle}" />  
2 </FixedPage>
```

3 This markup is rendered as follows:

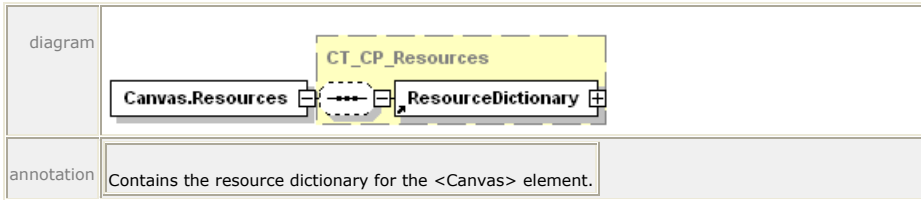


4  
5 *end example]*



1 **14.2.2 <Canvas.Resources> Element**

2 element **Canvas.Resources**



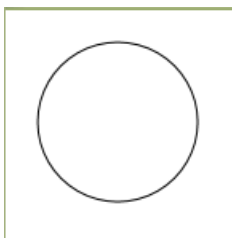
4 *Example 14-2. <Canvas.Resources> usage*

```

5 <Canvas
6   xmlns:x="http://schemas.microsoft.com/xps/2005/06/resourcedictionary-
7   key">
8   <Canvas.Resources>
9     <ResourceDictionary>
10      <PathGeometry x:Key="Circle">
11        <PathFigure StartPoint="20,70">
12          <ArcSegment
13            Point="120,70"
14            Size="50,50"
15            RotationAngle="0"
16            IsLargeArc="true"
17            SweepDirection="Clockwise" />
18          <ArcSegment
19            Point="20,70"
20            Size="50,50"
21            RotationAngle="0"
22            IsLargeArc="true"
23            SweepDirection="Clockwise" />
24        </PathFigure>
25      </PathGeometry>
26    </ResourceDictionary>
27  </Canvas.Resources>
28  <Path Stroke="#000000"
29        StrokeThickness="1"
30        Data="{StaticResource Circle}" />
31 </Canvas>

```

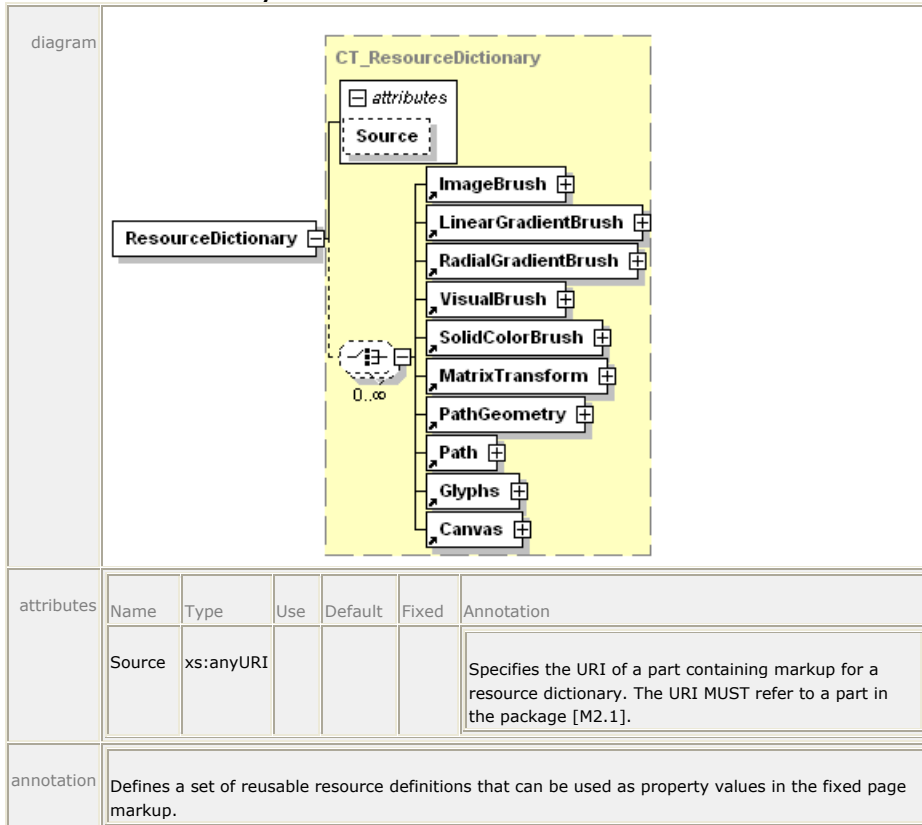
32 This markup is rendered as follows:



33  
34 *end example]*

1 **14.2.3 <ResourceDictionary> Element**

2 element **ResourceDictionary**



3 The <FixedPage.Resources> and <Canvas.Resources> property elements contain  
 4 exactly one <ResourceDictionary> element. A resource dictionary contains **resource**  
 5 **definition** element entries. Each resource definition has a key specified in the **x:Key**  
 6 attribute that is unique within the scope of the resource dictionary. The **x:Key**  
 7 attribute is included in the Resource Dictionary namespace specified in H, "Standard  
 8 Namespaces and Content Types" on page 405.

9 Resource dictionaries can be declared inline inside a <FixedPage.Resources> or  
 10 <Canvas.Resources> element, or they MAY be defined in a separate part and  
 11 referenced by a <ResourceDictionary> element inside a <FixedPage.Resources> or  
 12 <Canvas.Resources> element [O7.1]. This allows resource dictionaries to be shared  
 13 across parts. [Example: A single resource dictionary can be used by every fixed page  
 14 in the XPS Document. end example] See §14.2.3.1, "Remote Resource Dictionaries,"  
 15 on page 177 for more details.

16 A resource definition MAY reference another resource defined previously in the same  
 17 resource dictionary [O7.2]. If the resource dictionary does not appear in a separate

1 part, a resource definition MAY reference a previously defined resource in a resource  
2 dictionary of a parent or ancestor <Canvas> or <FixedPage> element [O7.3].

3 Namespace prefixes in resource definitions MUST apply in the context of the  
4 definition, rather than in the context of the resource reference [M7.2]. An **xml:lang**  
5 attribute within a resource definition MUST be interpreted in the context of the  
6 resource reference, not the resource definition [M7.3].

7 *Example 14-3. Resource dictionary markup*

8 The following markup defines two geometries, one for a rectangle, and the other for  
9 a circle:

```
10 <Canvas
11   xmlns:x="http://schemas.microsoft.com/xps/2005/06/resourcedictionary-
12   key">
13   <Canvas.Resources>
14     <ResourceDictionary>
15       <PathGeometry x:Key="Rectangle">
16         <PathFigure StartPoint="20,20" IsClosed="true">
17           <PolyLineSegment Points="120,20 120,70 20,70" />
18         </PathFigure>
19       </PathGeometry>
20       <PathGeometry x:Key="Circle">
21         <PathFigure StartPoint="20,70">
22           <ArcSegment
23             Point="120,70"
24             Size="50,50"
25             RotationAngle="0"
26             IsLargeArc="true"
27             SweepDirection="Clockwise" />
28           <ArcSegment
29             Point="20,70"
30             Size="50,50"
31             RotationAngle="0"
32             IsLargeArc="true"
33             SweepDirection="Clockwise" />
34         </PathFigure>
35       </PathGeometry>
36     </ResourceDictionary>
37   </Canvas.Resources>
38   <Path Data="{StaticResource Rectangle}">
39     <Path.Fill>
40       <SolidColorBrush Color="#FF0000" />
41     </Path.Fill>
42   </Path>
43 </Canvas>
```

44 *end example]*

#### 45 14.2.3.1 Remote Resource Dictionaries

46 A resource dictionary MAY be defined in a separate part [O7.1]. This is referred to as  
47 a **remote resource dictionary**. A remote resource dictionary MUST follow the  
48 requirements above that apply to all resource dictionaries [M7.4]. A remote resource  
49 dictionary MUST NOT contain any resource definition children that reference another  
50 remote resource dictionary [M7.5].

1 The <FixedPage.Resources> and <Canvas.Resources> property elements include a  
 2 remote resource dictionary via reference, using the **Source** attribute of the  
 3 <ResourceDictionary> element.

4 A <ResourceDictionary> element that specifies a remote resource dictionary in its  
 5 **Source** attribute MUST NOT contain any resource definition children [M7.6].  
 6 <FixedPage.Resources> and <Canvas.Resources> elements that include a remote  
 7 resource dictionary MUST include exactly one <ResourceDictionary> element  
 8 [M2.72].

9 A remote Resource Dictionary part MUST be added as a Required Resource  
 10 relationship from the FixedPage part that references it [M2.10]. In addition,  
 11 producers MUST add each resource such as fonts or images referenced in the  
 12 Resource Dictionary part as a Required Resource relationship from the FixedPage  
 13 part (*not* the Resource Dictionary part) to the indirectly required resource, even if  
 14 the particular fixed page does not reference the resource [M2.10]. For more  
 15 information, see §H.3, "Relationship Types," on page 407.

16 Inline references to fonts or images in remote resource dictionary entries MUST be  
 17 interpreted with the same base URI as the Remote Resource Dictionary part, not  
 18 from the base URI of the part referring to the particular remote resource dictionary  
 19 entry [M7.7].

20 *Example 14-4. A remote resource dictionary and reference*

21 The following markup defines a resource dictionary that contains two geometries,  
 22 one for a rectangle and the other for a circle:

```

23 <!-- Contents of /resource.xaml -->
24 <ResourceDictionary xmlns="http://schemas.microsoft.com/xps/2005/06"
25   xmlns:x="http://schemas.microsoft.com/xps/2005/06/resourcedictionary-
26   key">
27   <PathGeometry x:Key="Rectangle">
28     <PathFigure StartPoint="20,20" IsClosed="true">
29       <PolyLineSegment Points="120,20 120,70 20,70" />
30     </PathFigure>
31   </PathGeometry>
32   <PathGeometry x:Key="Circle">
33     <PathFigure StartPoint="20,70">
34       <ArcSegment
35         Point="120,70"
36         Size="50,50"
37         RotationAngle="0"
38         IsLargeArc="true"
39         SweepDirection="Clockwise" />
40       <ArcSegment
41         Point="20,70"
42         Size="50,50"
43         RotationAngle="0"
44         IsLargeArc="true"
45         SweepDirection="Clockwise" />
46     </PathFigure>
47   </PathGeometry>
48 </ResourceDictionary>

```

49 The following markup references the previously defined resource dictionary:

```

50 <Canvas
51   xmlns:x="http://schemas.microsoft.com/xps/2005/06/resourcedictionary-
52   key">

```

*XPS Specification and Reference Guide* 14.2 Resources and Resource References

```
1 <Canvas.Resources>
2   <ResourceDictionary Source="/resource.xaml"/>
3 </Canvas.Resources>
4 <Path Data="{StaticResource Rectangle}">
5   <Path.Fill>
6     <SolidColorBrush Color="#FF0000" />
7   </Path.Fill>
8 </Path>
9 </Canvas>
```

10 *end example]*

### 1 14.2.4 Resource References

2 To set a property value to a defined resource, use the form:

```
3 {StaticResource key}
```

4 Where *key* is the same string specified with **x:Key** in the resource definition.

5 The context of the resource reference determines how defined resources are  
6 rendered (such as the transformation matrix to be applied). Specifically, the effective  
7 coordinate space for rendering the referenced resource is a composition of the  
8 effective coordinate space of the referring element plus any Transform or  
9 RenderTransform properties included in the resource definition itself.

10 It is considered an error if a static resource reference cannot be resolved, or if it *can*  
11 be resolved but the resource type does not match the usage at the location of  
12 reference.

13 *Example 14-5. Using a resource reference to fill a brush*

14 In the following markup, the rectangular region defined by the geometry specified in  
15 the dictionary is filled by a solid color brush:

```
16 <Canvas
17     xmlns:x="http://schemas.microsoft.com/xps/2005/06/resourcedictionary-
18     key">
19     <Canvas.Resources>
20         <ResourceDictionary>
21             <PathGeometry x:Key="Rectangle">
22                 <PathFigure StartPoint="20,20" IsClosed="true">
23                     <PolyLineSegment Points="120,20 120,70 20,70" />
24                 </PathFigure>
25             </PathGeometry>
26         </ResourceDictionary>
27     </Canvas.Resources>
28     <Path Data="{StaticResource Rectangle}">
29         <Path.Fill>
30             <SolidColorBrush Color="#FF0000" />
31         </Path.Fill>
32     </Path>
33 </Canvas>
```

34 *end example]*

### 35 14.2.5 Scoping Rules for Resolving Resource References

36 The value of the **x:Key** attribute MUST be unique within the resource dictionary  
37 [M2.72]. However, the resource dictionary of a <Canvas> element MAY re-use an  
38 **x:Key** value defined in the resource dictionary of a parent or ancestor <Canvas> or  
39 <FixedPage> element [O7.5]. Resource references are resolved from the innermost  
40 to the outermost resource dictionary.

41 A resource definition MAY reference a previously defined resource with the same  
42 name that is defined in an ancestor resource dictionary [O7.6]; the reference MUST  
43 be resolved before the redefined resource is added to the dictionary [M7.8].

44 A resource definition MAY reference another resource defined prior to the point of  
45 reference, including a resource previously defined within the same resource

1 dictionary [O7.2]. If a resource definition references another resource, the reference  
 2 MUST be resolved in the context of the resource definition, not in the context of the  
 3 resource use [M7.9].

4 To find a resource, the nearest parent or ancestor canvas or fixed page is searched.  
 5 If the desired name is not defined in the initially searched resource dictionary, then  
 6 the next-nearest parent or ancestor canvas or fixed page is searched. An error  
 7 occurs if the search has continued to the root <FixedPage> element and a specified  
 8 resource has not been found. This search occurs only within the containing FixedPage  
 9 part.

10 *Example 14-6. Using scoping rules*

```

11 <FixedPage
12   xmlns="http://schemas.microsoft.com/xps/2005/06"
13   xmlns:x="http://schemas.microsoft.com/xps/2005/06/resourcedictionary-
14     key"
15   Height="1056" Width="816" xml:lang="en-US">
16   <FixedPage.Resources>
17     <ResourceDictionary>
18       <SolidColorBrush x:Key="FavoriteColorFill" Color="#808080" />
19     </ResourceDictionary>
20 </FixedPage.Resources>
21 <Canvas>
22   <Canvas.Resources>
23     <ResourceDictionary>
24       <SolidColorBrush x:Key="FavoriteColorFill"
25         Color="#000000" />
26     </ResourceDictionary>
27   </Canvas.Resources>
28   <!-- The following path is filed with color #000000 -->
29   <Path Fill="{StaticResource FavoriteColorFill}">
30     <Path.Data>
31       ...
32     </Path.Data>
33   </Path>
34 </Canvas>
35   <!-- The following path is filed with color #000000 -->
36   <Path Fill="{StaticResource FavoriteColorFill}">
37     <Path.Data>
38       ...
39     </Path.Data>
40   </Path>
41 </Canvas>
42 </Canvas>
43   <!-- The following path is filled with color #808080 -->
44   <Path Fill="{StaticResource FavoriteColorFill}">
45     <Path.Data>
46       ...
47     </Path.Data>
48   </Path>
49 </FixedPage>

```

50 *end example]*

### 51 **14.2.6 Support for Markup Compatibility**

52 If a resource dictionary contains Markup Compatibility elements and attributes, the  
 53 processing of the Markup Compatibility markup MUST occur in the context of the  
 54 definition of the resource dictionary, not in the context of resource references  
 55 [M2.10].

## 1 14.3 Clipping

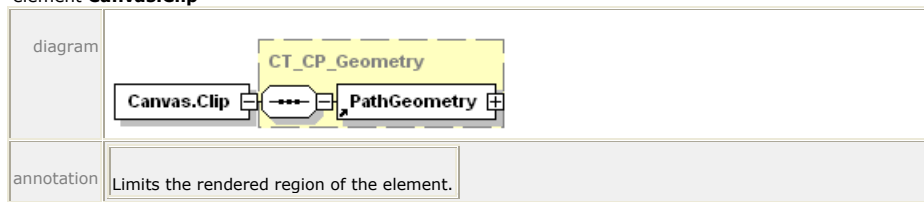
2 The Clip property specifies a geometric area that restricts the area to be filled with a  
3 brush.

4 The geometry is specified by a child <PathGeometry> element as detailed in §11.2,  
5 "Geometries and Figures," on page 77, or by abbreviated geometry syntax,  
6 described in §11.2.3, "Abbreviated Geometry Syntax," on page 90.

7 The default fill rule for geometries that do not specify a value is EvenOdd.

### 8 14.3.1 <Canvas.Clip> Element

9 element **Canvas.Clip**



10 The <Canvas.Clip> property element applies to all child and descendant elements of  
11 the canvas.

12 *Example 14-7. Canvas clip markup and rendering*

```

13 <Canvas>
14   <Canvas.Clip>
15     <PathGeometry>
16       <PathFigure StartPoint="25,25" IsClosed="true">
17         <PolyLineSegment Points="60,25 70,60 80,25 115,25
18           115,115 80,115 70,80 60,115 25,115" />
19       </PathFigure>
20     </PathGeometry>
21   </Canvas.Clip>
22   <Path Fill="#9999CC">
23     <Path.Data>
24       <PathGeometry>
25         <PathFigure StartPoint="20,70">
26           <ArcSegment
27             Point="120,70"
28             Size="50,50"
29             RotationAngle="0"
30             IsLargeArc="true"
31             SweepDirection="Clockwise" />
32           <ArcSegment
33             Point="20,70"
34             Size="50,50"
35             RotationAngle="0"
36             IsLargeArc="true"
37             SweepDirection="Clockwise" />
38         </PathFigure>
39       </PathGeometry>
40     </Path.Data>
41   </Path>
42 </Canvas>

```



1 This markup is rendered as follows:

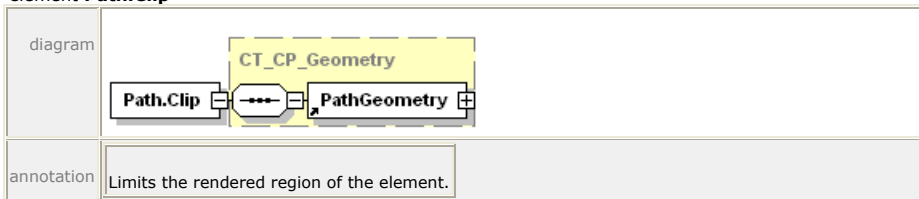


2

3 *end example]*

#### 4 **14.3.2 <Path.Clip> Element**

5 element **Path.Clip**



6 A clipping region may also be applied to a specific path.

7 *Example 14-8. <Path.Clip> usage*

8 The following markup describes a complex clipping behavior:

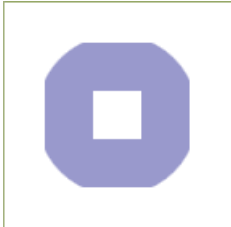
```

9 <Path Fill="#9999CC">
10 <Path.Clip>
11 <PathGeometry>
12 <PathFigure StartPoint="25,25" IsClosed="true">
13 <PolyLineSegment Points="115,25 115,115 25,115" />
14 </PathFigure>
15 <PathFigure StartPoint="55,55" IsClosed="true">
16 <PolyLineSegment Points="85,55 85,85 55,85" />
17 </PathFigure>
18 </PathGeometry>
19 </Path.Clip>
20 <Path.Data>
21 <PathGeometry>
22 <PathFigure StartPoint="20,70">
23 <ArcSegment
24 Point="120,70"
25 Size="50,50"
26 RotationAngle="0"
27 IsLargeArc="true"
28 SweepDirection="Clockwise" />
29 <ArcSegment
30 Point="20,70"
31 Size="50,50"
32 RotationAngle="0"
33 IsLargeArc="true"
34 SweepDirection="Clockwise" />
35 </PathFigure>
36 </PathGeometry>
37 </Path.Data>

```

1 `</Path>`

2 This markup is rendered as follows:

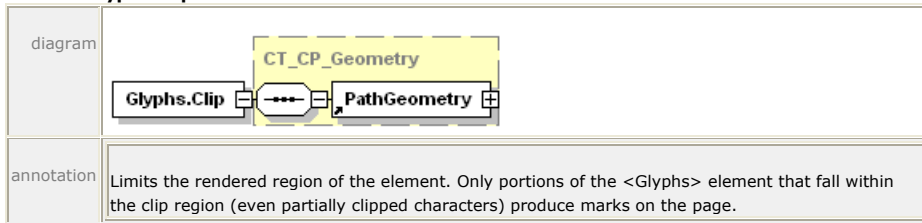


3

4 *end example]*

### 5 14.3.3 <Glyphs.Clip> Element

6 element **Glyphs.Clip**



7 *Example 14–9. <Glyphs.Clip> usage*

8 The following markup uses abbreviated geometry syntax to define the clipping  
9 region:

```
10 <Glyphs
11   Fill="#000000"
12   Clip="M 0,0 L 180,0 L 180,140 L 0,140 Z M 20,60 L 140,60 L 140,80
13     L 20,80 Z"
14   OriginX="20"
15   OriginY="130"
16   UnicodeString="N"
17   FontRenderingEmSize="170"
18   FontUri="../Resources/Fonts/Timesbd.ttf" />
```

19 This markup is rendered as follows:



20

21 *end example]*

## 14.4 Positioning Content

Content is positioned according to the properties specified for the fixed page or canvas, the properties specified for elements within the fixed page or canvas, and the compositional rules defined for the fixed payload namespace.

Elements are positioned relative to the current origin (0,0) of the coordinate space. The current origin can be moved by setting the `RenderTransform` property of a canvas, path, or glyph. The render transformation establishes a new coordinate frame for all children of the parent element.

Geometries and brushes may be manipulated in a similar way by setting the `Transform` property. The transform results are concatenated with the current render transformation to create an effective render transformation for the local element.

The `RenderTransform` and `Transform` properties both specify an affine matrix transformation to the local coordinate space, using the `<MatrixTransform>` element as their value. An abbreviated matrix transformation syntax MAY be used to specify a **`RenderTransform`** or **`Transform`** attribute value [M2.72].

### 14.4.1 `<MatrixTransform>` Element

element `MatrixTransform`

diagram						
attributes	Name	Type	Use	Default	Fixed	Annotation
	Matrix	<a href="#">ST_Matrix</a>	required			Specifies the matrix structure that defines the transformation.
	x:Key					Specifies a name for a resource in a resource dictionary. x:Key MUST be present when the current element is defined in a resource dictionary. x:Key MUST NOT be specified outside of a resource dictionary [M7.11].
annotation	Creates an arbitrary affine matrix transformation that manipulates objects or coordinate systems in a two-dimensional plane.					

The `<MatrixTransform>` element defines an arbitrary affine matrix transformation used to manipulate the coordinate systems of elements. A 3x3 matrix is used for transformations in an x,y plane. Affine transformation matrices can be multiplied to form any number of linear transformations, such as rotation and skew (shear), followed by translation. An affine transformation matrix has its final column equal to 0,0,1, so only the members in the first two columns are specified.

1

$$\begin{bmatrix} M11 & M12 & 0 \\ M21 & M22 & 0 \\ \text{OffsetX} & \text{OffsetY} & 1 \end{bmatrix}$$

2 This structure is specified by the **Matrix** attribute of the <MatrixTransform> element  
 3 as the six numbers in the first two columns. [*Example:*

4 "M11,M12,M21,M22,OffsetX,OffsetY". *end example*]

5 A matrix transform may also be specified as a RenderTransform or Transform  
 6 property attribute using the following abbreviated matrix transformation syntax:

7 `M11,M12,M21,M22,OffsetX,OffsetY`

8 The values M11, M12, M21, and M22 control linear transformations such as rotation  
 9 and skew, while OffsetX and OffsetY provide positional translation. Some typical  
 10 affine matrix transformation examples follow.

11 *Example 14-10. Matrix scaling*

$$\begin{bmatrix} \mathbf{X \ scale-} & 0 & 0 \\ \mathbf{factor} & & \\ 0 & \mathbf{Y \ scale-} & 0 \\ & \mathbf{factor} & \\ 0 & 0 & 1 \end{bmatrix}$$

12 *end example*]

13 *Example 14-11. Matrix reversing the x axis*

$$\begin{bmatrix} \mathbf{-1} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

14 *end example*]

15 *Example 14-12. Matrix reversing the y axis*

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \mathbf{-1} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

16 *end example*]

17 *Example 14-13. Matrix skewing*

$$\begin{bmatrix} 1 & \mathbf{Y \ skew-} & 0 \\ & \mathbf{factor} & \\ \mathbf{X \ skew-} & 1 & 0 \\ & & \end{bmatrix}$$

**factor**  
0            0            1

1 *end example]*

2 *Example 14-14. Matrix Rotating*

$$\begin{bmatrix} \cos \Theta & \sin \Theta & 0 \\ -\sin \Theta & \cos \Theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

3 *end example]*

4 *Example 14-15. Matrix positioning*

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \text{OffsetX} & \text{OffsetY} & 1 \end{bmatrix}$$

5 *end example]*

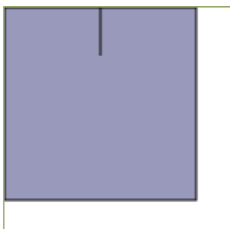
6 *Example 14-16. <MatrixTransform> usage*

7 The following markup describes a box (with the top edge marked) that is rotated 90°  
8 and shifted 50 units down and to the right:

```
9 <Path
10   Stroke="#000000"
11   Fill="#9999BB"
12   Data="M 0,0 L 60,0 L 60,25 L 60,0 L 120,0 L 120,120 L 0,120 z">
13   <Path.RenderTransform>
14     <MatrixTransform Matrix="0,1,-1,0,170,50" />
15   </Path.RenderTransform>
16 </Path>
```

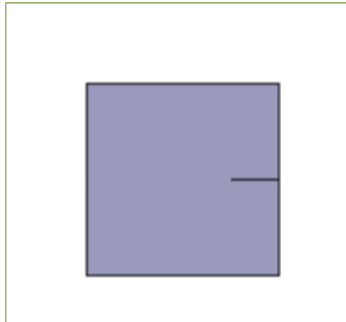
17 Since the x origin has been shifted, the overall box must be additionally shifted the  
18 width of the box to achieve the desired visual effect.

19 Before the render transformation, the box appears like this:



20

- 1 After the render transformation, the box appears like this:



- 2
- 3 *end example]*

1 *Example 14-17. Using abbreviated matrix transformation syntax*

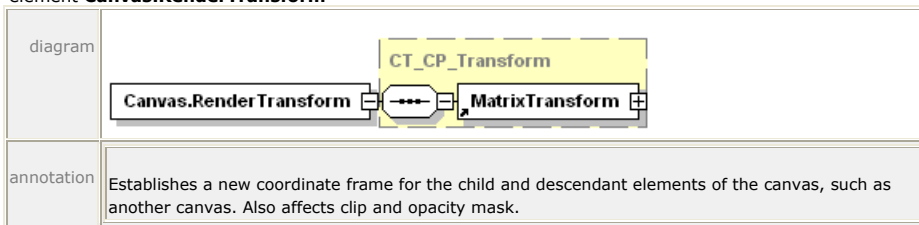
2 The following markup uses abbreviated syntax to produce the above image:

```
3 <Path
4   Stroke="#000000"
5   Fill="#9999BB"
6   Data="M 0,0 L 60,0 L 60,25 L 60,0 L 120,0 L 120,120 L 0,120 Z"
7   RenderTransform="0,1,-1,0,170,50" />
```

8 *end example]*

### 9 **14.4.2 <Canvas.RenderTransform> Element**

10 element **Canvas.RenderTransform**



11 *Example 14-18. <Canvas.RenderTransform> usage*

12 In the following markup, child elements of the canvas are positioned by the render  
13 transformation:

```
14 <Canvas>
15   <Canvas.Resources>
16     <ResourceDictionary>
17       <PathGeometry x:Key="StarFish">
18         <PathFigure StartPoint="50,0" IsClosed="true">
19           <PolyLineSegment Points="55,45 100,25 55,50 80,100 50,55
20             20,100 45,50 0,25 45,45" />
21         </PathFigure>
22       </PathGeometry>
23     </ResourceDictionary>
24   </Canvas.Resources>
25
26   <!-- Draw a green starfish shifted 25 to the right and 50 down -->
27   <Canvas>
28     <Canvas.RenderTransform>
29       <MatrixTransform Matrix="1,0,0,1,25,50" />
30     </Canvas.RenderTransform>
31     <Path Data="{StaticResource StarFish}">
32       <Path.Fill>
33         <SolidColorBrush Color="#00FF00" />
34       </Path.Fill>
35     </Path>
36   </Canvas>
37
38   <!-- Draw a red starfish shifted 100 to the right and 150 down -->
39   <Canvas>
40     <Canvas.RenderTransform>
41       <MatrixTransform Matrix="1,0,0,1,100,150" />
42     </Canvas.RenderTransform>
43     <Path Data="{StaticResource StarFish}">
44       <Path.Fill>
```

```

1      <SolidColorBrush Color="#FF0000" />
2      </Path.Fill>
3      </Path>
4      </Canvas>
5      </Canvas>

```

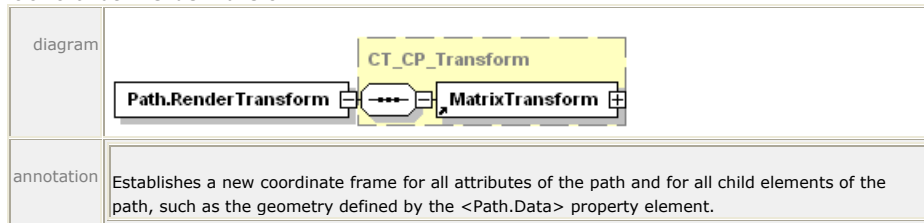
6 This markup is rendered as follows:



7  
8 *end example]*

### 9 14.4.3 <Path.RenderTransform> Element

10 element **Path.RenderTransform**



11 *Example 14-19. <Path.RenderTransform> usage*

12 The following markup describes a y-skew transformation applied to a circular path.  
13 (Before the render transformation, the middle of the right edge of the circle  
14 was marked with a horizontal line.)

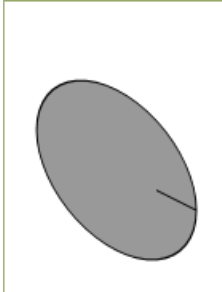
```

15 <Path
16   Fill="#999999"
17   Stroke="#000000"
18   Data="M 20,70 A 50,50 0 1 1 120,70 L 100,70 L 120,70 A 50,50 0 1 1
19     20,70 Z"
20   RenderTransform="1,0.5,0,1,0,0" />

```



1 This markup is rendered as follows:

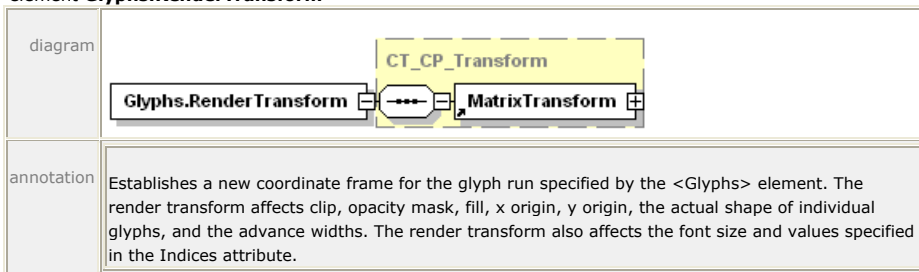


2

3 *end example]*

#### 4 14.4.4 <Glyphs.RenderTransform> Element

5 element **Glyphs.RenderTransform**



6 *Example 14-20. <Glyphs.RenderTransform> usage*

7 The following markup describes the letter J, flipped vertically and repositioned.

```
8 <Glyphs
9   Fill="#000000"
10  OriginX="20"
11  OriginY="130"
12  UnicodeString="J"
13  FontRenderingEmSize="170"
14  FontUri="./Resources/Fonts/Timesbd.ttf"
15  RenderTransform="1,0,0,-1,0,150" />
```

16 This markup is rendered as follows:

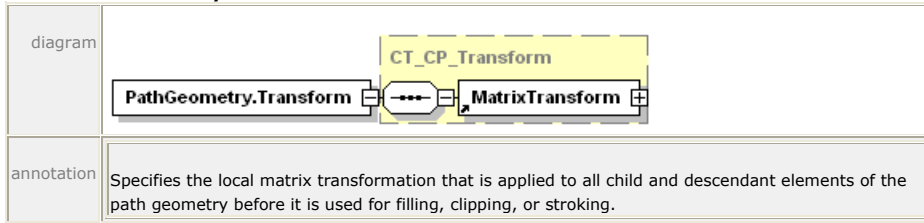


17

18 *end example]*

1 **14.4.5 <PathGeometry.Transform> Element**

2 element **PathGeometry.Transform**



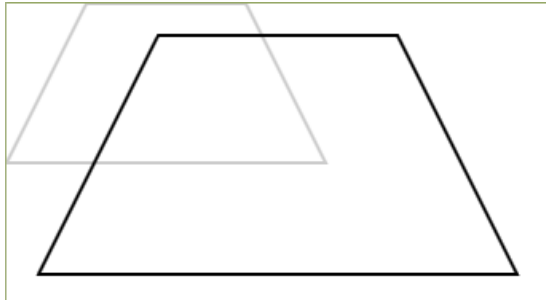
3 *Example 14–21. <PathGeometry.Transform> usage*

4 The following markup demonstrates a simple 150% zoom and positional  
5 transformation:

```

6 <Path StrokeThickness="2" Stroke="#000000">
7   <Path.Data>
8     <PathGeometry Transform="1.5,0,0,1.5,20,20">
9       <PathFigure StartPoint="50,0" IsClosed="true">
10        <PolyLineSegment Points="150,0 200,100 0,100" />
11      </PathFigure>
12    </PathGeometry>
13  </Path.Data>
14 </Path>
    
```

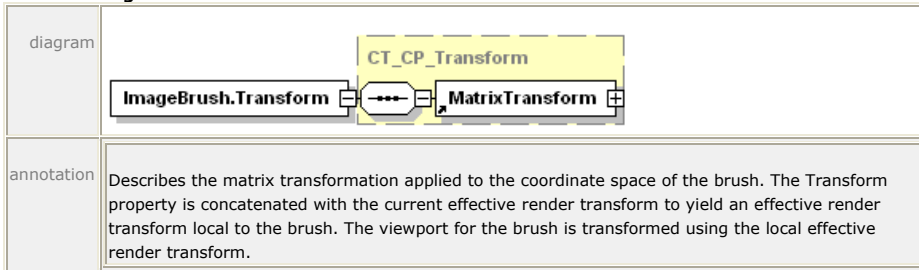
15 This markup is rendered as follows. The pre-transform path is indicated in light gray.  
16 Note that the stroke thickness did not change. If this transformation had been  
17 applied to the entire Path, the stroke thickness would also have increased by 150%.



18  
19 *end example]*

### 1 14.4.6 <ImageBrush.Transform> Element

2 element **ImageBrush.Transform**



3 The **Transform** property can result in a non-rectangular (that is, skewed) viewport  
 4 that defines the tile shape. In this circumstance, tile mode operations (FlipX, FlipY,  
 5 and FlipXY) are treated as if the tile was rectangular, a larger tile was constructed  
 6 from a 2-by-2 arrangement of regular tiles, the skew transform was applied  
 7 afterward, and the new non-rectangular tile was tiled with adjacent edges and  
 8 without flipping.

9 *Example 14-22. <ImageBrush.Transform> usage*

10 The following markup describes an image rotated 20° and repositioned within a path.  
 11 The path itself remains untransformed; the viewport of the image brush is  
 12 transformed instead.

```

13 <Path
14   StrokeThickness="5"
15   Stroke="#996666"
16   StrokeLineJoin="Round"
17   Data="M 25,25 L 350,25 L 355,250 L 25,250 Z">
18   <Path.Fill>
19     <ImageBrush
20       ImageSource="dog.jpg"
21       Transform=".939,.342,-.342,.939,0,-80"
22       TileMode="Tile"
23       Viewbox="0,0,270,423"
24       ViewboxUnits="Absolute"
25       Viewport="75,75,90,125"
26       ViewportUnits="Absolute" />
27   </Path.Fill>
28 </Path>

```

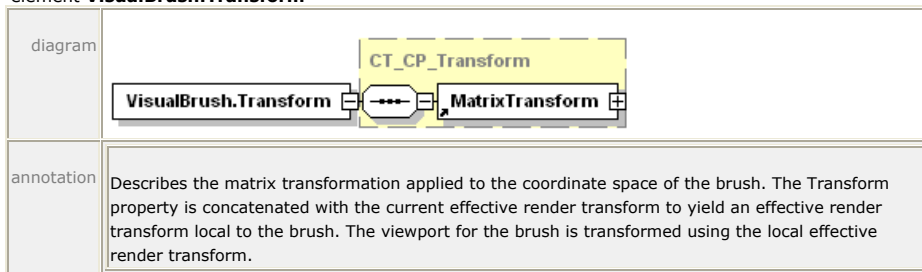
1 This markup is rendered as follows:



2  
3 *end example]*

#### 4 **14.4.7 <VisualBrush.Transform> Element**

5 element **VisualBrush.Transform**



6 The **Transform** property can result in a non-rectangular (that is, skewed) viewport  
7 that defines the tile shape. In this circumstance, tile mode operations (FlipX, FlipY,  
8 and FlipXY) are treated as if the tile was rectangular, a larger tile was constructed  
9 from a 2-by-2 arrangement of regular tiles, the skew transform was applied  
10 afterward, and the new non-rectangular tile was tiled with adjacent edges and  
11 without flipping.

12 *Example 14-23. <VisualBrush.Transform> usage*

13 The following markup describes a solid background and vertical pinstripe rotated 45°  
14 to fill a frame:

```
15 <Path
16   StrokeThickness="5"
17   Stroke="#336666"
18   StrokeLineJoin="Round"
19   Data="M 25,25 L 365,25 L 365,250 L 25,250 Z M 70,70 L 320,70
20         L 320,205 L 70,205 Z">
21   <Path.Fill>
22     <VisualBrush
```

```
1      Transform=".707,.707,-.707,.707,0,0"  
2      TileMode="Tile"  
3      Viewbox="0,0,60,100"  
4      ViewboxUnits="Absolute"  
5      Viewport="25,25,50,50"  
6      ViewportUnits="Absolute">  
7      <VisualBrush.Visual>  
8          <Canvas>  
9              <Path  
10                 Fill="#99CCCC"  
11                 Data="M 0,0 L 60,0 L 60,100 L 0,100 Z" />  
12                 <Path  
13                     Stroke="#336666"  
14                     Data="M 0,0 L 0,100 M 20,0 L 20,100 M 40,0 L 40,100  
15                         M 60,0 L 60,100 M 80,0 L 80,100" />  
16             </Canvas>  
17         </VisualBrush.Visual>  
18     </VisualBrush>  
19 </Path.Fill>  
20 </Path>
```

21 This markup is rendered as follows:



22

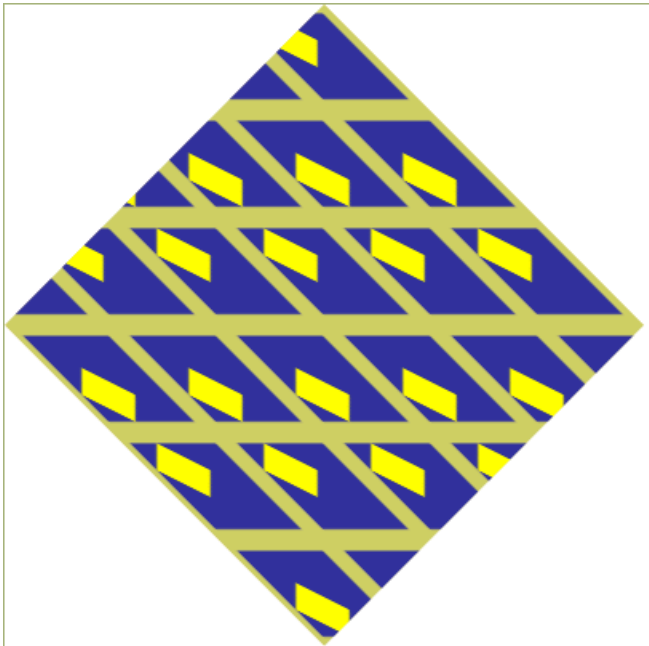
23 *end example]*

1 *Example 14-24. <VisualBrush.Transform> usage with tiling behavior*

2 This example demonstrates tile rendering behavior when applying a transform.

```
3 <!-- Draw background diamond to show where fill affects background -->
4 <Path Fill="#CCCC66" Data="M 200,0 L 400,200 L 200,400 L 0,200 Z" />
5 <Path Data="M 200,0 L 400,200 L 200,400 L 0,200 Z">
6   <Path.Fill>
7     <VisualBrush
8       Viewbox="0,0,1,1"
9       Viewport="200,133,67,67"
10      ViewboxUnits="Absolute"
11      ViewportUnits="Absolute"
12      Transform="1,0,1,1,0,0"
13      TileMode="FlipY">
14     <VisualBrush.Visual>
15       <Canvas>
16         <Path Fill="#333399" Data="M 0.1,0.1 L 0.9,0.1 L 0.9,0.9
17           L 0.1,0.9 Z" />
18         <Path Fill="#FFFF00" Data="M 0.1,0.35 L 0.35,0.1
19           L 0.6,0.35 L 0.35,0.6 Z" />
20       </Canvas>
21     </VisualBrush.Visual>
22   </VisualBrush>
23 </Path.Fill>
24 </Path>
```

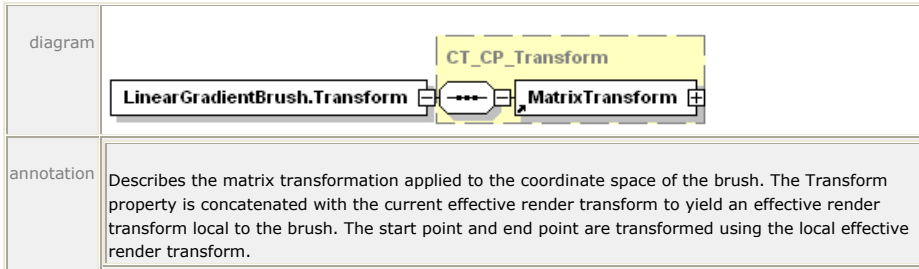
25 This markup is rendered as follows:



26  
27 *end example]*

### 1 14.4.8 <LinearGradientBrush.Transform> Element

2 element **LinearGradientBrush.Transform**



3 *Example 14-25. <LinearGradientBrush.Transform> usage*

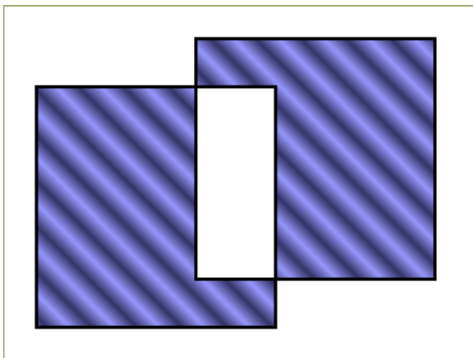
4 The following markup demonstrates a transform applied to the brush directly:

```

5 <Path Stroke="#000000" StrokeThickness="2" Data="M 20,50 L 170,50 L
6 170,200 L 20,200 Z M 120,20 L 270,20 L 270,170 120,170 Z">
7   <Path.Fill>
8     <LinearGradientBrush
9       MappingMode="Absolute"
10      Transform=".707,.707,-.707,.707,150,-30"
11      StartPoint="0,0"
12      EndPoint="0,10"
13      SpreadMethod="Reflect">
14       <LinearGradientBrush.GradientStops>
15         <GradientStop Color="#9999FF" Offset="0.0"/>
16         <GradientStop Color="#333366" Offset="1.0"/>
17       </LinearGradientBrush.GradientStops>
18     </LinearGradientBrush>
19   </Path.Fill>
20 </Path>

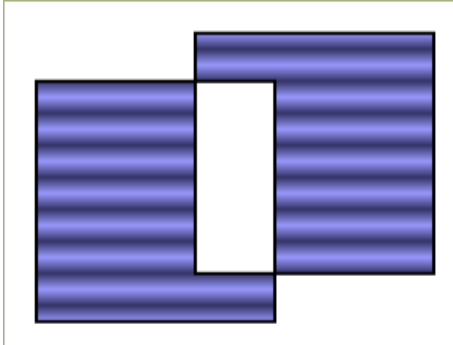
```

21 This markup is rendered as follows:



22

1 Without the Transform property, this markup would be rendered as follows:



2  
3 *end example]*

#### 4 **14.4.9 <RadialGradientBrush.Transform> Element**

5 element **RadialGradientBrush.Transform**

diagram	
annotation	<p>Describes the matrix transformation applied to the coordinate space of the brush. The Transform property is concatenated with the current effective render transform to yield an effective render transform local to the brush. The ellipse defined by the center, gradient origin, x radius, and y radius vaules is transformed using the local effective render transform.</p>

6 *Example 14-26. <RadialGradientBrush.Transform> usage*

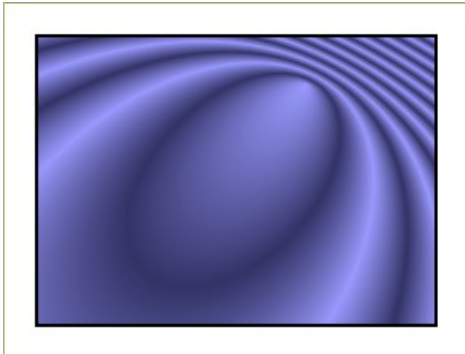
7 The following markup describes a rotation and reposition transform on a radial  
8 gradient:

```

9 <Path
10   Stroke="#000000"
11   StrokeThickness="2"
12   Data="M 20,20 L 270,20 L 270,200 L 20,200 Z"
13   <Path.Fill>
14     <RadialGradientBrush
15       MappingMode="Absolute"
16       Transform=".707,.707,-.707,.707,150,-10"
17       Center="80,90"
18       RadiusX="50"
19       RadiusY="80"
20       GradientOrigin="70,15"
21       SpreadMethod="Reflect">
22       <RadialGradientBrush.GradientStops>
23         <GradientStop Color="#9999FF" Offset="0.0" />
24         <GradientStop Color="#333366" Offset="1.0" />
25       </RadialGradientBrush.GradientStops>
26     </RadialGradientBrush>
27   </Path.Fill>
28 </Path>
```

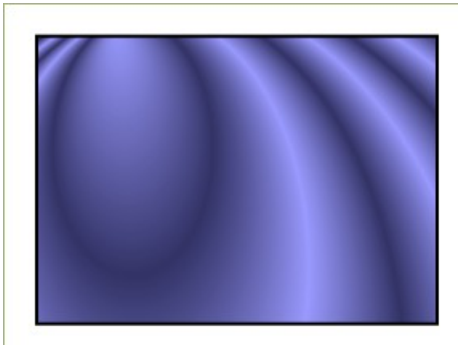


1 This markup is rendered as follows:



2

3 Without the Transform property, this markup is rendered as follows:



4

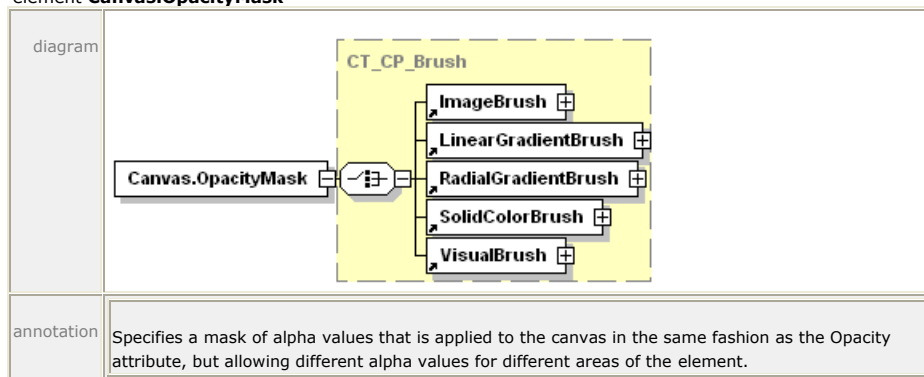
5 *end example]*

## 1 14.5 OpacityMask

2 The OpacityMask property defines a variable alpha mask for the parent element. The  
3 alpha for areas not marked by the brush is 0.0.

### 4 14.5.1 <Canvas.OpacityMask> Element

5 element **Canvas.OpacityMask**



6 *Example 14-27. <Canvas.OpacityMask> usage*

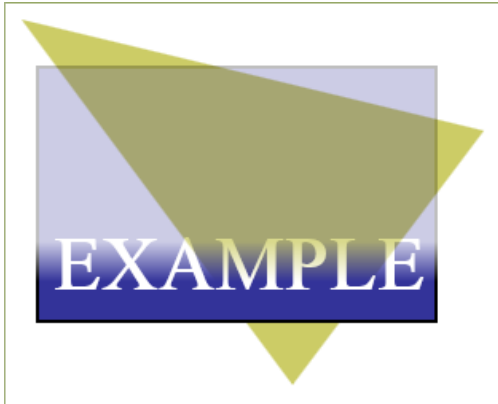
7 In the following markup, the contents of the canvas are opaque with respect to each  
8 other, but both elements are blended with the background triangle:

```

9 <Path Fill="#CCCC66" Data="M 10,10 L 300,80 L 180,240 Z" />
10 <Canvas>
11 <Canvas.OpacityMask>
12 <LinearGradientBrush
13 MappingMode="Absolute"
14 StartPoint="0,150"
15 EndPoint="0,175"
16 SpreadMethod="Pad">
17 <LinearGradientBrush.GradientStops>
18 <GradientStop Color="#40000000" Offset="0.0" />
19 <GradientStop Color="#FF000000" Offset="1.0" />
20 </LinearGradientBrush.GradientStops>
21 </LinearGradientBrush>
22 </Canvas.OpacityMask>
23 <Path
24 Stroke="#000000"
25 StrokeThickness="2"
26 Fill="#333399"
27 Data="M 20,40 L 270,40 L 270,200 L 20,200 Z" />
28 <Glyphs
29 OriginX="30"
30 OriginY="180"
31 UnicodeString="EXAMPLE"
32 FontUri=" ../Resources/Fonts/Timesbd.ttf"
33 FontRenderingEmSize="48"
34 Fill="#FFFFFF" />
35 </Canvas>

```

1 This markup is rendered as follows:

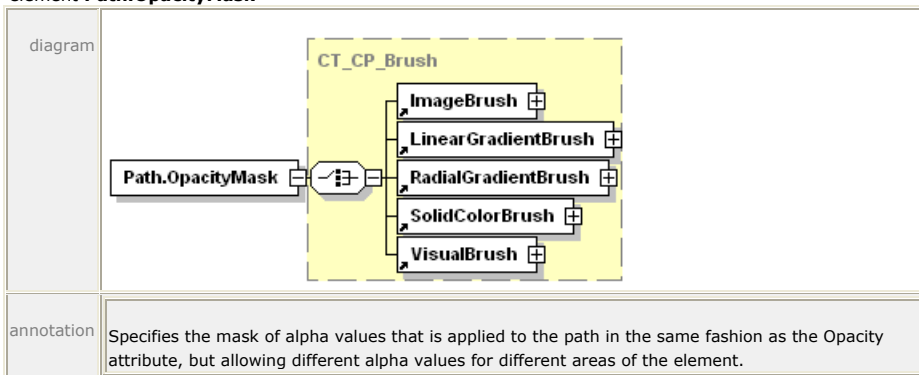


2

3 *end example]*

#### 4 14.5.2 <Path.OpacityMask> Element

5 element **Path.OpacityMask**



6 *Example 14–28. <Path.OpacityMask> usage*

7 The following markup describes a path that has a linear gradient both for the opacity  
8 mask and for the fill:

```

9 <Path
10   Stroke="#000000"
11   StrokeThickness="2"
12   Fill="#CCCC66"
13   Data="M 135,10 L 270,250 L 20,250 Z" />
14 <Path
15   Stroke="#000000"
16   StrokeThickness="2"
17   Data="M 20,40 L 270,40 L 270,200 L 20,200 Z">
18   <Path.OpacityMask>
19     <LinearGradientBrush
20       MappingMode="Absolute"
21       StartPoint="0,60"

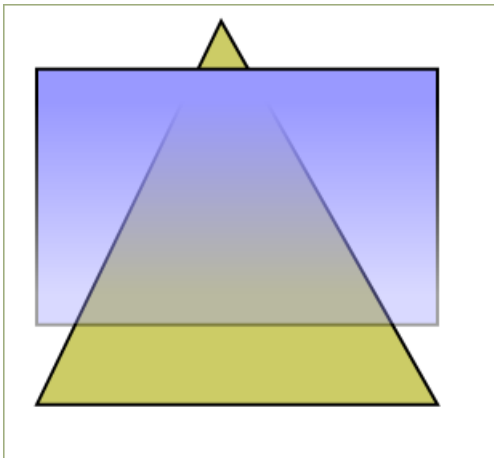
```

```

1      EndPoint="0,180"
2      SpreadMethod="Pad">
3      <LinearGradientBrush.GradientStops>
4          <GradientStop Color="#FF000000" Offset="0.0" />
5          <GradientStop Color="#60000000" Offset="1.0" />
6      </LinearGradientBrush.GradientStops>
7      </LinearGradientBrush>
8  </Path.OpacityMask>
9  <Path.Fill>
10     <SolidColorBrush Color="#9999FF" />
11 </Path.Fill>
12 </Path>

```

13 This markup is rendered as follows:

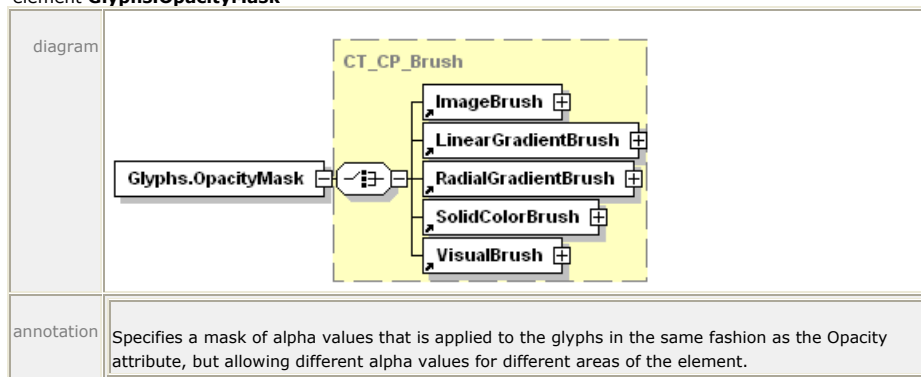


14

15 *end example]*

### 16 14.5.3 <Glyphs.OpacityMask> Element

17 element **Glyphs.OpacityMask**



1 *Example 14–29. <Glyphs.OpacityMask> usage*

2 The following markup demonstrates the use of an opacity mask to create a tile  
3 effect:

```
4 <Path Fill="#CCCC66" Data="M 40,40 L 480,40 L 260,120 Z" />
5 <Glyphs
6   OriginX="20"
7   OriginY="95"
8   UnicodeString="EXAMPLE"
9   FontUri="./Resources/Fonts/Timesbd.ttf"
10  FontRenderingEmSize="100"
11  Fill="#000080">
12  <Glyphs.OpacityMask>
13    <VisualBrush
14      Viewbox="0,0,2,2"
15      ViewboxUnits="Absolute"
16      Viewport="0,0,6,6"
17      ViewportUnits="Absolute"
18      TileMode="Tile">
19      <VisualBrush.Visual>
20        <Path
21          Fill="#CC000000"
22          Data="M 0,0 L 1.5,0 L 1.5,1.5 L 0,1.5 Z" />
23        </VisualBrush.Visual>
24      </VisualBrush>
25    </Glyphs.OpacityMask>
26  </Glyphs>
```

27 This markup is rendered as follows:



28

29 *end example]*



## 1 15. Color

2 The mechanisms described in this clause for storing advanced color information in  
3 XPS Documents apply to both vector graphics (including text) and raster images.  
4 Color producers such as digital cameras and consumers such as printers can store  
5 and render significantly more color information than many display devices can render  
6 (typically 8 bits per channel). Storing the advanced color information in an XPS  
7 Document and passing it through to printing consumers enables greater end-to-end  
8 color fidelity.

---

### 9 15.1 Color Support

10 XPS Documents support sRGB and rich color spaces, including scRGB, CMYK, N-  
11 Channel, and named colors. Consumers MUST support the following color features:

- 12 • sRGB colors (8 bit-per-channel) in vector data, with and without alpha [M8.1]
- 13 • sRGB colors in image data, using the JPEG, PNG, TIFF, or Windows Media  
14 Photo image formats [M8.2]
- 15 • scRGB color specification in vector data, with and without alpha [M8.3]
- 16 • scRGB colors in image data, using the Windows Media Photo image format  
17 [M8.4]
- 18 • CMYK colors in vector data [M8.5]
- 19 • CMYK colors in image data, using the TIFF or Windows Media Photo image  
20 formats [M8.6]
- 21 • N-Channel colors in vector data [M8.7]
- 22 • N-Channel colors in image data, using the Windows Media Photo image format  
23 [M8.8]
- 24 • ICC Version 2 profiles for 3-, 4-, 5-, 6-, 7-, and 8-channel color [M8.9]
- 25 • ICC Version 2 profiles with a Windows Color System (WCS) profile embedded  
26 as a private tag [M8.10]

27 When non-sRGB color information is used, color value specifications are expressed  
28 using markup from the XPS Document schema.

29 Consumers are not required to handle all color spaces natively, but rather MAY  
30 convert data specified in a rich color space to sRGB at an early stage [O8.1].  
31 Consumers that do not handle rich colors natively may experience reduced fidelity.

#### 32 15.1.1 sRGB Color Space

33 The XPS Document format supports colors in the sRGB color space for both vector  
34 and raster graphics.

**1 15.1.2 scRGB Color Space**

2 The XPS Document format supports colors in the scRGB color space for both vector  
3 and raster graphics. See G for the scRGB gamut definition.

**4 15.1.3 Gray Color Space**

5 Gray colors for vector elements can be specified as sRGB or scRGB colors with the  
6 red, blue and green components set to the same value. Gray colors for raster images  
7 can be specified using any image format.

**8 15.1.4 CMYK Color Space**

9 CMYK color is supported through the use of color management transformations from  
10 an ICC profile.

**11 15.1.5 N-Channel Color Spaces**

12 N-channel color is supported through the use of color management transformations  
13 from an ICC profile.

**14 15.1.6 Named Color for Spot Colors and N-tone Images**

15 Named colors are supported for images and spot colors through the use of color  
16 management transformations from an ICC profile.

**17 15.1.7 Device Color Spaces**

18 To specify colors in the native color space (usually CMYK or N-Channel) for a device,  
19 use the standard markup with an ICC profile that approximates the device color  
20 space. Include the PageDeviceColorSpaceProfileURI PrintTicket setting to indicate to  
21 the device that this particular color specification is a native device color and MUST  
22 NOT be color-managed according to the included profile unless forced to do so for  
23 transparency effects [M8.11].

24 It is the responsibility of the consumer to identify the ICC profile as one that  
25 correctly approximates the native colors of the device.

**26 15.1.8 ICC Profiles**

27 XPS Documents MAY include ICC profile parts [O2.3]. ICC profiles MUST conform to  
28 the requirements in the ICC Color Profile specification, Version 3.4 [M8.12].

29 Consumers MUST support color profiles as specified in the ICC specification [M8.9].  
30 However, the set of usable N-component LUT-based profiles is limited to 3-, 4-, 5-,  
31 6-, 7-, or 8-color channels.

32 All ICC profiles used in XPS Documents MUST be one of the following [M8.13]:

- 33 • Input
- 34 • Output
- 35 • Monitor (RGB)
- 36 • ColorSpace Conversion



1 One tag is explicitly supported as specified in ICC Version 4.0.0 profiles: the colorant  
 2 table for named colors. A consumer incapable of supporting named colors SHOULD  
 3 treat this tag as a user-defined custom tag, and therefore ignore it. It SHOULD  
 4 instead use the color tables as provided in the profile to convert the specified colors  
 5 to the Profile Connection Space (PCS) [S8.14].

6 ICC profiles SHOULD be used when embedded in any image format with any color  
 7 space, except the sRGB color space, for which the gamut boundary described in G is  
 8 assumed [S8.1].

9 [Note: Some consumers do not correctly apply ICC profiles to grayscale images. *end*  
 10 *note*] If consistency of appearance is important, the producer SHOULD adjust the  
 11 gray tone response curve of the image before adding it to the XPS Document [S8.2].

12 An ICC profile MAY contain the private tag, "MS00", which specifies an embedded  
 13 Windows Color System (WCS) profile [O8.2].

#### 14 **15.1.9 WcsProfilesTag**

15 The WcsProfilesTag (Windows color space profiles tag) is a private Microsoft ICC  
 16 profile tag that is used in ICC profiles created by WcsCreateIccProfile to contain input  
 17 WCS profiles. This tag conforms to ICC profile requirements for profile tags. In  
 18 particular, the tag header is in big-endian byte ordering, but the embedded WCS  
 19 XML profiles remain in their native byte order. Furthermore, the tag data must be  
 20 aligned on a 4-byte boundary (measured from the start of the ICC profile). The  
 21 structure of the tag is defined by the WcsProfilesTagType below.

22 The WcsProfilesTag signature is "MS00". This is the tag signature that will appear in  
 23 the ICC profiles tag table for the WcsProfilesTag.

24 *Table 15-3. WcsProfilesTagType structure*

Byte offset	Content
0-3	MS10-type signature.
4-7	Reserved, must be set to 0 (according to ICC convention).
8-11	Byte offset from the beginning of the tag to the CDMP data.
12-15	Size of the CDMP data in bytes.
16-19	Byte offset from the beginning of the tag to the CAMP data.
20-23	Size of the CAMP data in bytes.
24-27	Byte offset from the beginning of the tag to the GMMP data.
28-31	Size of the GMMP data in bytes.
32- <i>n</i>	A sequence of bytes of length <i>element_size</i> -32, where <i>element_size</i> is the tag size recorded in the ICC profile tag table entry for this tag. These are the WCS XML profiles that were used to create this ICC profile via WcsCreateIccProfileFromWcsProfileHandle. The WCS profiles are ordered as follows: the CDMP (required) first, followed by the CAMP (if present), followed by the GMMP (if present).

1 **15.1.10 WCS Color Profiles**

2 XPS Documents include only WCS color profiles embedded in ICC color profiles, as  
3 described above.

4 **15.1.11 Vector Color Syntax**

5 Vector colors may be specified in XPS Document markup in the following locations:

- 6 • The **Color** attribute of the <SolidColorBrush> element
- 7 • The **Color** attribute of the <GradientStop> element
- 8 • The **Fill** attribute of the <Path> element
- 9 • The **Fill** attribute of the <Glyphs> element
- 10 • The **Stroke** attribute of the <Path> element

11 The last three locations are an abbreviated syntax for expressing a solid color brush  
12 with the specified color.

1 Table 15-1. Syntax summary

Color type	Syntax	Example
sRGB w/o alpha	Color="#RRGGBB"	Color="#FFFFFF"
sRGB with alpha	Color="#AARRGGBB"	Color="#80FFFFFF"
scRGB w/o alpha	Color="sc#RedFloat, GreenFloat,BlueFloat"	Color="sc#1.0,0.5,1.0"
scRGB with alpha	Color="sc#AlphaFloat,RedFloat, GreenFloat,BlueFloat"	Color="sc#0.3,1.0,0.5,1.0"
CMYK with alpha	Color="ContextColor ProfileURI AlphaFloat, Chan0Float, Chan1Float, Chan2Float, Chan3Float"	Color="ContextColor /swopcmypfile.icc 1.0,1.0,0.0,0.0,0.0"
N-Channel with alpha	Color="ContextColor ProfileURI AlphaFloat, Chan0Float, ..., ChanN-1Float"	Color="ContextColor /5nchannelprofile.icc 1.0, 1.0, 0.0, 0.0, 1.0, 0.0"
Named color with alpha	Color="ContextColor ProfileURI AlphaFloat, TintFloat, 0, 0"	Color="ContextColor /namedtintprofile.icc 1.0, 1.0, 0, 0"

2 Real numbers specified for color channel values of scRGB and ContextColor colors  
3 MUST NOT use exponent forms of numbers [M8.14].

#### 4 15.1.12 sRGB Color Syntax

5 The sRGB color syntax is the same as that used in HTML, with the red, green, and  
6 blue channels represented by two hexadecimal digits. XPS Documents may specify  
7 an sRGB color either with or without an alpha channel value, which is also expressed  
8 as two hexadecimal digits.

9 The syntax is as follows (without alpha):

10 #RRGGBB

11 or (with alpha):

12 #AARRGGBB

13 When an sRGB color is specified without an alpha value, an alpha of "FF" is implied.

#### 14 15.1.13 scRGB Color Syntax

15 The scRGB color syntax allows XPS Document producers to specify a color using the  
16 full scRGB color space, which is much larger than the sRGB color space and can  
17 represent the entire range of colors perceivable by the human eye.

1 This syntax is expressed either as:

```
2 sc#RedFloat,GreenFloat,BlueFloat
```

3 or:

```
4 sc#AlphaFloat,RedFloat,GreenFloat,BlueFloat
```

5 When an scRGB color is specified with three numeric values, an alpha of 1.0 is  
6 implied. When an scRGB color is specified with four numeric values, the first value is  
7 the alpha channel. Although alpha values smaller than 0.0 and larger than 1.0 can be  
8 specified, they MUST be clamped to the valid range from 0.0 to 1.0 before any  
9 further processing [M8.15].

#### 10 **15.1.14 CMYK Color Syntax**

11 XPS Document producers specify CMYK colors using the context color syntax, which  
12 allows specification of an ICC profile and the individual color channel values as real  
13 numbers.

14 The syntax is as follows:

```
15 ContextColor ProfileURI AlphaFloat, Chan0Float, Chan1Float, Chan2Float,  
16 Chan3Float
```

17 ProfileURI specifies a part containing the binary data of the color profile. The profile  
18 URI MUST be added as a Required Resource relationship to the FixedPage part  
19 [M2.10].

20 Although alpha values smaller than 0.0 and larger than 1.0 can be specified, they  
21 MUST be clamped to the valid range from 0.0 to 1.0 before any further processing  
22 [M8.16]. Channel float values MUST also be clamped to the valid range from 0.0 to  
23 1.0 before further processing. If the value is used as input for an ICC profile color  
24 transformation, it MUST subsequently be linearly scaled to the range from 0 to 255  
25 or from 0 to 65535, depending on whether the profile uses 8-bit or 16-bit input  
26 tables [M8.31].

#### 27 **15.1.15 N-Channel Color Syntax**

28 XPS Document producers specify N-channel colors using the context color syntax,  
29 which allows specification of an ICC profile and the individual color channel values as  
30 real numbers. The syntax is expressed as follows:

```
31 ContextColor ProfileURI AlphaFloat,Chan0Float,...,ChanN-1Float
```

32 ProfileURI specifies a part containing the binary data of the color profile. The profile  
33 URI MUST be added as a Required Resource relationship to the FixedPage part  
34 [M2.10]. The profile can be a 3-, 4-, 5-, 6-, 7- or 8-channel profile. The context color  
35 MUST specify a matching number of channel float values, setting unused ones to 0.0  
36 [M8.17].

37 Although alpha values smaller than 0.0 and larger than 1.0 can be specified, they  
38 MUST be clamped to the valid range from 0.0 to 1.0 before any further processing  
39 [M8.18]. Channel float values MUST also be clamped to the valid range from 0.0 to  
40 1.0 before further processing. If the value is used as input for an ICC profile color  
41 transformation, it MUST subsequently be linearly scaled to the range from 0 to 255

- 1 or from 0 to 65535, depending on whether the profile uses 8-bit or 16-bit input
- 2 tables [M8.31].
- 3 To represent a 1-channel profile, create a 3-channel profile and use only the first
- 4 channel. To represent a 2-channel profile, create a 3-channel profile and use only the
- 5 first and second channels.

### 1 15.1.16 Named Color Syntax

2 A `named color` is an industry-defined color specification that identifies a particular  
 3 color in a well-defined color schema, usually for the purpose of printing. There are  
 4 currently several named color schemas. Producers specify named colors using the  
 5 context color syntax, which allows specification of a named color using an ICC  
 6 profile. The named color schema used is determined by the ICC profile.

7 A named color is expressed as a combination of an ink name stored in the ICC profile  
 8 and a tint level (percentage ink dilution). The ink name for the tint is contained in  
 9 the `colorantTable clrt` tag. This tag is not defined for ICC Version 3.4 profiles, and its  
 10 presence is benignly ignored by ICM V2 implementations. Therefore, it is used in XPS  
 11 Documents to specify the names of named colors.

12 The syntax for referencing a single named color is as follows:

```
13 ContextColor ProfileURI AlphaFloat,TintFloat,0,0
```

14 `ProfileURI` specifies a part containing the binary data of the color profile. The profile  
 15 URI MUST be added as a Required Resource relationship to the `FixedPage` part  
 16 [M2.10]. The profile MUST have 3, 4, 5, 6, 7 or 8 channels (and an  $n$ CLR signature,  
 17 where  $n$  is the number of channels), mapping to a valid PCS [M8.19]. To represent a  
 18 single tone or duotone profile, create a 3-channel profile and specify a colorant name  
 19 in the profile's `colorantTable` with zero-length (name field all set to 0) for the unused  
 20 channels.

21 `AlphaFloat` specifies the alpha to be applied to the named color. `TintFloat` specifies  
 22 how diluted with respect to the color schema's white color point the named color is,  
 23 with 1.0 being the pure named color and 0.0 being fully diluted. The final two values  
 24 are always set to 0 to specify a single named color.

25 For duotone named colors, use the first two values and set the final value to 0.

26 *[Example: The syntax for referencing a duotone named color is as follows:*

```
27 ContextColor ProfileURI AlphaFloat,Tint0Float,Tint1Float,0
```

28 *end example]*

29 Although alpha values smaller than 0.0 and larger than 1.0 can be specified, they  
 30 MUST be clamped to the valid range from 0.0 to 1.0 before any further processing  
 31 [M8.20]. The tint float value MUST also be clamped to the valid range from 0.0 to  
 32 1.0 before further processing. If the value is used as input for an ICC profile color  
 33 transformation, it MUST subsequently be linearly scaled to the range from 0 to 255  
 34 or from 0 to 65535, depending on whether the profile uses 8-bit or 16-bit input  
 35 tables [M8.31].

36 Consumers that do not understand the named color MUST compute a color  
 37 approximation through ICM-compliant color management functions using the  
 38 specified profile. When a named color is used in a gradient brush or with  
 39 transparency, the result produced by consumers that are not named-color aware  
 40 MAY differ significantly from the result produced by consumers that are named-color  
 41 aware [O8.3].

42 An XPS consumer that is aware of named colors looks for the `clrt` tag to find out if a  
 43 specific `ContextColor` designates a named color description. A producer of XPS

1 documents containing named colors SHOULD create the color profile in such a way  
2 that a linear ramp of the channel values corresponding to a named colorant maps to  
3 PCS values resulting in the same color appearance for consumers unaware of named  
4 colors (or the specific colorant) [S8.3].

5 This method can be used for 1 to 8 named colors; however, the ContextColor syntax  
6 requires a minimum of 1 Alpha value and 3 Channel values. It is RECOMMENDED  
7 that a 1 or 2 tone profile uses the first 1 or 2 channels, respectively, and specifies 0  
8 for the remaining channels [S8.4].

9 If the consumer does not know ALL of the colorants named in the clr tag, it SHOULD  
10 treat the profile as if it were a regular N-channel source profile and SHOULD NOT  
11 attempt to use any of the known colorants, as that would result in undefined results  
12 [S8.6].

13 A named color profile MAY be used with images [O8.4], which is especially useful for  
14 the reproduction of corporate logo images using strictly controlled ink sets. A named  
15 profile could contain SWOP CMYK in the first 4 channels, and use the remaining 4  
16 channels for highlight colors. This allows the use of up to 4 accent colors in images.

---

## 17 15.2 Rich Colors in Raster Images

18 This section describes specific considerations for including raster images with rich  
19 colors in XPS Documents.

### 20 15.2.1 sRGB Raster Images

21 XPS Documents support sRGB raster images in the following formats:

- 22 • JPEG
- 23 • PNG
- 24 • TIFF
- 25 • Windows Media Photo

26 The following Windows Media Photo pixel formats are supported:

- 27 • WICPixelFormat24bppRGB
- 28 • WICPixelFormat24bppBGR
- 29 • WICPixelFormat32bppBGR
- 30 • WICPixelFormat32bppBGRA
- 31 • WICPixelFormat32bppPBGRA
- 32 • WICPixelFormat48bppRGB
- 33 • WICPixelFormat64bppRGBA
- 34 • WICPixelFormat64bppPRGBA

35 Pixel formats WICPixelFormat32bppPBGRA and WICPixelFormat64bppPRGBA are pre-  
36 multiplied alpha formats. See §18.4.1, "Pre-Multiplied Alpha and Superluminous  
37 Colors," on page 281 for details.

1 The following Windows Media Photo packed pixel formats are supported:

- 2 • WICPixelFormat16bppBGR555
- 3 • WICPixelFormat16bppBGR565
- 4 • WICPixelFormat32bppBGR101010

5 See §9.1.5, “Image Parts,” on page 24 for more details.

### 6 **15.2.2 scRGB Raster Images**

7 XPS Documents support scRGB raster images only in the Windows Media Photo  
8 image format. The following pixel formats are supported:

- 9 • WICPixelFormat48bppRGBFixedPoint
- 10 • WICPixelFormat48bppRGBHalf
- 11 • WICPixelFormat96bppRGBFixedPoint
- 12 • WICPixelFormat128bppRGBFloat
- 13 • WICPixelFormat64bppRGBAFixedPoint
- 14 • WICPixelFormat64bppRGBFixedPoint
- 15 • WICPixelFormat64bppRGBHalf
- 16 • WICPixelFormat64bppRGBHalf
- 17 • WICPixelFormat128bppRGBAFixedPoint
- 18 • WICPixelFormat128bppRGBFixedPoint
- 19 • WICPixelFormat128bppRGBFloat
- 20 • WICPixelFormat128bppPRGBAFloat
- 21 • WICPixelFormat32bppRGBE

22 Pixel format WICPixelFormat128bppPRGBAFloat is a pre-multiplied alpha format. See  
23 18.4.1, “Pre-Multiplied Alpha and Superluminous Colors,” on page 281 for details.

### 24 **15.2.3 Gray Raster Images**

25 XPS Documents support gray raster images in the following formats:

- 26 • JPEG
- 27 • PNG
- 28 • TIFF
- 29 • Windows Media Photo

30 The following Windows Media Photo pixel formats are supported:

- 31 • WICPixelFormatBlackWhite
- 32 • WICPixelFormat8bppGray
- 33 • WICPixelFormat16bppGray
- 34 • WICPixelFormat16bppGrayFixedPoint (scRGB range)
- 35 • WICPixelFormat16bppGrayHalf (scRGB range)



- 1 • WICPixelFormat32bppGrayFixedPoint (scRGB range)
- 2 • WICPixelFormat32bppGrayFloat

### 3 **15.2.4 CMYK Raster Images**

4 CMYK images are stored in TIFF 6.0 or Windows Media Photo format.

#### 5 **15.2.4.1 TIFF CMYK Raster Images**

6 CMYK TIFF image tags are described in §9.1.5.3, "TIFF Images."

7 ICC profiles can be associated with CMYK raster images by using an ICC profile  
8 embedded in the TIFF file (tag 34675) or associated using the mechanism described  
9 in §15.2.8, "Images and Color Profile Association," on page 216.

#### 10 **15.2.4.2 Windows Media Photo CMYK Raster Images**

11 The Windows Media Photo CMYK format is described in the Windows Media Photo  
12 specification. The following formats are supported:

- 13 • WICPixelFormat32bppCMYK
- 14 • WICPixelFormat40bppCMYKAlpha
- 15 • WICPixelFormat64bppCMYK
- 16 • WICPixelFormat80bppCMYKAlpha

#### 17 **15.2.4.3 JPEG CMYK Raster Images**

18 Support for JPEG CMYK images varies by implementation and SHOULD NOT be used  
19 in XPS Documents [S2.7]. See §9.1.5.1, "JPEG Images" for more details.

### 20 **15.2.5 N-channel Raster Images**

21 N-channel images are stored in the Windows Media Photo image file format using an  
22 ICC profile. The following formats are supported:

- 23 • WICPixelFormat24bpp3Channels, WICPixelFormat48bpp3Channels
- 24 • WICPixelFormat32bpp4Channels, WICPixelFormat64bpp4Channels
- 25 • WICPixelFormat40bpp5Channels, WICPixelFormat80bpp5Channels
- 26 • WICPixelFormat48bpp6Channels, WICPixelFormat96bpp6Channels
- 27 • WICPixelFormat56bpp7Channels, WICPixelFormat112bpp7Channels
- 28 • WICPixelFormat64bpp8Channels, WICPixelFormat128bpp8Channels
- 29 • WICPixelFormat32bpp3ChannelsAlpha, WICPixelFormat64bpp3ChannelsAlpha
- 30 • WICPixelFormat40bpp4ChannelsAlpha, WICPixelFormat80bpp4ChannelsAlpha
- 31 • WICPixelFormat48bpp5ChannelsAlpha, WICPixelFormat96bpp5ChannelsAlpha
- 32 • WICPixelFormat56bpp6ChannelsAlpha, WICPixelFormat112bpp6ChannelsAlpha
- 33 • WICPixelFormat64bpp7ChannelsAlpha, WICPixelFormat128bpp7ChannelsAlpha
- 34 • WICPixelFormat72bpp8ChannelsAlpha, WICPixelFormat144bpp8ChannelsAlpha

### 1 **15.2.6 Named Color Raster Images**

2 Named color (N-tone) raster images are stored in the Windows Media Photo image  
3 file format using an ICC profile that maps the tint channel combinations to valid PCS  
4 values. See §15.2.5, "N-channel Raster Images," on page 215 for pixel format  
5 definitions.

6 Consumers unaware of named colors can then compute color approximations using  
7 the PCS values computed from the profile.

### 8 **15.2.7 Device Color Raster Images**

9 Device color (N-channel) raster images are stored in the Windows Media Photo image  
10 file format in the same manner as a named color raster image. See §15.1.7, "Device  
11 Color Spaces," on page 206 for more details. RGB and CMYK raster images can also  
12 be stored in the TIFF image file format. JPEG CMYK images SHOULD NOT be used  
13 [S2.7].

### 14 **15.2.8 Images and Color Profile Association**

15 Images can use a color profile matching the channel configuration of the image using  
16 one of two methods:

- 17 • Color profile embedded in an image using the image format specific  
18 mechanism
- 19 • Color profile contained in a separate part associated with the image using the  
20 following markup:

```
21 <ImageBrush ImageSource="{ColorConvertedBitmap image.tif profile.icc}"  
22 ... />
```

23 An associated color profile overrides an embedded color profile and is  
24 processed instead of any embedded color profile. The profile URI MUST be  
25 added as a Required Resource relationship to the FixedPage part [M2.10].

### 26 **15.2.9 Color Space Pixel Formats for Raster Images**

27 If an ICC profile is not embedded or associated with a raster image or if the  
28 embedded or associated profile is not compatible with the pixel format of the image,  
29 the default pixel formats for each color space MUST be treated as follows [M8.30].

30 *Table 15–2. Color Space Pixel Format Defaults*

Color Space	Pixel Formats
sRGB	Integer RGB
	Integer Grayscale
	Integer 3-Channel
scRGB (wcsRGB gamut)	Floating Point scRGB
	Half-Float scRGB
	Fixed-Point scRGB
	Floating Point Grayscale
	Half-Float Grayscale
	Fixed-Point Grayscale

---

CMYK (with SWOP profile)	Integer CMYK
	Integer 4-Channel
	Integer 5-Channel (ignore channel 5)
	Integer 6-Channel (ignore channels 5 and 6)
	Integer 7-Channel (ignore channels 5, 6, and 7)
	Integer 8-Channel (ignore channels 5, 6, 7, and 8)

---

### 1 15.3 Color Separation

2 Consumers MAY perform color separation, if desired [O8.5].

3 A named color used for markings that are intended to be rendered on every layer of  
4 the separation can be specified with the DocumentImpositionColor PrintTicket  
5 setting.

6 The color name specified by the DocumentImpositionColor PrintTicket setting MUST  
7 be matched only to profiles containing exactly one non-zero-length colorant name in  
8 the profile's colorantTable [M8.22]. The color name specified by the  
9 DocumentImpositionColor setting serves as a label for that color only and MUST NOT  
10 be matched against any Named Colors known by the consumer [M8.23]. The  
11 comparison of the color name specified by the DocumentImpositionColor PrintTicket  
12 setting with the colorant name in the profile's colorantTable MUST be performed as a  
13 case-sensitive ASCII comparison after trimming leading and trailing whitespace from  
14 each string [M8.24].

15 The imposition named color is used *only* to compute XYZ values for consumers that  
16 do not perform separation. For consumers that do perform separation, it is an  
17 indicator that the tint level supplied SHOULD be used for all device colorants [S8.7].  
18 Producers SHOULD create the profile used by the imposition color in such a way that  
19 it does not lay down excessive ink when printed on a device that does not perform  
20 separation and uses the profile to compute XYZ values instead [S8.8].

---

### 21 15.4 Alpha and Gradient Blending with Rich Colors

22 For consumers that understand rich colors, it is necessary to understand how they  
23 may be blended to create gradient or transparency effects. The PrintTicket specifies  
24 the color space that SHOULD be used for blending gradients and transparencies in  
25 the PageBlendColorSpace setting [S8.9]. These settings apply to the page level.

26 If a consumer understands the PageBlendColorSpace PrintTicket setting, it SHOULD  
27 convert all color to the specified blending color space before performing a blend  
28 operation [S8.9]. For gradients, the specified blending color space is used only if no  
29 gradient stop color values are specified using sRGB or scRGB colors. If any of the  
30 gradient stop color values are specified using sRGB or scRGB colors or the consumer  
31 does not understand the PageBlendColorSpace PrintTicket setting, the color  
32 interpolation mode of the gradient brush MUST be used instead [M8.25].

33 The behavior of documents using rich color features is implementation specific.  
34 Consumers MUST support sRGB [M8.1], but they MAY support rich color spaces such  
35 as scRGB or CMYK as well [O8.6]. Consumers that encounter any document using

#### 15.4 *Alpha* and Gradient Blending with Rich Colors

*Color*

- 1 non-sRGB colors MAY process those colors using the simpler sRGB color space,
- 2 resulting in deviations, especially for alpha blending [O8.6].

---

## 1 15.5 PrintTicket Color Settings

2 This section summarizes the color-related PrintTicket settings. For more information,  
3 refer to the Print Schema.

4 *Table 15-3. PrintTicket color settings*

Feature or ParameterDef	Option/ScoredProperty or Properties	Description
PageColorManagement	Device (default)	Perform color management only in device.
	Driver	Allow driver to perform color management. The driver MAY color manage elements or convert them to different color spaces [O8.7].
PageDeviceColorSpaceProfileURI	<i>profileUri properties</i>	Identifies an ICC profile contained in the XPS package. The processing of this option depends of the setting of the PageDeviceColorSpaceUsage feature. No default value is specified. Contains an absolute part name relative to the package root. All elements using that profile are assumed to be already in the appropriate device color space, and will not be color managed in the driver or device.
PageDeviceColorSpaceUsage	MatchToDefault (default)	If the device determines that the profile specified by the PageDeviceColorSpaceProfileURI feature can be used as a device color space profile, all elements using the same profile are treated as already being specified in device color space. However, the device's internal color profile SHOULD be used for color management of all other elements [S8.10].

---

		If the profile cannot be used as a device color space profile, elements using the profile MUST be color managed like any other element using a color profile [M8.27].
	OverrideDeviceDefault	If the profile specified by the PageDeviceColorSpaceProfileURI parameter definition has a number of channels matching the number of primaries of the device, it SHOULD be used instead of the device's internal color management for all elements [S8.11]. Elements using this profile are assumed to be in device color space and will not be color managed further.
PageBlendColorSpace	sRGB (default)	The sRGB color space that SHOULD be used for blending [S8.9].
	scRGB	The scRGB color space that SHOULD be used for blending [S8.9].
	ICCProfile	The Uri property of the option specifies an ICC profile defining the color space that SHOULD be used for blending [S11.13]. The Uri is an absolute part name relative to the package root.  The profile MUST be an output profile (containing AToB0Tag, BToA0Tag, AToB1Tag, BToA1Tag, AToB2Tag, and BToA2Tag), otherwise it MUST be ignored [M8.28]. The rendering intent specified by PageICMRenderingIntent PrintTicket setting is used, unless the profile specifies a rendering intent of its own.

		Elements using the profile specified by PageBlendColorSpace MAY be blended naively (channel-by-channel) without converting through PCS [O8.8].
PageICMRenderingIntent	AbsoluteColorimetric RelativeColorimetric (default) Photographs BusinessGraphics	The rendering intent as defined by the ICC Version 2 specification. This value SHOULD be ignored for elements using a profile that specifies the rendering intent in the profile [S8.13].
DocumentImpositionColor	<i>colorName properties</i>	Elements using the named color identified by the colorName properties MUST appear on all color separations [M8.29]. See §15.3, "Color Separation," on page 217 and §15.1.16, "Named Color Syntax," on page 212 for details.  Consumers that do not produce separations treat these elements like other elements using named color, without any additional required processing steps.  No default value is specified.
PageBlackGenerationProcessing	Automatic	Default.
	Custom/TotalInkCoverageLimit	The maximum allowable sum of the four ink coverages anywhere in an image or element (200% to 400%).
	Custom/BlackInkLimit	The maximum allowed K-channel value (0% to 100%).
	Custom/GrayComponentReplacementLevel	The percentage of gray component replacement to perform (0% to 100%).
	Custom/GrayComponentReplacementStart	The point in the highlight-to-shadow range where GCR should start (0% to

---

	100%; 100% = darkest shadow).
Custom/ GrayComponentReplacement Extent	The extent beyond neutrals (into chromatic colors) that GCR applies. 0% = Undercolor component replacement; 100% = Gray component replacement.
Custom/UnderColorAdditionS tart	The shadow level below which UCA will be applied (0% to 100%).
Custom/UnderColorAdditionL evel	The amount of chromatic ink (in gray component ratios) to add to areas where GCR/UCR has generated "BlackInkLimit" (or "UCAStart", if specified) in the dark neutrals and near-neutral areas (0% to 100%).

---

1

2



## 1 16. Document Structure and Interactivity

2 Some consumers support enhanced interactive functionality through features such as  
3 text selection, navigation, and hyperlinking. Others, such as screen readers, provide  
4 enhanced accessibility. These features rely on structural information beyond what  
5 can be inferred from the page markup. Producers can author this information  
6 explicitly.

7 The methods for adding document structure described here are OPTIONAL [O9.1].  
8 Consumers MAY ignore any authored document structure or hyperlinks [O9.1],  
9 particularly where they are not relevant (such as in the case of printers).  
10 Recommended consumer behavior in the absence of document structure information  
11 is also described.

12 Document structure is defined with markup in the FixedPage, FixedDocument,  
13 DocumentStructure, and StoryFragments parts.

---

### 14 16.1 Document Structure Markup

15 Document structure markup consists of two structural concepts. The first is the  
16 [document outline](#), which contains a structured list of indices into the XPS Document,  
17 similar to a table of contents. The second is the [document content](#), which identifies  
18 blocks of individually readable content. These blocks are called [stories](#).

19 A story can extend across multiple pages, and several stories can share a single  
20 page. A story can include the entire contents of an XPS Document, or it can include  
21 only an individual block of readable content, such as a single newspaper article. Like  
22 a newspaper article, the story may appear in blocks throughout the XPS Document.  
23 [Example: The first part could appear on page 1 and the second part on page 5. end  
24 example] Since a story can span multiple pages, the document content identifies  
25 which FixedPage parts contain fragments of a particular story.

26 A [story fragment](#) is the portion of a story that appears within a single fixed page.  
27 The story fragment contains the structural markup for all text and images related to  
28 a particular story on a particular page. When a producer specifies the document  
29 structure, every FixedPage part has a corresponding StoryFragments part that  
30 contains all of the story fragments for that page.

31 Each story fragment contains content structure information. [Content structure](#) is the  
32 set of markup elements that allow expression of well-understood semantic blocks,  
33 such as paragraphs, tables, lists, and figures. Content structure markup enables  
34 features such as paragraph and table selection, screen reading, and rich-format  
35 copying.

36 Producers MAY provide either the document outline or the document content, or  
37 both; consumers MAY ignore either or both [O9.2].

### 1 16.1.1 DocumentStructure Part

2 The fundamental building block of document structure markup is the named element.  
 3 A **named element** refers to an element in the fixed page markup with a specified  
 4 **Name** attribute. Every meaningful element in the fixed page markup SHOULD  
 5 specify a **Name** attribute in order for the document structure markup to refer to it  
 6 [S9.1].

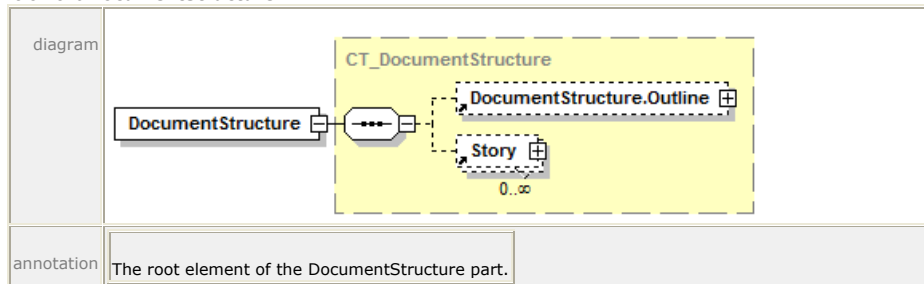
7 Document structure markup SHOULD NOT refer to a single named element more  
 8 than once in the document content or to a named element that embeds another  
 9 named element that it also refers to. When referring to a <Canvas> element,  
 10 producers SHOULD consider all descendant elements to be referenced in markup  
 11 order [S9.3]. Consumers MAY choose to interpret these scenarios as duplicate  
 12 document content [O9.3].

13 Children of <VisualBrush> elements SHOULD NOT be referenced by document  
 14 structure markup [S9.30].

15 Because each named element in a FixedPage part that is intended as an addressable  
 16 location is specified in the <PageContent.LinkTargets> element in the  
 17 FixedDocument part, consumers MAY first attempt to locate named elements directly  
 18 from the FixedDocument part [O9.4].

#### 19 16.1.1.1 <DocumentStructure> Element

20 element **DocumentStructure**



21 The <DocumentStructure> element is the root element of the DocumentStructure  
 22 part. It MAY contain a single <DocumentStructure.Outline> element and zero or  
 23 more <Story> elements [M2.72].

24 *Example 16-1. Document structure markup*

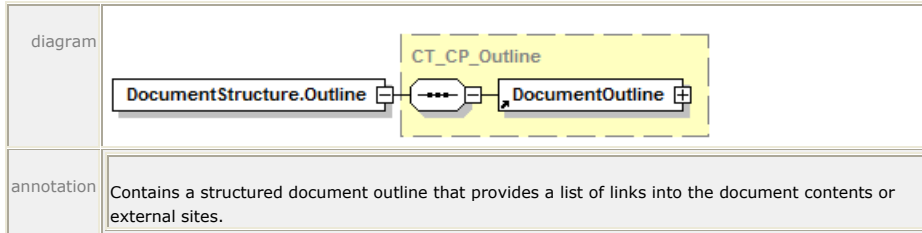
```

25 <DocumentStructure
26   xmlns="http://schemas.microsoft.com/xps/2005/06/documentstructure">
27   <DocumentStructure.Outline>
28     ...
29   </DocumentStructure.Outline>
30   <Story>
31     ...
32   </Story>
33   <Story>
34     ...
35   </Story>
36 </DocumentStructure>
  
```

1 *end example]*

1 **16.1.1.2 <DocumentStructure.Outline> Element**

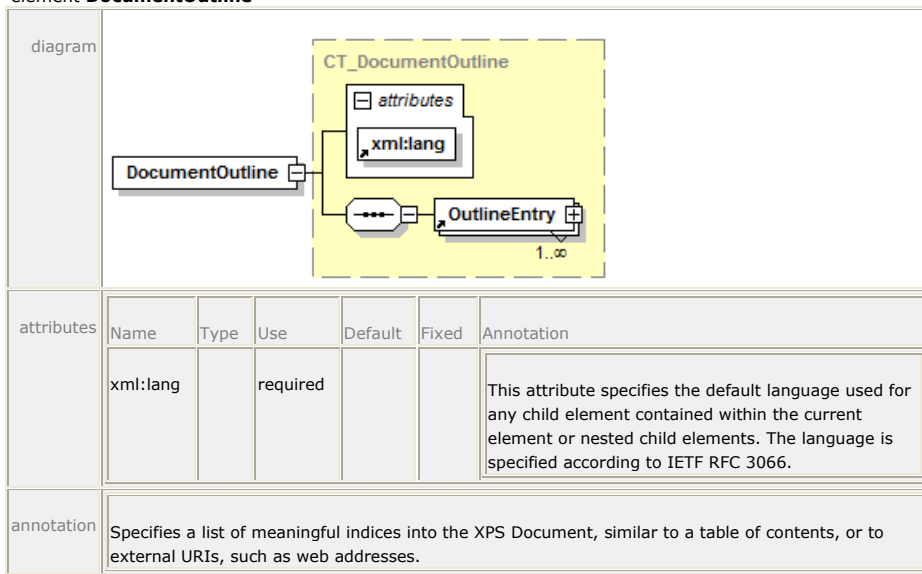
2 element **DocumentStructure.Outline**



3 The <DocumentStructure.Outline> element is the root element of the document  
 4 outline. The <DocumentStructure.Outline> element contains only a single  
 5 <DocumentOutline> element.

6 **16.1.1.3 <DocumentOutline> Element**

7 element **DocumentOutline**



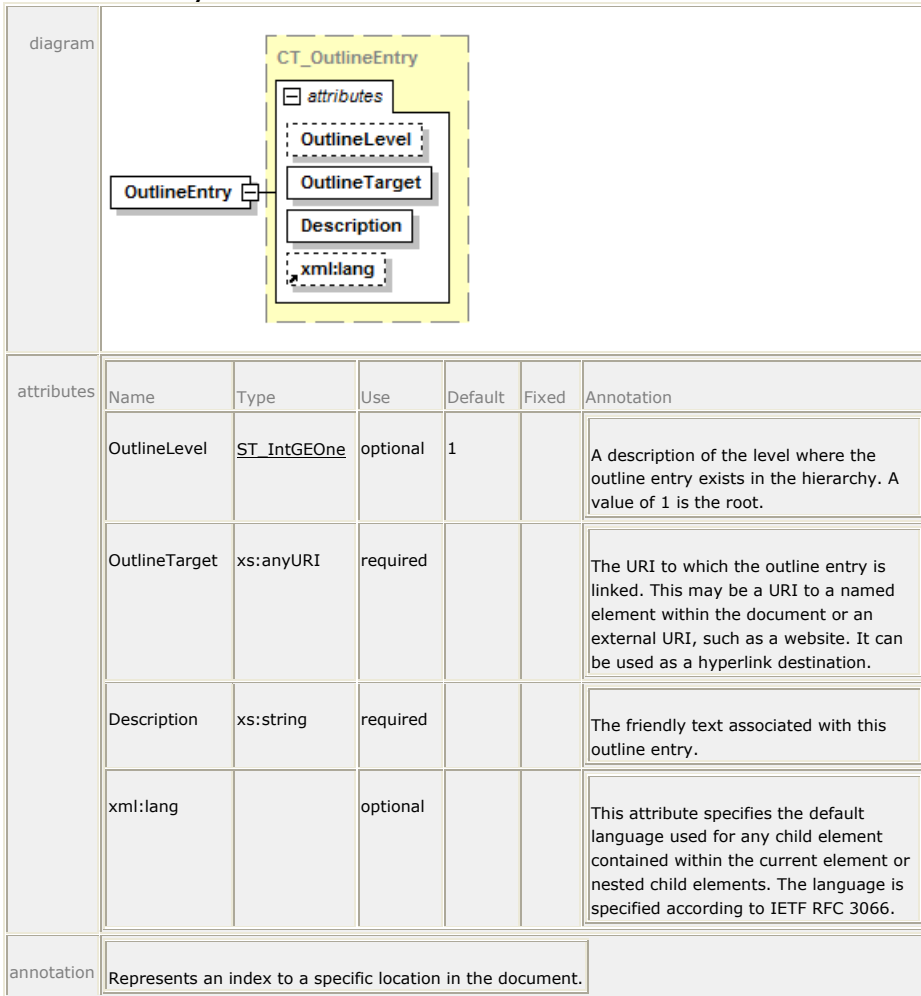
8 The <DocumentOutline> element lets producers specify an organizational hierarchy  
 9 in the form of a list of URIs to locations in the fixed page markup or to external  
 10 addresses, similar to a table of contents or a set of bookmarks. The  
 11 <DocumentOutline> element contains only <OutlineEntry> elements.

12 The **xml:lang** attribute specifies the default language used by the **Description**  
 13 attribute of the child <OutlineEntry> element.

14 Consumers can use the document outline to implement such features as a table of  
 15 contents or a navigation pane.

1 **16.1.1.4 <OutlineEntry> Element**

2 element **OutlineEntry**



3 Each <OutlineEntry> element represents an index to a specific location in the  
 4 document or a specific location external to the document. Consumers can use the  
 5 document outline information to support interactive functionality.

6 *Example 16-2. Document outline markup*

7 A viewing consumer can create a navigation pane that uses the Unicode value of the  
 8 **Description** attribute of each <OutlineEntry> element. The corresponding location  
 9 is specified by the **OutlineTarget** attribute, which are specified in a manner identical  
 10 to hyperlinks. The **OutlineLevel** attribute allows consumers to indent entries in the  
 11 navigation pane.

```

1 <DocumentStructure
2   xmlns="http://schemas.microsoft.com/xps/2005/06/documentstructure">
3   <DocumentStructure.Outline>
4     <DocumentOutline>
5       <OutlineEntry
6         OutlineLevel="1"
7         Description="1. Documents"
8         OutlineTarget="../FixedDoc.fdoc#Documents_1" />
9       <OutlineEntry
10        OutlineLevel="2"
11        Description="1.1. Paragraphs"
12        OutlineTarget="../FixedDoc.fdoc#Paragraphs_1_1" />
13     </DocumentOutline>
14   </DocumentStructure.Outline>
15 </DocumentStructure>

```

16 A consumer might display this information as follows, with the first entry linked to  
 17 Documents\_1 and the second entry linked to Paragraphs\_1\_1.

```

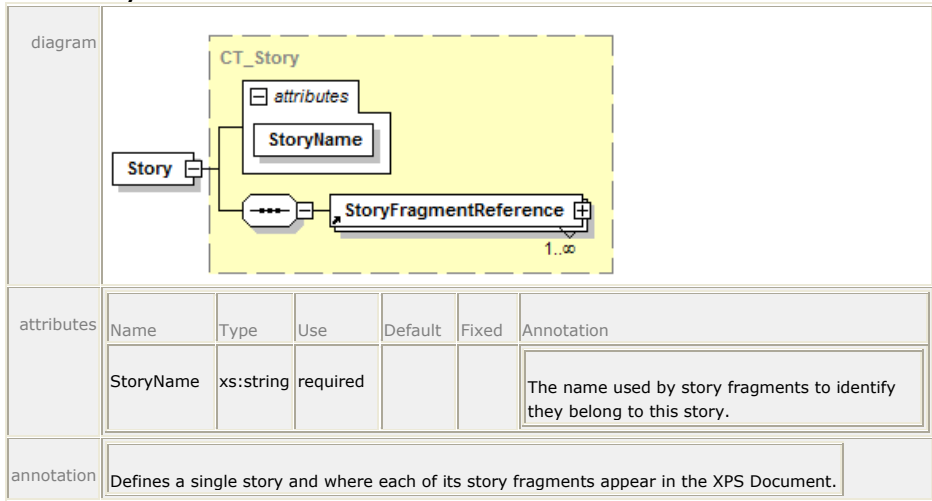
18 1. Documents
19 1.1. Paragraphs

```

20 *end example]*

21 **16.1.1.5 <Story> Element**

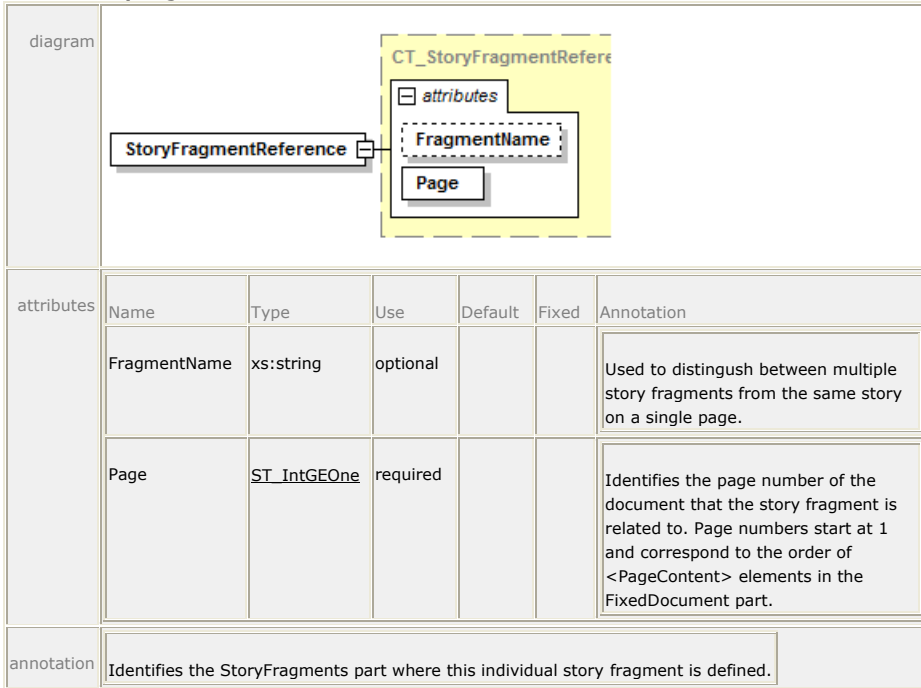
22 element **Story**



23 The <Story> element is the root for a single story and orders all of the story  
 24 fragments containing content structure information such as sections, paragraphs,  
 25 and tables. Each story has a unique name that is used to correlate the content  
 26 structure for each page to that story. The <Story> element contains one or more  
 27 <StoryFragmentReference> elements.

1 **16.1.1.6 <StoryFragmentReference> Element**

2 element **StoryFragmentReference**



3 The <StoryFragmentReference> element identifies the page with a relationship to  
 4 the StoryFragments part in which the single story fragment is defined. By identifying  
 5 where in the XPS Document each story fragment appears, consumers can easily  
 6 access only the pages that contain a particular story.

7 Each page that contains a story fragment is identified by number. This number refers  
 8 to the *n*th page of the XPS Document referenced within the fixed document sequence  
 9 and fixed document markup, starting at the fixed payload root. This value is  
 10 identified in the **Page** attribute. The StoryFragments part containing the  
 11 corresponding content structure is referenced by retrieving the part associated via  
 12 relationship from the indicated page. This allows consumers to access only the pages  
 13 of the document that contain the story of interest. It is also possible for a single  
 14 story to return to a page containing a different fragment of the same story.

15 The **FragmentName** attribute MUST be unique within the scope of the story  
 16 [M2.72].

1 *Example 16-3. Simple multi-story document*

2 The following markup describes a four-page document containing one story that  
3 covers the first one and one-half pages and then continues on page 4. It is  
4 interrupted by a second story that begins in the middle of page 2 and concludes on  
5 page 3.

```
6 <DocumentStructure
7   xmlns="http://schemas.microsoft.com/xps/2005/06/documentstructure">
8   <Story Name="Story1">
9     <StoryFragmentReference Page="1"/>
10    <StoryFragmentReference Page="2"/>
11    <StoryFragmentReference Page="4"/>
12  </Story>
13  <Story Name="Story2">
14    <StoryFragmentReference Page="2"/>
15    <StoryFragmentReference Page="3"/>
16  </Story>
17 </DocumentStructure>
```

18 *end example]*

19 *Example 16-4. Story flowing back and forth across a page boundary*

20 The following markup describes a page containing two tables, arranged side-by-side,  
21 each of which continues to the following page. In this case, the fragment is split and  
22 a fragment name is specified. `FragmentA` refers to the content leading up to the  
23 middle of the first (left) table and `FragmentB` is the continuation of this table on the  
24 following page. The flow then returns to the second (right) table on page 1  
25 (`FragmentC`) before continuing with the rest of the story in `FragmentD`.

```
26 <DocumentStructure
27   xmlns="http://schemas.microsoft.com/xps/2005/06/documentstructure">
28   <Story Name="Story_1">
29     <StoryFragmentReference FragmentName="FragmentA" Page="1"/>
30     <StoryFragmentReference FragmentName="FragmentB" Page="2"/>
31     <StoryFragmentReference FragmentName="FragmentC" Page="1"/>
32     <StoryFragmentReference FragmentName="FragmentD" Page="2"/>
33   </Story>
34 </DocumentStructure>
```

35 *end example]*

### 36 16.1.2 StoryFragments Part

37 The StoryFragments part contains content structure markup (describing such things  
38 as tables and paragraphs) for each story fragment that appears on the page. The  
39 content structure is expressed by tags that ultimately wrap `<NamedElement>`  
40 references that point to fixed page markup.

41 *Table 16-1. StoryFragments part elements*

Name	Description
<code>&lt;StoryFragments&gt;</code>	Root element.
<code>&lt;StoryFragment&gt;</code>	Contains all content structure markup elements for a single story fragment.
<code>&lt;StoryBreak&gt;</code>	Presence of this element indicates that the following or preceding markup is not continued to the previous or



---

	next story fragment, depending on whether the element is at the beginning or end of the story fragments markup.
<SectionStructure>	Arbitrary structural grouping element.
<TableStructure>	Contains a full table definition.
<TableRowGroupStructure>	Contains a group of table rows.
<TableRowStructure>	Contains a row of table cells.
<TableCellStructure>	Contains structural elements representing the contents of a table cell.
<ListStructure>	Group of related items.
<ListItemStructure>	Individual item in a list.
<FigureStructure>	Group of related named elements that should be interpreted as a whole (such as a diagram).
<ParagraphStructure>	Group of named elements that constitute a paragraph.
<NamedElement>	Element that links the document structure markup to the fixed page markup.

---

1 Because a single content structural element may be split across pages, the  
2 <StoryBreak> element is provided to identify that a given element continues *to* the  
3 next story fragment or continues *from* a previous story fragment. A <StoryBreak>  
4 element MUST NOT be included in a position other than the first or last child element  
5 of a <StoryFragment> element [M2.72].

6 If a <StoryBreak> element is not present at the beginning of the content structure  
7 markup, consumers SHOULD consider the markup a continuation of the previous  
8 story fragment that must be merged [S9.4]. Likewise, if a <StoryBreak> element is  
9 not present at the end of the content structure markup, consumers SHOULD consider  
10 the markup a continuation to the next story fragment that must be merged to  
11 determine the cross-fragment content structure [S9.4].

12 Content structure is merged on an element-by-element basis, merging the last  
13 element closed in the leading story fragment with the first element opened in the  
14 trailing story fragment. This process continues until the closing tag from the leading  
15 story fragment no longer matches the opening tag from the trailing story fragment.

16 <TableCellStructure> elements require special merging, such that all  
17 <TableCellStructure> elements within a <TableRowStructure> element are merged.  
18 In order to merge the table cells and rows correctly, producers MUST specify empty  
19 <TableCellStructure> elements for cells that do not break across story fragments  
20 [M9.1].

1 *Example 16-5. Content structure spanning pages*

2 Given the following two StoryFragments parts, consumers may construct the content  
3 structure as shown.

```
4 <!-- First StoryFragments part -->
5
6 <StoryFragments
7   xmlns="http://schemas.microsoft.com/xps/2005/06/documentstructure">
8   <StoryFragment FragmentType="Header">
9     <StoryBreak />
10    <ParagraphStructure>
11      <NamedElement NameReference="Block1" />
12    </ParagraphStructure>
13    <StoryBreak />
14  </StoryFragment>
15  <StoryFragment StoryName="Story1" FragmentType="Content">
16    <StoryBreak />
17    <SectionStructure>
18      <TableStructure>
19        <TableRowGroupStructure>
20          <TableRowStructure>
21            <TableCellStructure>
22              <ParagraphStructure>
23                <NamedElement NameReference="Block2" />
24                <NamedElement NameReference="Block3" />
25              </ParagraphStructure>
26            </TableCellStructure>
27            <TableCellStructure>
28              <ParagraphStructure>
29                <NamedElement NameReference="Block4" />
30              </ParagraphStructure>
31            </TableCellStructure>
32          </TableRowStructure>
33          <TableRowStructure>
34            <TableCellStructure>
35              <ParagraphStructure>
36                <NamedElement NameReference="Block5" />
37                <NamedElement NameReference="Block6" />
38              </ParagraphStructure>
39            </TableCellStructure>
40            <TableCellStructure>
41              <ParagraphStructure>
42                <NamedElement NameReference="Block7" />
43              </ParagraphStructure>
44            </TableCellStructure>
45          </TableRowStructure>
46        </TableRowGroupStructure>
47      </TableStructure>
48    </SectionStructure>
49  </StoryFragment>
50  <StoryFragment FragmentType="Footer">
51    <StoryBreak />
52    <ParagraphStructure>
53      <NamedElement NameReference="Block8" />
54    </ParagraphStructure>
55    <StoryBreak />
56  </StoryFragment>
57 </StoryFragments>
```

58

```

1 <!-- Second StoryFragments part -->
2
3 <StoryFragments
4   xmlns="http://schemas.microsoft.com/xps/2005/06/documentstructure">
5   <StoryFragment FragmentType="Header">
6     <StoryBreak />
7     <ParagraphStructure>
8       <NamedElement NameReference="Block9" />
9     </ParagraphStructure>
10    <StoryBreak />
11  </StoryFragment>
12  <StoryFragment StoryName="Story1" FragmentType="Content">
13    <SectionStructure>
14      <TableStructure>
15        <TableRowGroupStructure>
16          <TableRowStructure>
17            <TableCellStructure />
18            <TableCellStructure>
19              <ParagraphStructure>
20                <NamedElement NameReference="Block10" />
21                <NamedElement NameReference="Block11" />
22              </ParagraphStructure>
23            </TableCellStructure>
24          </TableRowStructure>
25          <TableRowStructure>
26            <TableCellStructure>
27              <ParagraphStructure>
28                <NamedElement NameReference="Block12" />
29              </ParagraphStructure>
30            </TableCellStructure>
31            <TableCellStructure>
32              <ParagraphStructure>
33                <NamedElement NameReference="Block13" />
34              </ParagraphStructure>
35            </TableCellStructure>
36          </TableRowStructure>
37        </TableRowGroupStructure>
38      </TableStructure>
39    </SectionStructure>
40    <StoryBreak />
41  </StoryFragment>
42  <StoryFragment FragmentType="Footer">
43    <StoryBreak />
44    <ParagraphStructure>
45      <NamedElement NameReference="Block14" />
46    </ParagraphStructure>
47    <StoryBreak />
48  </StoryFragment>
49 </StoryFragments>

```

```

50
51 <!-- Resulting merged content structure for Story1 -->
52
53 <SectionStructure>
54   <TableStructure>
55     <TableRowGroupStructure>
56       <TableRowStructure>
57         <TableCellStructure>
58           <ParagraphStructure>
59             <NamedElement NameReference="Block2" />
60             <NamedElement NameReference="Block3" />
61           </ParagraphStructure>

```

```

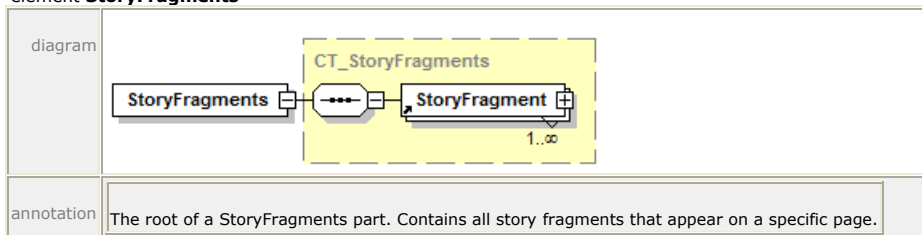
1      </TableCellStructure>
2      <TableCellStructure>
3          <ParagraphStructure>
4              <NamedElement NameReference="Block4" />
5          </ParagraphStructure>
6      </TableCellStructure>
7  </TableRowStructure>
8  <TableRowStructure>
9      <TableCellStructure>
10         <ParagraphStructure>
11             <NamedElement NameReference="Block5" />
12             <NamedElement NameReference="Block6" />
13         </ParagraphStructure>
14     </TableCellStructure>
15     <TableCellStructure>
16         <ParagraphStructure>
17             <NamedElement NameReference="Block7" />
18             <NamedElement NameReference="Block10" />
19             <NamedElement NameReference="Block11" />
20         </ParagraphStructure>
21     </TableCellStructure>
22 </TableRowStructure>
23 <TableRowStructure>
24     <TableCellStructure>
25         <ParagraphStructure>
26             <NamedElement NameReference="Block12" />
27         </ParagraphStructure>
28     </TableCellStructure>
29     <TableCellStructure>
30         <ParagraphStructure>
31             <NamedElement NameReference="Block13" />
32         </ParagraphStructure>
33     </TableCellStructure>
34 </TableRowStructure>
35 </TableRowGroupStructure>
36 </TableStructure>
37 </SectionStructure>

```

38 *end example]*

### 39 16.1.2.1 <StoryFragments> Element

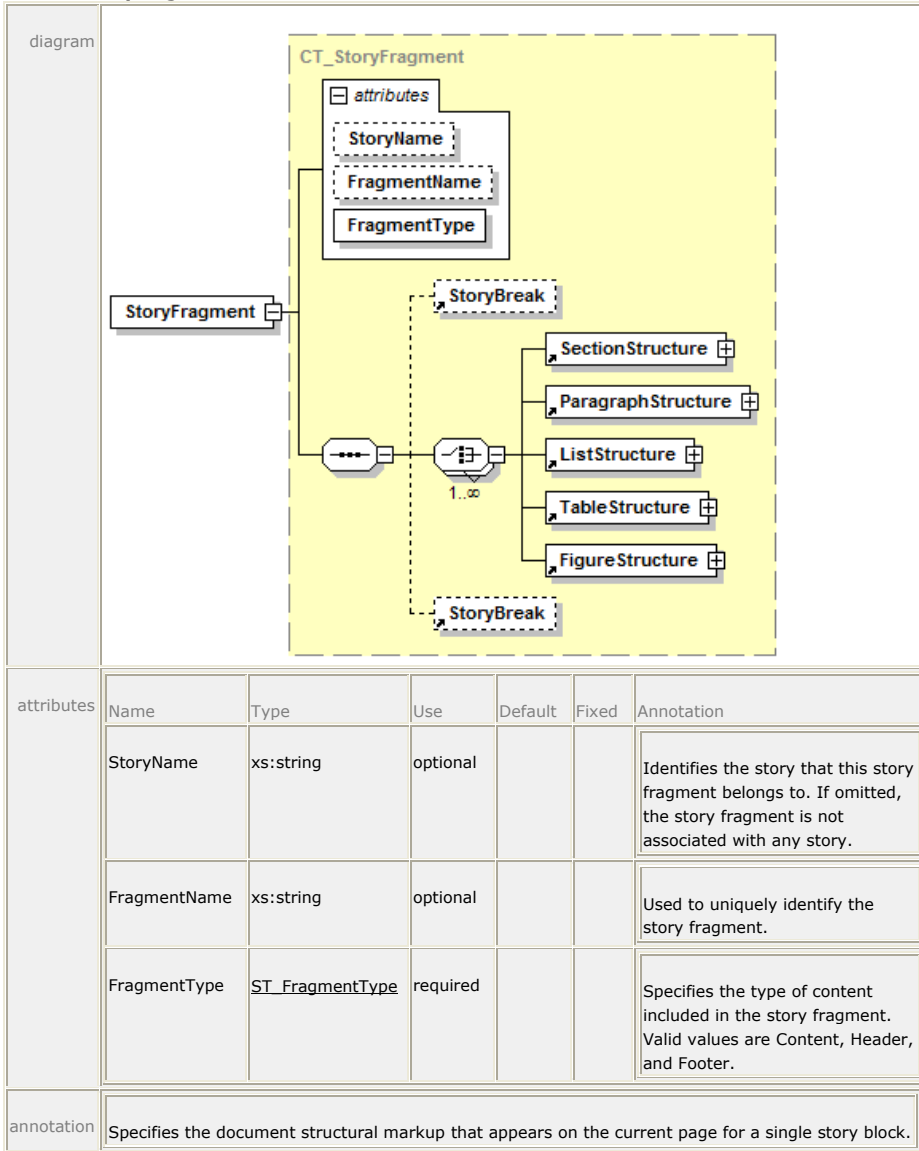
40 element **StoryFragments**



41 The <StoryFragments> element groups all of the <StoryFragment> elements on a  
42 page.

1 **16.1.2.2 <StoryFragment> Element**

2 element **StoryFragment**



3

1 Each <StoryFragment> has a **StoryName** attribute that associates it with a story  
 2 defined in the DocumentStructure part. It also has a **FragmentType** attribute, the  
 3 values for which are Content (the default), Header, or Footer.

4 Headers and footers are defined in their own story fragment on each page. These  
 5 stories do not specify a **StoryName** value, so they are essentially unreferenced  
 6 stories that exist only on a single page.

7 Producers authoring document structure information SHOULD reference every  
 8 element of the fixed page markup that has semantic meaning (such as text or  
 9 images) in the StoryFragments parts [S9.5].

10 *Example 16-6. StoryFragments part markup*

11 The following markup describes the StoryFragments part of a one-page document:

```

12 <StoryFragments
13   xmlns="http://schemas.microsoft.com/xps/2005/06/documentstructure">
14   <StoryFragment FragmentType="Header">
15     <StoryBreak />
16     <ParagraphStructure>
17       <NamedElement NameReference="Block13" />
18       <NamedElement NameReference="Block14" />
19     </ParagraphStructure>
20     <StoryBreak />
21   </StoryFragment>
22   <StoryFragment StoryName="Story1" FragmentType="Content">
23     <StoryBreak />
24     <ParagraphStructure>
25       <NamedElement NameReference="Block1" />
26       <NamedElement NameReference="Block2" />
27     </ParagraphStructure>
28     <TableStructure>
29       <TableRowGroupStructure>
30         <TableRowStructure>
31           <TableCellStructure>
32             <ParagraphStructure>
33               <NamedElement NameReference="Block3" />
34               <NamedElement NameReference="Block4" />
35             </ParagraphStructure>
36           </TableCellStructure>
37           <TableCellStructure>
38             <ParagraphStructure>
39               <NamedElement NameReference="Block5" />
40             </ParagraphStructure>
41           </TableCellStructure>
42         </TableRowStructure>
43       </TableRowGroupStructure>
44     </TableStructure>
45     <SectionStructure>
46       <ParagraphStructure>
47         <NamedElement NameReference="Block6" />
48       </ParagraphStructure>
49       <ParagraphStructure>
50         <NamedElement NameReference="Block7" />
51         <NamedElement NameReference="Block8" />
52       </ParagraphStructure>
53     </SectionStructure>
54   </StoryFragment>
55   <FigureStructure>
56     <NamedElement NameReference="Block9" />
57   </FigureStructure>

```

```

1      <ListStructure>
2          <ListItemStructure>
3              <ParagraphStructure>
4                  <NamedElement NameReference="Block10" />
5              </ParagraphStructure>
6          </ListItemStructure>
7          <ListItemStructure>
8              <ParagraphStructure>
9                  <NamedElement NameReference="Block11" />
10             </ParagraphStructure>
11         </ListItemStructure>
12         <ListItemStructure>
13             <ParagraphStructure>
14                 <NamedElement NameReference="Block12" />
15             </ParagraphStructure>
16         </ListItemStructure>
17     </ListStructure>
18 </SectionStructure>
19 <StoryBreak />
20 </StoryFragment>
21 <StoryFragment FragmentType="Footer">
22     <StoryBreak />
23     <ParagraphStructure>
24         <NamedElement NameReference="Block15" />
25         <NamedElement NameReference="Block16" />
26         <NamedElement NameReference="Block17" />
27     </ParagraphStructure>
28     <StoryBreak />
29 </StoryFragment>
30 </StoryFragments>

```

31 *end example]*

32 A <StoryFragment> element MAY be identified with a FragmentName attribute to  
33 distinguish it from other fragments for the same story on a single page [M2.72].

34 *Example 16-7. Story fragments markup using a fragment name*

```

35 <StoryFragments
36     xmlns="http://schemas.microsoft.com/xps/2005/06/documentstructure">
37     <StoryFragment
38         StoryName="Story1"
39         FragmentName="Fr1"
40         FragmentType="Content">
41         <StoryBreak />
42         <ParagraphStructure>
43             <NamedElement NameReference="Block1" />
44             <NamedElement NameReference="Block2" />
45         </ParagraphStructure>
46         <StoryBreak />
47     </StoryFragment>
48     <StoryFragment
49         StoryName="Story1"
50         FragmentName="Fr2"
51         FragmentType="Content">
52         <StoryBreak />
53         <ParagraphStructure>
54             <NamedElement NameReference="Block8" />
55         </ParagraphStructure>
56         <StoryBreak />
57     </StoryFragment>
58 </StoryFragments>

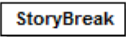
```

1 *end example]*



1 **16.1.2.3 <StoryBreak> Element**

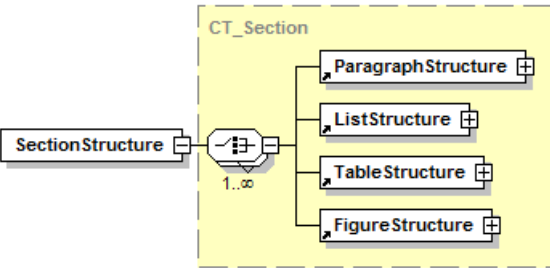
2 element **StoryBreak**

diagram	
annotation	If located at the beginning of a <StoryFragment> definition, indicates that the following markup elements should not be merged with the markup from the previous <StoryFragment>. If located at the end of a <StoryFragment> definition, indicates that the preceding markup elements should not be merged with the subsequent <StoryFragment>.

3 The <StoryBreak> element signals to the consumer not to perform merging across  
4 story fragments to determine the content structure.

5 **16.1.2.4 <SectionStructure> Element**

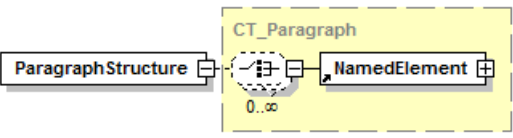
6 element **SectionStructure**

diagram	
annotation	Provides an arbitrary grouping of content structural markup elements.

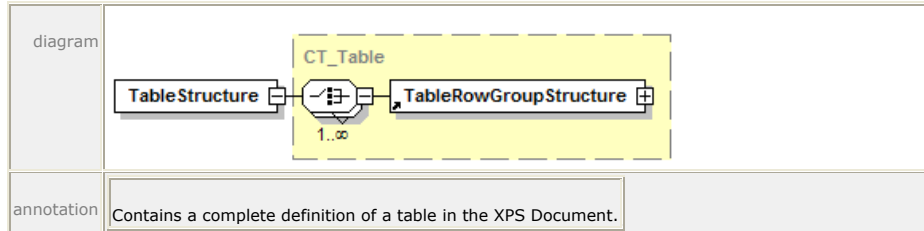
7 The <SectionStructure> element provides an arbitrary grouping of <Paragraph>,  
8 <TableStructure>, <ListStructure>, and <FigureStructure> elements.

9 **16.1.2.5 <ParagraphStructure> Element**

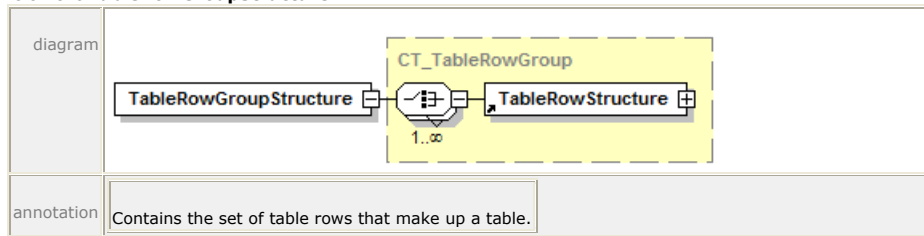
10 element **ParagraphStructure**

diagram	
annotation	Contains the named elements that constitute a single paragraph.

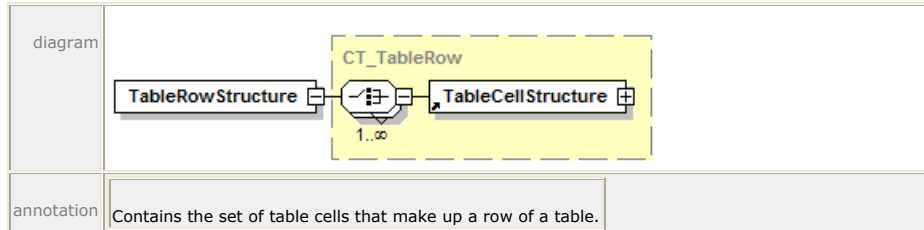
11 A <ParagraphStructure> element describes the list of <NamedElement> elements  
12 that constitute a single paragraph.

1 **16.1.2.6 <TableStructure> Element**2 element **TableStructure**

3 A <TableStructure> element is the complete definition of a table. An implementation  
 4 MAY use it to build special functionality, such as row or column selection [O9.5].

5 **16.1.2.7 <TableRowGroupStructure> Element**6 element **TableRowGroupStructure**

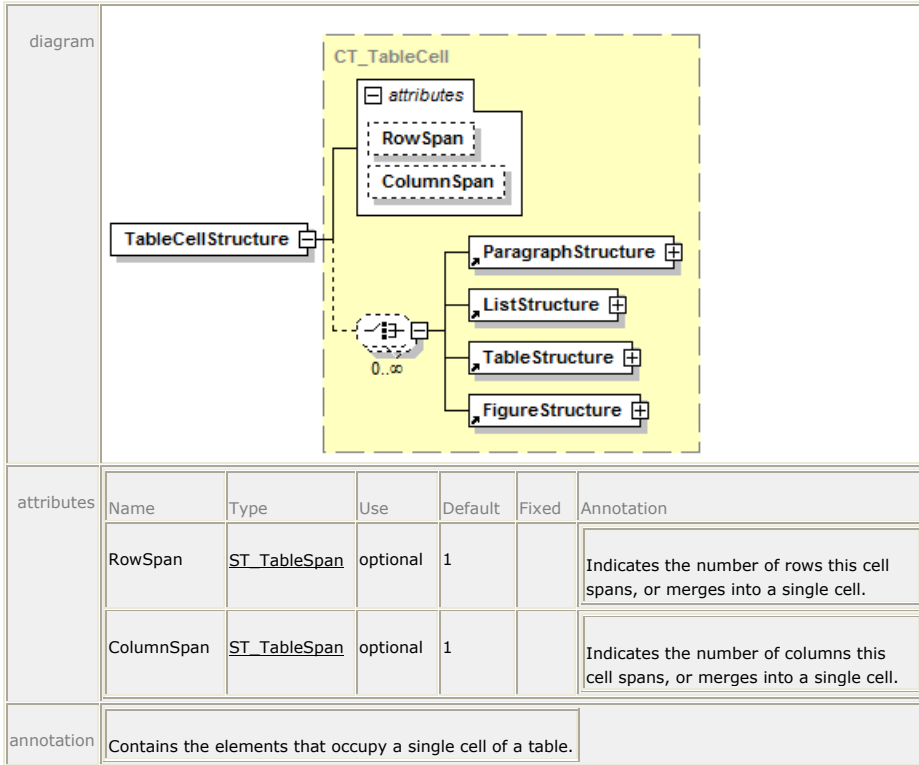
7 A <TableRowGroupStructure> element is REQUIRED in order to specify a set of  
 8 <TableRowStructure> elements [M2.72].

9 **16.1.2.8 <TableRowStructure> Element**10 element **TableRowStructure**

11 This element groups <TableCellStructure> child elements that define a single row of  
 12 a table.

1 **16.1.2.9 <TableCellStructure> Element**

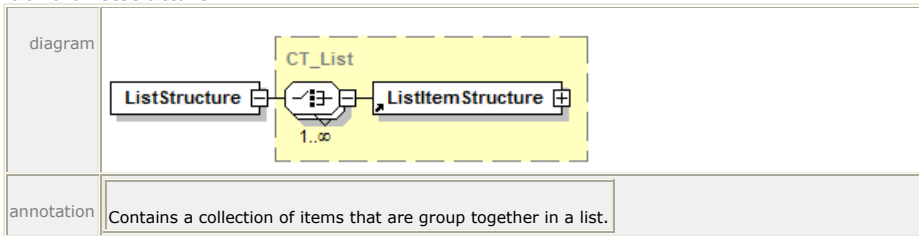
2 element **TableCellStructure**



3 This element defines the appearance of a table cell. It MAY contain nested  
 4 `<TableStructure>` elements [M2.72].

5 **16.1.2.10 <ListStructure> Element**

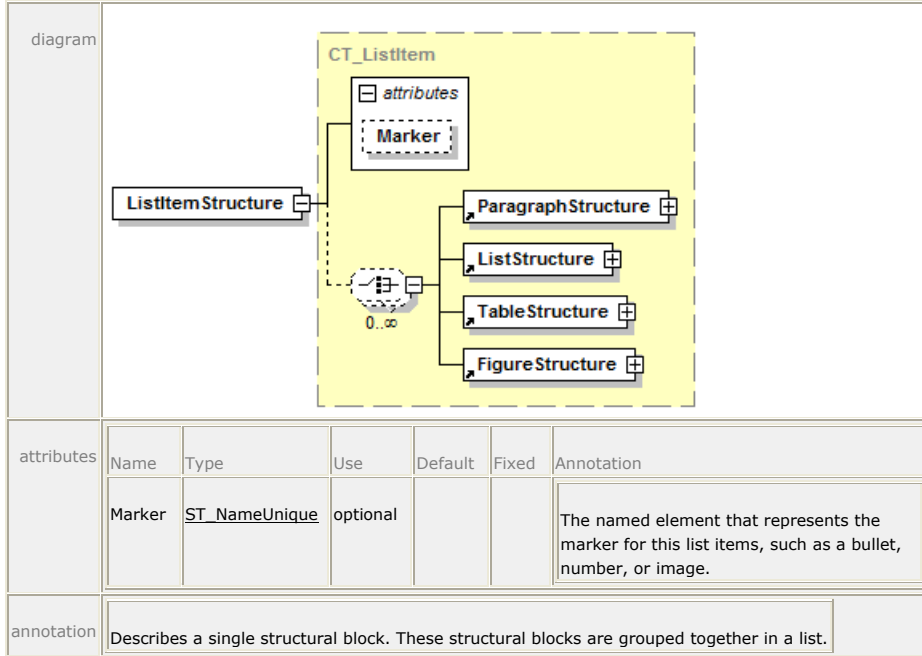
6 element **ListStructure**



7 The `<ListStructure>` element is the complete definition of a list of related items.

1 **16.1.2.11 <ListItemStructure> Element**

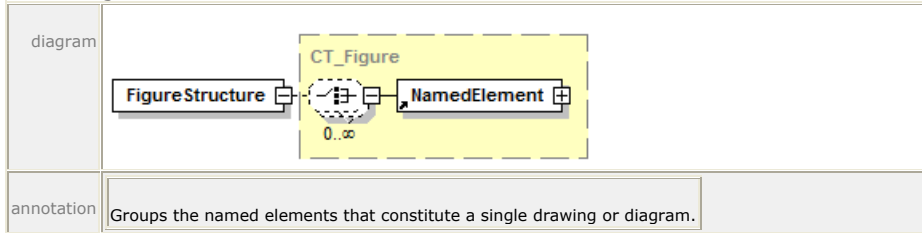
2 element **ListItemStructure**



3 A <ListItemStructure> element defines a single item in a list.

4 **16.1.2.12 <FigureStructure> Element**

5 element **FigureStructure**



6 A <FigureStructure> element includes a group of named elements that comprise a  
 7 single drawing or diagram.

1 **16.1.2.13 <NamedElement> Element**2 element **NamedElement**

diagram						
attributes	Name	Type	Use	Default	Fixed	Annotation
	NameReference	ST_Name	required			Identifies the named element in the FixedPage part markup that is referenced by the document structure markup.
annotation	All document structure is related to the fixed page markup using this element. The <NamedElement> points to a single markup element contained in the fixed page markup.					

3 A <NamedElement> references a specific element in the fixed page by using the  
 4 **NameReference** attribute to specify an element in the fixed page markup with a  
 5 corresponding name.

6 If the targeted fixed page uses markup compatibility markup that changes the  
 7 presence of certain named elements, the StoryFragments part should also use it in  
 8 order to reference each element in either representation.

9 **16.2 Hyperlinks**

10 If consumers enable user interactivity, they SHOULD support hyperlink activation  
 11 and addressing [S9.6].

12 **16.2.1 Hyperlink Activation**

13 Hyperlinks are specified inline on any <Canvas>, <Path>, or <Glyphs> element by  
 14 means of the **FixedPage.NavigateUri** attribute. The value of the attribute is the  
 15 destination URI. If hyperlinked <Path> or <Glyphs> elements are rendered as  
 16 overlapping on the page, consumers MUST treat the topmost element as the only  
 17 hyperlink that may be activated in the overlapping region [M9.2].

18 When activating a hyperlink, consumers SHOULD load the specified resource if they  
 19 understand the URI type. If the URI is an internal reference to the XPS Document,  
 20 consumers SHOULD navigate to the URI [S9.7].

21 If a producer specifies a **FixedPage.NavigateUri** attribute on a <Canvas> element,  
 22 consumers MUST treat all child elements of that canvas as having an associated  
 23 hyperlink [M9.3]. Child or descendant elements can override this value with their  
 24 own **FixedPage.NavigateUri** attribute.

- 1 Relative internal hyperlinks between FixedPage parts MUST specify, at a minimum,  
2 the named address relative to the FixedDocument part [M9.4].
- 3 Producers can mark any <FixedPage>, <Canvas>, <Path>, or <Glyphs> element as  
4 an addressable location within the XPS Document by specifying a value for the  
5 **Name** attribute. The name SHOULD be unique within the scope of the fixed  
6 document [S9.8]. If it is not unique, only the first occurrence of the named address  
7 is addressable.
- 8 These elements, if specified as a <VisualBrush.Visual> property element are not  
9 addressable by a hyperlink.
- 10 It is RECOMMENDED that **Name** attribute values be unique within an entire fixed  
11 document sequence [S9.9]. If they are not, only the first occurrence of the named  
12 address is addressable from an external location. Internal hyperlinks can specify a  
13 named element fragment relative to a particular fixed document, but consumers MAY  
14 interpret such a URI relative to the entire fixed document sequence instead [O9.6].
- 15 In order to be addressable by either a hyperlink or the document outline, the named  
16 address MUST appear in the <PageContent.LinkTargets> element in the fixed  
17 document [M9.5]. If a named address appears in the <PageContent.LinkTargets>  
18 element in the fixed document but is not found in the **Name** attribute of an element  
19 within the associated fixed page, consumers MUST treat the top of the associated  
20 fixed page as the named address [M9.6]. If the named address in a URI fragment is  
21 not found, consumers MUST ignore the fragment portion of the URI [M9.7].

22 *Example 16-8. A relative, internal, named-address hyperlink*

```
23 FixedPage.NavigateUri="../MyDocument.fdoc#MyAddress"
```

24 *end example]*

### 25 16.2.2 Hyperlink Addressing

26 XPS Documents specify two forms of URI fragment identifiers to address locations  
27 within an XPS Document. The first is a named address. [*Example:*  
28 "http://xps/MyPackage#MyAddress", where "http://xps/MyPackage" is an XPS  
29 Document and "MyAddress" is a named address within the document. *end example]*  
30 The second is an absolute page number within the XPS Document. [*Example:*  
31 "http://xps/MyPackage#15", where "15" references the FixedPage part associated  
32 with the fifteenth <PageContent> entry among all the fixed documents in the fixed  
33 document sequence. *end example]*

34 Page number fragment identifiers refer to the absolute page number (1-based) in the  
35 fixed document sequence. [*Example:* If an XPS Document has a 3-page fixed  
36 document, followed by a 10-page fixed document, followed by an 8-page fixed  
37 document, the fragment identifier "#15" refers to the second page of the third fixed  
38 document in the fixed document sequence. *end example]* Internal references MUST  
39 specify a page address relative to the fixed document sequence [M9.8].

40 *Example 16-9. A relative internal page address hyperlink*

```
41 FixedPage.NavigateUri="../../MyDocSeq.fdseq#12"
```

42 *end example]*

### 1 **16.2.3 Name Attribute**

2 The **Name** attribute contains a string value that identifies the current element as a  
 3 named, addressable point for the purpose of hyperlinking. The **Name** attribute is  
 4 optional. Names SHOULD be unique within a fixed document [S9.8], and it is  
 5 RECOMMENDED that they be unique within a fixed document sequence [S9.9]. The  
 6 **Name** attribute MUST NOT be specified on any children of a <ResourceDictionary>  
 7 element [M9.10].

8 If the **Name** attribute is specified, producers SHOULD also create a corresponding  
 9 <LinkTarget> element in the FixedDocument part within the <PageContent>  
 10 element that links to the parent fixed page [S9.10]. Consumers MAY ignore this  
 11 attribute [O9.7], but devices that support user interaction with the contents of XPS  
 12 Documents SHOULD support hyperlinks [S9.6].

13 The **Name** value, if specified, MUST meet the following requirements [M2.72]:

- 14 1. The initial character MUST be an underscore character or a letter, that is, it  
 15 falls within the Lu, Ll, Lo, Lt, and Nl categories [M2.72].
- 16 2. Trailing characters MUST be an underscore character or a letter or number,  
 17 that is, they fall within the Lu, Ll, Lo, Lt, Nl, Mn, Mc, and Nd categories  
 18 [M2.72].

19 [*Note*: These requirements match those of XML identifiers with additional  
 20 restrictions. *end note*]

21 The category abbreviations, as defined within the Unicode Character Database, are  
 22 partially reproduced in Table 16–2.

23 *Table 16–2. Unicode character categories*

Abbreviation	Description
Lu	Letter, uppercase
Ll	Letter, lowercase
Lt	Letter, titlecase
Lo	Letter, other
Mn	Mark, non-spacing
Mc	Mark, spacing combining
Nd	Number, decimal
Nl	Number, letter

### 24 **16.2.4 FixedPage.NavigateUri Attribute**

25 The **FixedPage.NavigateUri** attribute associates a hyperlink URI with an element,  
 26 making it a hyperlink source. Its value may be a relative or absolute URI that  
 27 addresses a resource that is internal or external to the XPS Document package,  
 28 respectively. The base URI used to resolve a relative URI is that of the FixedPage  
 29 part in which the element with the **FixedPage.NavigateUri** attribute appears.  
 30 Therefore, a hyperlink to a destination within the fixed document of the source MUST  
 31 specify the destination in the context of the FixedDocument part [M9.4]. [*Example*:  
 32 “../FixedDoc1.fdoc#MyDestination”. *end example*] A destination in the same fixed  
 33 document SHOULD be expressed as a relative URI [S9.11].

1 The **FixedPage.NavigateUri** attribute is OPTIONAL [M2.72]. It SHOULD be included  
2 *only* if the element is intended to be a hyperlink. Consumers MAY ignore this  
3 attribute [O9.8], but devices that support user interaction with the contents of XPS  
4 Documents SHOULD support hyperlinks [S9.6].

---

## 5 **16.3 Selection**

6 Viewing consumers that support interactivity MAY support selection and copying  
7 [O9.9]. If selection is supported, consumers SHOULD provide a visual cue over or  
8 around selected elements [S9.12]. Selection order within an XPS Document SHOULD  
9 follow reading order [S9.13].

10 Consumers MAY use the **FragmentType** attribute of the <StoryFragment> element  
11 to determine selection behavior, such as disallowing selection of both the page  
12 header and the page contents while allowing independent selection within those  
13 stories [O9.10].

---

## 14 **16.4 Accessibility**

15 Accessibility refers to features that are important to provide equal access to XPS  
16 Documents for users of all abilities. One common example of an accessibility  
17 application is a screen reader, which reads the contents of a document aloud for  
18 vision-impaired individuals.

### 19 **16.4.1 Reading Order**

20 In the absence of document structure information provided in the XPS Document,  
21 consumers MAY infer the reading order from the position of elements on the page  
22 [O9.11], but SHOULD, at minimum, rely on the markup order to determine reading  
23 order [S9.14]. Producers SHOULD order the markup in FixedPage parts to reflect the  
24 order in which it is intended to be read [S9.15]. When document structure  
25 information is present, consumers SHOULD rely on the order of appearance of  
26 named elements in the content structure markup to determine reading order  
27 [S9.16].

28 The RECOMMENDED reading order of a page-centric application is as follows [S9.17]:

- 29 • Order the content by page.
- 30 • Within a page, order by story fragment in the order the <StoryFragment>  
31 elements are specified in the StoryFragments part for that page. Producers  
32 SHOULD order <StoryFragment> elements in their intended reading order  
33 [S9.18].
- 34 • Within a <StoryFragment> element, order by <NamedElement> reference.
- 35 • Append all un-referenced elements that appear in the fixed page markup,  
36 ordered by markup order.

37 Although producers SHOULD reference every element of the fixed page markup in  
38 the content structure markup [S9.10], consumers MUST expose every element of the  
39 fixed page markup to an accessibility interface in the determined reading order, even  
40 if the elements are not referenced in the content structure markup [M9.9].



1 Consumers MAY use the **FragmentType** attribute of the <StoryFragment> element  
2 to determine reading order by interpreting elements that have **FragmentType**  
3 values of Header and Footer as belonging first or last in the reading order,  
4 respectively [O9.12].

5 The RECOMMENDED reading order of a story-centric application is as follows  
6 [S9.19]:

- 7 • Order content by story in the sequence the <Story> elements appear in the  
8 DocumentStructure part. Producers SHOULD order <Story> elements in their  
9 intended reading order [S9.20].
- 10 • Within a story, order <StoryFragmentReference> elements in the sequence  
11 they appear in the DocumentStructure part. Producers SHOULD order  
12 <StoryFragmentReference> elements in their intended reading order [S9.21].
- 13 • Within a story fragment, order by <NamedElement> references in the  
14 StoryFragments part markup.
- 15 • Append all un-referenced elements that appear in the fixed page markup,  
16 ordered by page number, then markup order.

#### 17 **16.4.2 Screen Reader Applications**

18 Screen reader applications read the contents of the document aloud. A screen reader  
19 consumer SHOULD read the document according to its reading order [S9.22]. The  
20 application SHOULD use the **UnicodeString** attribute of each <Glyphs> element  
21 [S9.23]. In addition, screen readers MAY inspect the **Indices** attribute to resolve  
22 potential ambiguities [O9.13].

23 If the screen reader provides features to navigate the document by structural  
24 elements, such as paragraphs or table rows, it SHOULD use any document structure  
25 information included in the XPS Document [S9.24].

26 If the screen reader provides features to describe images, it SHOULD read the text  
27 provided in the **AutomationProperties.Name** and  
28 **AutomationProperties.HelpText** attributes [S9.25].

29 If the screen reader provides features to describe hyperlink addresses, it SHOULD  
30 read the text provided in the **FixedPage.NavigateUri** attribute [S9.26].

#### 31 **16.4.3 Text Alternatives for Graphics and Images**

32 Images and graphics SHOULD specify text alternatives for images and graphics to  
33 make this content accessible to vision-impaired individuals [S9.27]. There are short  
34 and long textual descriptions, specified in the **AutomationProperties.Name** and  
35 **AutomationProperties.HelpText** attributes of <Path> and <Canvas>,  
36 respectively.

37 The **AutomationProperties.Name** attribute SHOULD contain a short description of  
38 the basic contents of the image or vector graphic [S9.27]. [*Example*: "A sitting dog."  
39 *end example*]The **AutomationProperties.HelpText** attribute can contain a more  
40 detailed description of the image or graphic. [*Example*: "A cocker spaniel with brown  
41 eyes, golden fur, and its tongue hanging out. It is sitting on a beanbag directly facing  
42 the camera." *end example*]

- 1 An image SHOULD specify the **AutomationProperties.Name** and  
2 **AutomationProperties.HelpText** attributes on the <Path> element that is filled  
3 with an <ImageBrush> [S9.28]. These attributes describe the content specified by  
4 the ImageSource attribute of the <ImageBrush> element.
- 5 A vector graphic (a collection of one or more <Path> elements representing a single  
6 drawing) SHOULD specify the **AutomationProperties.Name** and  
7 **AutomationProperties.HelpText** attributes only once, directly on a <Canvas>  
8 element wrapping the <Path> elements comprising the graphic [S9.29].
- 9 Individual <Path> elements that do not provide any semantic meaning (such as a  
10 line between sections or outlining a table) SHOULD NOT specify these text  
11 alternative attributes [S9.27].
- 12

## 1 17. XPS Document Package Features

2 The XPS Document format extends package-level interleaving and digital signatures  
3 as described in the OPC specification.

---

### 4 17.1 Interleaving Optimizations

5 Interleaving concerns the physical organization of XPS Documents, rather than their  
6 logical structure. It allows consumers to linearly process the bytes that make up a  
7 physical package from start to finish, without regard for context. In other words,  
8 consumers can make correct determinations about the types of logical parts and the  
9 presence of relationships on a logical part when consuming packages in a linear  
10 fashion. Consumers are never required to return to previously encountered parts and  
11 revise their determination of the content type or presence of relationships.

12 Interleaving is OPTIONAL [O10.1]. However, if the XPS Document is interleaved,  
13 these rules SHOULD be followed:

- 14 • The Content Types stream SHOULD be interleaved according to the  
15 recommendations in the OPC specification [S10.1].
- 16 • PrintTicket parts SHOULD be written to the package before the part to which  
17 they are attached [S10.2].
- 18 • The portion of the relationship data attaching the PrintTicket to a part SHOULD  
19 be written to the package before the part to which it is attached or in close  
20 proximity to the part to which it is attached [S10.3].
- 21 • If no PrintTicket settings are specified for a FixedDocumentSequence,  
22 FixedDocument, or FixedPage part, an empty PrintTicket part SHOULD be  
23 attached to the part, and the portion of the relationship data attaching the  
24 empty PrintTicket SHOULD be written to the package before the part to which  
25 it is attached or in close proximity to the part to which it is attached [S10.4].
- 26 • The last piece of the Relationships part for a FixedPage part SHOULD be  
27 written to the package in close proximity to the first piece of the FixedPage  
28 part [S10.5].

29 Following these recommendations allows more efficient processing by certain  
30 consumers. Not following these recommendations could result in less efficient  
31 processing by most consumers because they will need to wait until all parts required  
32 to process a part (attached PrintTicket, required resources) have been consumed.  
33 However, consumers MUST be prepared to correctly process packages in which the  
34 PrintTicket or the portion of the relationship data attaching the PrintTicket appears in  
35 the package after the affected part [M10.1].

36 Consumers can choose to parse an XPS Document in a head-first or tail-first manner.  
37 Tail-first parsing reveals certain package errors earlier, such as inconsistencies  
38 between the ZIP central directory and local file headers. Head-first XPS Document  
39 consumers SHOULD attempt to detect inconsistent packages as soon as possible and  
40 SHOULD generate an error message, even if they have already processed the pages  
41 that resulted in the error [S10.18]. Head-first consumers that discard parts would

- 1 need to retain the name and length of any discarded part to comply with this
- 2 recommendation.

### 1 **17.1.1 Empty PrintTicket**

2 An empty PrintTicket has the following form:

```
3 <psf:PrintTicket
4 xmlns:psf="http://schemas.microsoft.com/windows/2003/08/printing/printsch
5 emaframework" version="1"/>
```

6 It is RECOMMENDED that one empty PrintTicket be shared for all parts that attach an  
7 empty PrintTicket [S10.6].

### 8 **17.1.2 Optimizing Interleaving Order**

9 Producers MAY optimize the interleaving order of parts to help consumers avoid stalls  
10 during read-time streaming, and to allow consumers to manage their memory  
11 resources more efficiently [O10.2].

12 The optimization strategy is suggested by the consumer architecture. Therefore,  
13 interleaving optimization is typically implemented by a software component such as  
14 a driver or filter that is specific to (or aware of) the consumer architecture.

#### 15 **17.1.2.1 Single-Threaded Parsing Architectures**

16 An optimal interleaving scheme for consumers with a single-threaded parsing model  
17 interleaves parts so that each part that is required to consume a single page  
18 (FixedPage, images, and fonts) is contained in the package in its entirety, prior to  
19 the FixedPage part being referenced from the FixedDocument part's markup.

20 Single-threaded parsing architectures typically require more run-time memory  
21 resources than multi-threaded parsing architectures because the context in which a  
22 resource is used is unknown at the time the resource is received. This requires  
23 deferred processing and additional buffering.

24 [Note: When interleaving entities containing XML markup, such as the DiscardControl  
25 part, the Content Types stream, and the FixedDocument part, there is no guarantee  
26 that XML element boundaries will align with piece boundaries in the physical  
27 package. This adds a complexity to single-threaded parsing architectures: the parser  
28 must be pre-emptable. Certain existing XML parser implementations may require a  
29 pre-tokenization step. *end note*]

30 *Example 17-1. Optimized interleaving for a single-threaded parsing architecture*

31 The following markup describes a sequence of two fixed documents, the first having  
32 two FixedPage parts and the second having one FixedPage part:

33

Part/Piece	Markup
Font1.ttf	...binary font data...
Other resources	...resource data...
Page1	<FixedPage xmlns="http://schemas.microsoft.com /xps/2005/06" ...> <Glyphs FontURI="Font1.ttf"/> </FixedPage>
Page1.rels	<Relationships xmlns=

	<pre>"http://schemas.openxmlformats.org/package/2006/re relationships"&gt; &lt;Relationship Type= "http://schemas.microsoft.com/xps/2005/06 /required-resource" Target="Font1.ttf"/&gt; &lt;/Relationships&gt;</pre>
FixedDocument1/[0].piece	<pre>&lt;FixedDocument xmlns= "http://schemas.microsoft.com/xps/2005/06"&gt; &lt;PageContent Source="Page1"/&gt;</pre>
Sequence1/[0].piece	<pre>&lt;FixedDocumentSequence xmlns= "http://schemas.microsoft.com/xps/2005/06"&gt; &lt;DocumentReference Source="FixedDocument1"/&gt;</pre>
_rels/.rels/[0].piece	<pre>&lt;Relationships xmlns= "http://schemas.openxmlformats.org/package/2006/re relationships"&gt; &lt;Relationship Type="StartPart" Target="Sequence1"/&gt;</pre>
Page2	<pre>&lt;FixedPage xmlns= "http://schemas.microsoft.com/xps/2005/06" ...&gt;...&lt;/FixedPage&gt;</pre>
FixedDocument1/[1].last.piece	<pre>&lt;PageContent Source="Page2"/&gt; &lt;/FixedDocument&gt;</pre>
Page3	<pre>&lt;FixedPage xmlns= "http://schemas.microsoft.com/xps/2005/06" ...&gt;...&lt;/FixedPage&gt;</pre>
FixedDocument2	<pre>&lt;FixedDocument xmlns= "http://schemas.microsoft.com/xps/2005/06"&gt; &lt;PageContent Source="Page3"/&gt; &lt;/FixedDocument&gt;</pre>
Sequence1/[1].last.piece	<pre>&lt;DocumentReference Source="FixedDocument2" /&gt; &lt;/FixedDocumentSequence&gt;</pre>
_rels/.rels/[1].last.piece	<pre>&lt;/Relationships&gt;</pre>

1 *end example]*

#### 2 **17.1.2.2 Multi-Threaded Parsing Architectures**

3 An optimal interleaving scheme for consumers with a multi-threaded parsing model  
4 interleaves parts so that each resource part that is required to consume a single  
5 page (images and fonts) is contained in the package after the FixedPage part  
6 referencing it.

7 Multi-threaded parsing architectures typically require less run-time memory  
8 resources than single-threaded parsing architectures because the context in which  
9 resources appear is fully determined and, therefore, resources can be processed  
10 immediately.

11 *[Note: When interleaving entities containing XML markup, such as the DiscardControl*  
12 *part, the content type stream, and the FixedDocument part, there is no guarantee*  
13 *that XML element boundaries will align with piece boundaries in the physical*

- 1 package. A multi-threaded parsing architecture is naturally suited to address this
- 2 problem. *end note*]

1 *Example 17-2. Optimized interleaving for a multi-threaded parsing architecture*

2 The following markup describes a sequence of two FixedDocument parts, the first  
3 having two FixedPage parts and the second having one FixedPage part:

4

Part/Piece	Markup
_rels/.rels/[0].piece	<pre>&lt;Relationships xmlns="http://schemas.microsoft.com/package /2005/06/relationships"&gt; &lt;Relationship Type="StartPart" Target= "Sequence1"/&gt; &lt;/Relationships&gt;</pre>
Sequence1/[0].piece	<pre>&lt;FixedDocumentSequence xmlns= "http://schemas.microsoft.com/xps/2005/06"&gt; &lt;DocumentReference Source="FixedDocument1"/&gt; &lt;/FixedDocumentSequence&gt;</pre>
FixedDocument1/[0].piece	<pre>&lt;FixedDocument xmlns= "http://schemas.microsoft.com/xps/2005/06"&gt; &lt;PageContent Source="Page1"/&gt; &lt;/FixedDocument&gt;</pre>
Page1.rels	<pre>&lt;Relationships xmlns= "http://schemas.openxmlformats.org/package/2006/re lationships"&gt; &lt;Relationship Type= "http://schemas.microsoft.com/xps/2005/06 /required-resource" Target="Font1.ttf"/&gt; &lt;/Relationships&gt;</pre>
Page1	<pre>&lt;FixedPage xmlns="http://schemas.microsoft.com /xps/2005/06" ...&gt; &lt;Glyphs FontURI="Font1.ttf"/&gt; &lt;/FixedPage&gt;</pre>
Font1.ttf	...binary font data...
Other resources	...resource data...
FixedDocument1/[1].last.piece	<pre>&lt;PageContent Source="Page2"/&gt;</pre>
Page2	<pre>&lt;FixedPage xmlns= "http://schemas.microsoft.com/xps/2005/06" ...&gt;...&lt;/FixedPage&gt;</pre>
FixedDocument1/[2].last.piece	<pre>&lt;/FixedDocument&gt;</pre>
Sequence1/[1].last.piece	<pre>&lt;DocumentReference Source="FixedDocument2" /&gt; &lt;/FixedDocumentSequence&gt;</pre>
FixedDocument2	<pre>&lt;FixedDocument xmlns= "http://schemas.microsoft.com/xps/2005/06"&gt; &lt;PageContent Source="Page3"/&gt; &lt;/FixedDocument&gt;</pre>
Page3	<pre>&lt;FixedPage xmlns="http://schemas.microsoft.com/xps/2005/06"&gt;</pre>



---

```

...>...</FixedPage>
_rels/.rels/[1].last.piece </Relationships>

```

---

1 *end example]*

### 2 **17.1.3 Consuming Interleaved Packages**

3 Consumers MUST be able to consume packages regardless of their interleaving  
4 structure [M10.2]. Consumers that lack the resources to process a part MUST  
5 indicate an error condition [M10.3]. Such a resource constraint exists when a  
6 consumer lacks sufficient memory resources to hold enough of the package to  
7 resolve all the references required to process a part.

8 To address resource constraints:

- 9 • Consumers MAY discard FixedPage parts once they have been processed  
10 [O10.3]
- 11 • Consumers MAY discard FixedDocument and FixedDocumentSequence parts  
12 after all their child elements and their closing tags have been processed  
13 [O10.4].
- 14 • In the absence of explicit directives to the contrary (see §17.1.4, “Consumers  
15 with Resource Constraints,” below), consumers MAY discard parts as directed  
16 by the DiscardControl part [O10.5]. Consumers MUST NOT discard any other  
17 parts [*Example*: Such as parts containing fonts, images, or other resources  
18 *end example]* unless they have the ability to access the parts again [M10.4].

19 If a consumer encounters a reference to an unknown part, it MUST continue to  
20 receive further bytes of the package until the unknown part has been transmitted *or*  
21 until the end of the package is reached (indicating an error condition) [M10.5].

### 22 **17.1.4 Consumers with Resource Constraints**

23 To produce an XPS Document for streaming consumption by consumers with limited  
24 memory resources, some producers MAY choose a suitable interleaving order by  
25 modeling the resource management behavior of the consumer [O10.6]. These  
26 producers, referred to as **drivers**, must have specific knowledge of the XPS  
27 Document consumer. Due to resource constraints, some consumers are unable to  
28 consume arbitrary XPS Documents and always require assistance from an external  
29 driver.

30 When some consumers with limited memory resources receive a XPS Document in a  
31 streaming fashion, there might be an opportunity to discard parts when necessary  
32 and reload them again when needed. Producers, such as drivers, that target such  
33 consumers SHOULD follow these steps [S10.7]:

- 34 • Conservatively model the memory usage of the device.
- 35 • Interleave pieces of parts in the correct order.
- 36 • Decide when certain parts can be discarded by the consumer and inform the  
37 consumer within the package stream (see §17.1.4.1, “DiscardControl Part,”  
38 below).
- 39 • Add to the package a uniquely named copy of a resource that could have been  
40 discarded, if the resource is referenced by a part sent later in the stream.

1 Those later references are also updated to refer to the new copy of the  
2 resource.

### 3 17.1.4.1 DiscardControl Part

4 In addition to optimally ordering interleaved parts, producers may support  
5 consumers with resource constraints by means of the DiscardControl part. The  
6 DiscardControl part is a well-known part containing a list of resources that are safe  
7 for the consumer to discard. DiscardControl parts are stored in XPS Documents in an  
8 interleaved fashion, allowing a resource-constrained consumer to discard a part as  
9 soon as it appears in the DiscardControl part. DiscardControl parts are targeted with  
10 a DiscardControl package relationship, as specified in H, "Standard Namespaces and  
11 Content Types." There MUST NOT be more than one DiscardControl package  
12 relationship [M10.23]. The DiscardControl part MUST NOT reference itself [M10.6];  
13 doing so is considered an error.

14 DiscardControl parts that are not well-formed SHOULD NOT be processed and an  
15 error SHOULD NOT be reported [S10.8]. The consumer MAY decide to ignore the  
16 malformed DiscardControl part in its entirety or from the first malformed node  
17 onward [O10.7].

18 In some cases, producers might rewrite the contents of a package so that parts are  
19 provided more than once, allowing consumers to discard a part in order to free  
20 resources for additional processing. Each instance of a part MUST be stored as a  
21 new, uniquely named part in the package [M10.24].

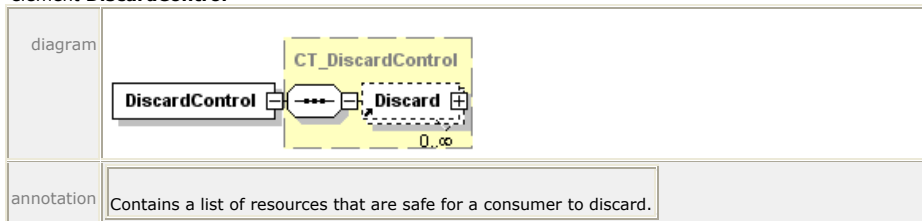
22 *Example 17-3. A DiscardControl part*

```
23 <DiscardControl xmlns="http://schemas.microsoft.com/xps/2005/06/discard-
24 control">
25   <!-- May discard partname1 as soon as starting to process
26     pagel1.xaml -->
27   <Discard SentinelPage="/pagel1.xaml" Target="/partname1" />
28   <!-- May discard partname2 as soon as starting to process
29     pagel3.xaml -->
30   <Discard SentinelPage="/pagel3.xaml" Target="/partname2" />
31   ...
32 </DiscardControl>
```

33 *end example]*

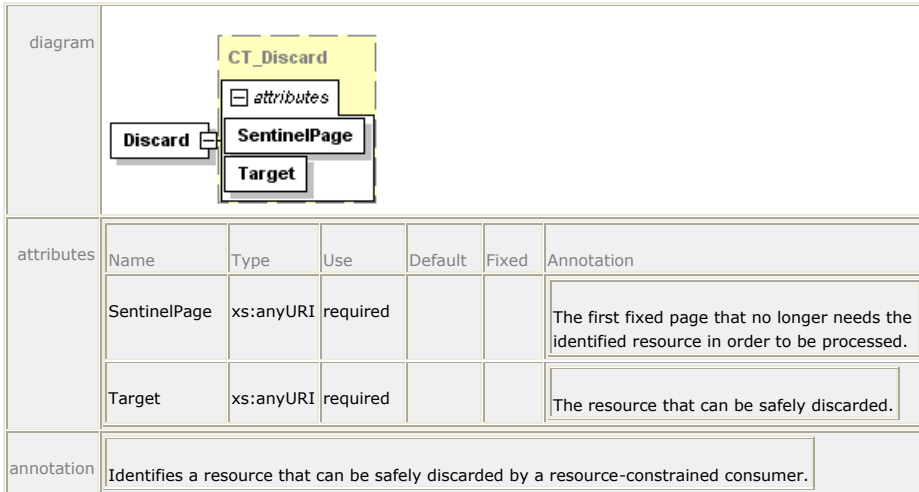
#### 34 17.1.4.1.1 <DiscardControl> Element

35 element **DiscardControl**



#### 36 17.1.4.1.2 <Discard> Element

37 element **Discard**



1 Parts that can be discarded are identified in a <Discard> element by the **Target**  
 2 attribute value, which is expressed as relative to the package root, and by the  
 3 **SentinelPage** attribute value, which identifies the first FixedPage part that no longer  
 4 requires the discarded part. (The processing order for FixedPage parts is implied by  
 5 the order of <PageContent> element references in the FixedDocument part.  
 6 Therefore, the value of the **SentinelPage** attribute is unambiguous.)

7 If either the **Target** attribute or the **SentinelPage** attribute contain an invalid  
 8 reference (refer outside the package), the respective <Discard> element **MUST** be  
 9 ignored [M10.7]. If a <Discard> element is encountered where either or both of the  
 10 **Target** attribute and **SentinelPage** attribute identify a part which has not been  
 11 processed yet (is still unknown), the <Discard> element **SHOULD** be retained until  
 12 both parts identified by the **Target** attribute and **SentinelPage** attribute have been  
 13 processed or until the end of the package is reached [S10.9].

#### 14 17.1.5 Interleaving Optimizations and Digital Signatures

15 In general, it is not feasible to produce well-ordered, interleaved ZIP packages *and*  
 16 apply digital signatures in a way that enables reasonable consumption scenarios for  
 17 the following reasons:

- 18 • The digital signature parts must be known to consumers before they process  
 19 other signed parts because the selected hash-methods and transforms must  
 20 be known. A streaming consumer might not be able to access part data after it  
 21 has been processed for printing.
- 22 • Producers cannot create the digital signature parts before producing the  
 23 signed packages.
- 24 • There are cyclic dependencies with signed relationship parts containing the  
 25 relationship to the signature parts themselves.

26 Therefore, when adding a digital signature to an interleaved package, producers of  
 27 digitally signed documents that are intended for streaming consumption **SHOULD**

1 add all digital signature parts and the package relationship to the digital signature  
2 parts at the beginning of the package, before adding any other part [S10.10].

---

## 3 **17.2 Digital Signatures**

4 The digital signature specification for XPS Documents is described in the OPC  
5 specification. It allows users to sign arbitrary parts, relationship parts, and individual  
6 relationships. Although XPS Documents also use these digital signature mechanisms,  
7 they have a specific signature policy and a specific signing mechanism for documents  
8 containing co-signing requests.

### 9 **17.2.1 Signature Policy**

10 This specification defines the signature policy that governs the methods of signing  
11 and verifying signatures for XPS Documents. The XPS Document signature policy  
12 includes a specific set of signing rules and validity rules. All producers and  
13 consumers signing and verifying signatures for end users or applications **MUST**  
14 adhere to these rules consistently [M10.8] to ensure that end users can rely on  
15 applications to display accurate signature information.

16 When signing a document, users may choose to make any of the following actions  
17 invalidate the signature:

- 18 • Editing core properties
- 19 • Adding signatures

20 Consumers **SHOULD** inform the end user if an action they are going to take will  
21 invalidate an existing signature [S10.11]. Consumers **MUST NOT** disable functionality  
22 based on whether or not an action will invalidate a signature [M10.9].

#### 23 **17.2.1.1 Signing Rules**

24 An XPS Document **MUST** be considered signed according to the XPS Document  
25 signing policy, regardless of the validity of that signature, if the following **signing**  
26 **rules** are followed [M10.10]:

- 27 1. The following parts **MUST** be signed [M10.10]:
  - 28 a. The <SignedInfo> portion of the Digital Signature XML Signature part  
29 containing this signature.
  - 30 b. The FixedDocumentSequence part that is the target of the Start Part  
31 package relationship.
  - 32 c. All FixedDocument parts referenced in the markup of the  
33 FixedDocumentSequence part. (Adding a FixedDocument part to a signed  
34 XPS Document will invalidate the signature.)
  - 35 d. All FixedPage parts referenced by all signed FixedDocument parts.
  - 36 e. All parts associated with each signed FixedPage part by means of a  
37 Required Resource relationship (such as fonts, images, color profiles,  
38 remote resource dictionaries).
  - 39 f. All DocumentStructure parts associated via a Document Structure  
40 relationship with all signed FixedDocument parts.

- 1 g. All StoryFragments parts associated via Story Fragments relationship with  
2 all signed FixedPage parts.
- 3 h. All SignatureDefinitions parts associated via a Signature Definitions  
4 relationship with any signed FixedDocument part. (Once a document is  
5 signed, adding any new signature definitions will invalidate the signature.)
- 6 i. All Thumbnail parts associated via a Thumbnail relationship from the  
7 package root or with any signed FixedPage or FixedDocument part.
- 8 2. The following parts MAY be signed [M10.10]:
- 9 a. The CoreProperties part.
- 10 b. The Digital Signature Origin part.
- 11 c. A Digital Signature Certificate part.
- 12 d. PrintTicket parts.
- 13 e. DiscardControl parts.
- 14 3. All relationships with the following RelationshipTypes (see H, "Standard  
15 Namespaces and Content Types") MUST be signed [M10.10]:
- 16 a. StartPart relationship from the package root
- 17 b. DocumentStructure relationship from a FixedDocument part
- 18 c. StoryFragments relationship from a FixedPage part
- 19 d. Digital Signature Definitions relationship from a FixedDocument part
- 20 e. Required Resource relationship from a FixedPage part
- 21 f. Restricted Font relationship from a FixedDocument part
- 22 g. Thumbnail relationship from a FixedPage part, a FixedDocument part, or  
23 the package root
- 24 4. All relationships with the following RelationshipTypes MUST be signed if their  
25 Target part is signed [M10.10]:
- 26 a. Core Properties relationship
- 27 b. Digital Signature Origin relationship
- 28 c. Digital Signature Certificate relationship from a Digital Signature XML  
29 Signature part
- 30 d. PrintTicket relationship
- 31 e. DiscardControl relationship
- 32 5. Relationships with the following RelationshipTypes MAY be signed as a group  
33 (they MUST NOT be signed individually) [M10.10]:
- 34 a. All Digital Signature XML Signature relationships from the Digital Signature  
35 Origin part (signing all relationships of this RelationshipType will cause this  
36 signature to break when a new signature is added).
- 37 6. All of the above-referenced parts and relationships MUST be signed using a  
38 single digital signature [M10.10].
- 39 An XPS Document MUST NOT be considered signed according to the XPS Document  
40 signing policy if [M10.11]:

- 1 1. Any part not covered by the signing rules above is included in the signature.
  - 2 2. Any relationship not covered by the signing rules above is included in the
  - 3 signature.
- 4 An XPS Document digital signer MUST NOT sign an XPS Document that contains
- 5 content (parts or relationships parts) to be signed that defines the Markup
- 6 Compatibility namespace but the signer does not fully understand all elements,
- 7 attributes, and alternate content representations introduced through the markup
- 8 compatibility mechanisms [M10.12]. An XPS Document digital signer MAY choose not
- 9 to sign any content (parts or relationships parts) that defines the Markup
- 10 Compatibility namespace, even if the content is fully understood [O10.8].

#### 11 17.2.1.2 Signing Validity

12 An XPS Document digital signature MUST be shown as an **incompliant digital**

13 **signature** if [M10.13]:

- 14 • It violates any of the signing rules described above regarding parts or
- 15 relationships that MUST or MUST NOT be signed.

16 An XPS Document digital signature MUST be shown as a **broken digital signature** if

17 [M10.14]:

- 18 • It is not an incompliant digital signature, but the signature fails the signature
- 19 validation routines described in the Open Packaging Conventions.

20 An XPS Document digital signature MUST be shown as a **questionable digital**

21 **signature** if any of the following are true [M10.15]:

- 22 • It is not an incompliant or broken digital signature, but the certificate cannot
- 23 be authenticated against the certificate authority.
- 24 • It is not an incompliant or broken digital signature, but the signed content
- 25 (parts and relationships) contain elements or attributes from an unknown
- 26 namespace introduced through the Markup Compatibility mechanisms.

27 An XPS Document digital signature MAY be shown as a questionable digital signature

28 if [O10.9]:

- 29 • It is not an incompliant or broken digital signature, but contains some other
- 30 detectable problem at the discretion of the consumer.

31 An XPS Document digital signature MUST be shown as a **valid digital signature** if

32 [M10.16]:

- 33 • It is not an incompliant, broken, or questionable digital signature.

#### 34 17.2.1.3 Adding Signatures

35 XPS Documents MAY be signed more than once [O10.10]. A user who signs an XPS

36 Document might or might not want to allow any additional signing of the document.

37 To prohibit additional signatures in an XPS Document, the signing application MUST

38 sign all the Digital Signature Origin part's relationships of relationship type Digital

39 Signature with the same signature as the rest of the content [M10.17].

1 **17.2.1.4 Certificate Store**

2 XPS Document signatures MUST NOT refer to a remote certificate store (certificate  
3 not contained in the XPS Document). All certificates MUST be stored in the XPS  
4 Document either as a Certificate part or in the Digital Signature XML Signature part  
5 [M10.18].

6 **17.2.1.5 Printing Signed Documents**

7 When printing signed documents, the PrintTicket setting  
8 JobDigitalSignatureProcessing SHOULD be used to control the digital signature  
9 processing behavior [S10.12]. Producers MAY include the  
10 JobDigitalSignatureProcessing setting in the job-level PrintTicket within the XPS  
11 Document content [O10.11]. Consumers SHOULD process this PrintTicket setting, if  
12 present [S10.12]. For more information, see the Print Schema specification.

13 *Table 17-1. JobDigitalSignatureProcessing PrintTicket settings*

Name	Description
PrintInvalidSignatures	Print the job regardless of the validity of the digital signatures. Digital signatures can be ignored.
PrintInvalidSignaturesWithErrorReport	Print the job regardless of the validity of the digital signatures. In the event an invalid signature is encountered, an error page should print at the end of the job. Digital signatures cannot be ignored.
PrintOnlyValidSignatures	Print the job only if all digital signatures are valid. Digital signatures cannot be ignored.

14 **17.2.2 Signature Definitions**

15 In some workflow scenarios, documents must be signed as a means of approving  
16 their content. [*Example*: Document producers may be required to sign their  
17 documents in order to provide proof of authenticity. *end example*] In other cases,  
18 reviewers may be required to co-sign content before it can be submitted for  
19 publication. These requirements can be fulfilled with a digitally signed XPS  
20 Document.

21 Whereas the XPS package model supports the signing of arbitrary content in a  
22 package, an XPS Document signing workflow requires additional features, including  
23 the ability to specify co-signature requirements and to include workflow-specific  
24 signature information in the document. XPS Document authors and signing parties  
25 provide such information in an XML [signature definition](#).

26 Signature definitions are represented by <SignatureDefinition> elements within a  
27 single <SignatureDefinitions> element.

1 *Example 17-4. A SignatureDefinitions part*

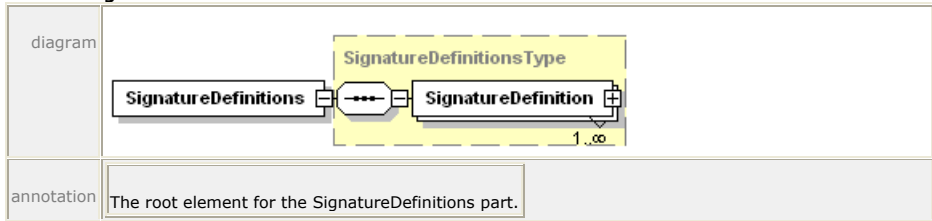
```

2 <SignatureDefinitions xmlns="http://schemas.microsoft.com/xps/2005/06/
3 signature-definitions">
4 <SignatureDefinition SignerName="Dorena Paschke" SpotID="AE2351H">
5 <SpotLocation
6 PageURI="/Documents/1/Pages/2.fpage"
7 StartX="0.0"
8 StartY="0.0" />
9 <Intent>I have read and agree</Intent>
10 <SignBy>2005-08-20</SignBy>
11 <SigningLocation>Redmond, WA</SigningLocation>
12 </SignatureDefinition>
13 </SignatureDefinitions>
    
```

14 *end example]*

15 **17.2.2.1 <SignatureDefinitions> Element**

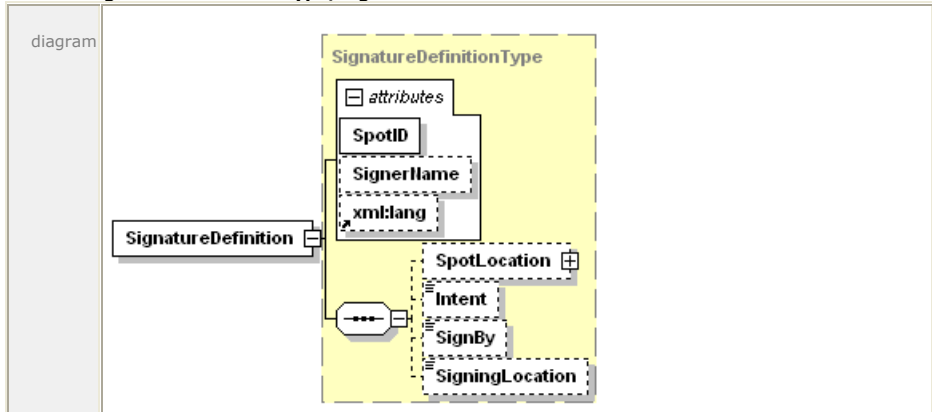
16 element **SignatureDefinitions**



17 If the SignatureDefinitions part exists, it MUST contain only one  
 18 <SignatureDefinitions> element [M2.72]. The XML namespace for the  
 19 <SignatureDefinitions> element is specified in §H.1, "XML Namespace URIs," on  
 20 page 405.

21 **17.2.2.2 <SignatureDefinition> Element**

22 element **SignatureDefinitionsType/SignatureDefinition**





attributes	Name	Type	Use	Default	Fixed	Annotation
	SpotID	xs:ID	required			A globally unique identifier for this signature spot.
	SignerName	xs:string				A string representing the identity of the individual who is requested to sign the XPS Document, or the name of the individual who has signed the XPS Document.
	xml:lang					Specifies the language used for the current element and its descendants. The language is specified according to RFC 3066.
annotation	A single signature definition.					

1 If the SignatureDefinitions part exists, there MUST be *at least* one  
 2 <SignatureDefinition> element [M2.72].

3 **17.2.2.2.1 SpotID Attribute**

4 The SpotID attribute is REQUIRED [M2.72]. This attribute MAY be used to link an  
 5 existing signature to the <SignatureDefinition> element [O10.12]. The value of this  
 6 attribute MUST be globally unique to ensure that a Signature part can be linked to  
 7 only one <SignatureDefinition> element [M2.72]. To link a <SignatureDefinition> to  
 8 a signature, the value of the **SpotID** MUST be specified in the **Id** attribute of the  
 9 corresponding <Signature> element in the Digital Signature XML Signature part  
 10 [M10.19]. For more information, see "Digital Signatures" in the OPC specification.

11 **17.2.2.3 <SpotLocation> Element**

12 element **SignatureDefinitionType/SpotLocation**

attributes	Name	Type	Use	Default	Fixed	Annotation
	PageURI	xs:anyURI	required			Specifies the page on which the signature spot should be displayed.
	StartX	xs:double	required			Specifies the x coordinate of the origin point (upper-left corner) on the page where the
	StartY					
diagram	<pre> classDiagram     class SpotLocationType {         +attributes         +PageURI         +StartX         +StartY     }     class SpotLocation     SpotLocationType -- SpotLocation             </pre>					

					signature spot should be displayed.
	StartY	xs:double	required		Specifies the y coordinate of the origin point (upper-left corner) on the page where the signature spot should be displayed.
annotation	Specifies where a consumer should place a signature spot.				

1 The <SpotLocation> element is OPTIONAL [M2.72]. It specifies where an XPS  
 2 Document viewer should place a visual representation or **signature spot** to indicate  
 3 that a digital signature has been applied or requested. The viewing consumer  
 4 SHOULD use the values specified in this element. Due to space and rendering  
 5 limitations, producers MUST NOT assume that consumers will use these values  
 6 [M10.20]. If the location specified by this element is not used, it is RECOMMENDED  
 7 that consumers choose a location that does not contain any page content [S10.13].

8 The size and shape of the signature spot are determined by the consumer.  
 9 Consumers MAY choose a size and shape based on the desired display information  
 10 and page content [O10.13]. However, it is RECOMMENDED that they render  
 11 signature spots as consistently sized rectangles that include the signer name, the  
 12 intent, the signing location, and the scope of the XPS Document to be signed  
 13 [S10.14]. It is also RECOMMENDED that the signature spot be a clickable area used  
 14 to launch the digital signing process [S10.15].

15 *Figure 17-1. A sample signature spot*



16

17 **17.2.2.4 <Intent> Element**


18 element **SignatureDefinitionType/Intent**

diagram	
annotation	A string that represents the intent to which the signing party agrees when signing the document.

19 Consumers MUST display the full value of the <Intent> element to the signing party,  
 20 either in the signature spot or through some other mechanism [M10.21].

21 *[Note: Consumers that wish to display signature spots must consider the implications*  
 22 *of supporting any Unicode character that can be specified in the <Intent> element.*  
 23 *They must also make decisions about the appropriate font face and size to use as*  
 24 *well as determine the proper layout and interactivity of the signature spot. These*  
 25 *decisions are implementation-specific. end note]*

1 **17.2.2.5 <SignBy> Element**2 element **SignatureDefinitionType/SignBy**

diagram	
annotation	The date and time by which the requested party is to sign the XPS Document.

3 If specified, the consumer SHOULD NOT allow the signing party to sign the document  
 4 using this particular signature spot after the date and time specified [S10.16]. The  
 5 date and time MUST be specified as a complete date plus hours, minutes, and  
 6 seconds in UTC time, as described in the W3C Note "Date and Time Formats"  
 7 [M10.22], [*Example*: "2006-12-31T23:59:59Z" or 11:59 PM (UTC) on December 31,  
 8 2006. *end example*]

9 **17.2.2.6 <SigningLocation> Element**10 element **SignatureDefinitionType/SigningLocation**

diagram	
annotation	The legal location where the document is signed.

11 The <SigningLocation> element MAY be set by the original producer of the XPS  
 12 Document or by the signing party at the time of signing [O10.14].

13 **17.3 Core Properties**

14 XPS Documents use the Core Properties part described in the OPC. The core  
 15 properties specified in that part SHOULD refer to the entire fixed payload, including  
 16 the root FixedDocumentSequence part and the compilation of all FixedDocument  
 17 parts it references [S10.17].



## 1 **18. Rendering Rules**

2 The set of rules described here ensures precise and consistent rendering of XPS  
3 Document markup across various implementations. Producers MUST generate XPS  
4 Documents that can be accurately rendered by following the rules described in this  
5 clause [M11.1]. Consumers MUST adhere to the rules described in this clause when  
6 rendering XPS Documents [M11.1]. In addition to rules for visual elements,  
7 implementation limits are also discussed.

---

### 8 **18.1 Coordinate System and Rendering Placement**

9 In the  $x,y$  coordinate system, one unit is initially equal to  $1/96''$ , expressed as a real  
10 number. The initial origin of the coordinate system is the top left corner of the fixed  
11 page. The  $x$ -coordinate value increases from left to right; the  $y$ -coordinate value  
12 increases from top to bottom.

13 A `RenderTransform` property can be specified on any path, glyphs, or canvas to apply  
14 an affine transform to the current coordinate system.

15 A `Transform` property can be specified on any visual brush, image brush, linear  
16 gradient brush, radial gradient brush, or path geometry to apply an affine transform  
17 to the current coordinate system.

#### 18 **18.1.1 Page Dimensions**

19 The logical page dimensions correspond to the media size specified in the application  
20 page layout and are specified by the **Width** and **Height** attributes of the  
21 `<FixedPage>` element. Further optional attributes on the `<FixedPage>` element are  
22 used to specify details about the areas of the fixed page that contain rendered  
23 content. For more information, see §10.3, "`<FixedPage>` Element," on page 53.

24 The physical page dimensions correspond to the media size specified for printing.  
25 The physical page dimensions are specified in the `PrintTicket PageMediaSize`  
26 keyword.

27 The interaction of the logical page dimensions and the physical page dimensions is  
28 print-specific. For more information, see §10.3.3, "Page Size Terminology,"  
29 and §10.3.4, "Media Orientation and Scaling."

#### 30 **18.1.2 Rounding of Coordinates**

31 All computations on coordinate values SHOULD be performed with at least single  
32 floating-point precision [S11.1]. Final conversion (after all transforms have been  
33 computed) to device coordinates SHOULD retain at least as much fractional precision  
34 as a 28.4 fixed-point representation before performing pixel coverage calculations  
35 [S11.1].

36 Very high resolution devices MAY use lower fractional precision to represent device  
37 coordinates [O11.1].

1 When converting from real-number coordinate values to device coordinate values,  
2 rounding is performed according to the following rule:

3 
$$\text{coord}_d = \text{ROUND}(\text{coord}_r * 16.0) / 16$$

4 Where  $\text{coord}_r$  expresses a real-number coordinate value and  $\text{coord}_d$  expresses a  
5 device coordinate value.

### 6 **18.1.3 Transforms**

7 XPS Document markup supports affine transforms as expressed through the  
8 `RenderTransform` and `Transform` properties. An affine transform is represented as a  
9 list of six real numbers:  $m_{11}$ ,  $m_{12}$ ,  $m_{21}$ ,  $m_{22}$ ,  $dx$ ,  $dy$ . (For markup details,  
10 see §14.4, "Positioning Content," on page 185.)

11 The full matrix is as follows:

$$\begin{bmatrix} m_{11} & m_{12} & 0 \\ m_{21} & m_{22} & 0 \\ dx & dy & 1 \end{bmatrix}$$

12 A given  $x,y$  coordinate is transformed with a render transform to yield the resulting  
13 coordinate  $x',y'$  by applying the following computations:

14 
$$\begin{aligned} x' &= x * m_{11} + y * m_{21} + dx \\ y' &= x * m_{12} + y * m_{22} + dy \end{aligned}$$

16 When rendering a child or descendant element, the effective transform used for  
17 rendering is the concatenation of all the transforms specified by the  
18 `RenderTransform` or `Transform` property on parent or ancestor elements, starting  
19 from the outermost ancestor.

20 Non-invertible effective transforms can be specified in markup or occur as a result of  
21 limited numerical precision during concatenation. If a non-invertible transform is  
22 encountered during rendering, consumers MUST omit rendering the affected element  
23 and all of its child and descendant elements [M11.2].

24 If a non-invertible transform is encountered on a brush (as specified directly on the  
25 brush, as a result of the **Viewbox** or **Viewport** attributes, or through  
26 concatenation), the brush is treated according to §18.7.1, "Small Tiles," on page  
27 299.

28 The `Width` and `Height` values specified in the **Viewbox** and **Viewport** attributes of  
29 an `<ImageBrush>` or `<VisualBrush>` element MUST NOT be negative [M2.72].

30 If a non-invertible transform is encountered on a geometry (as specified directly on  
31 the geometry or through concatenation), the geometry MUST be considered to  
32 contain no area [M11.3].

33 A final, device-dependent step using the horizontal resolution and vertical resolution  
34 of the device converts the resulting coordinates  $x',y'$  to device coordinates  $x'',y''$ , as  
35 follows:

36 
$$\begin{aligned} x'' &= x' * R_x / 96 \\ y'' &= y' * R_y / 96 \end{aligned}$$

1 Where  $R_x$  is the horizontal resolution and  $R_y$  is the vertical resolution of the device,  
2 specified in device pixels per inch.

### 3 **18.1.4 Pixel Center Location, Pixel Placement, and Pixel Inclusion**

4 A pixel covers the range from  $x$  to  $x+1$ .

5 An *ideal* consumer implementation SHOULD render pixels in an 8x8 sub-pixel space,  
6 perform an 8x8 box filter sampling, and set the pixel to the resulting color value  
7 [S11.2]. Other implementations MAY use different rendering logic as long as it  
8 closely approximates this logic [O11.2].

9 When rendering a shape, a *practical* implementation (such as a bi-tonal printing  
10 device) SHOULD turn on each pixel whose center (at  $x+0.5$ ) is covered by the shape,  
11 or is touched by the shape with the shape extending beyond the pixel center in the  
12 positive  $x$  or  $y$  direction of the device [S11.3]. Devices MAY use sub-pixel masking  
13 instead [O11.3].

14 By definition, a shape with an area width of 0 (that is, no included area) does not  
15 touch or cover any pixel centers. A stroke with a width of 0 is treated in the same  
16 manner.

17 As a result of these rules, the behavior for very thin lines is implementation-specific:

- 18 • An implementation capable of anti-aliasing MAY draw a thin line in a way that  
19 blends with the background to varying degrees [O11.4].
- 20 • A bi-tonal implementation on a printer MAY draw thin lines with or without  
21 drop-outs, or by applying half-toning, depending on the desired output quality  
22 [O11.5].

23 [Note: Also see section 11.6.12, "Consistent Nominal Stroke Width" for discussion of  
24 thin strokes. *end note*]

### 25 **18.1.5 Maximum Placement Error**

26 When rendering geometries, consumers SHOULD render curves so they appear  
27 smooth from a normal viewing distance [S11.4]. Producers MUST NOT assume a  
28 specific placement error for curve decomposition or rely on side-effects of a specific  
29 consumer implementation [M11.4].

### 30 **18.1.6 Pixel Placement for Glyphs**

31 Regardless of other rules expressed here, consumers MAY apply pixel placement  
32 rules optimized for character rendering to individual glyphs in a <Glyphs> element  
33 [O11.6]. Such rules may result from font hinting applied by the typeface scaler used  
34 by a consumer implementation.

### 35 **18.1.7 Abutment of Shapes**

36 When no anti-aliasing is used, abutting shapes that share the same device  
37 coordinates for the end-points and control-points of an edge SHOULD be rendered  
38 without overlap and without gaps [S11.5]. Ideally, an implementation SHOULD also  
39 follow this rule for shapes that are mathematically abutting without sharing device  
40 coordinates for end-points and control-points of edges [S11.5].

### 1 18.1.8 Clipping Behavior

2 Clipping occurs as if a mask were created from the clip geometry according to the  
3 pixel placement rules defined in §18.1.4, "Pixel Center Location, Pixel Placement,  
4 and Pixel Inclusion," on page 269. An ideal consumer SHOULD create such a mask in  
5 an 8x8 sub-pixel space and subsequently draw only those sub-pixels of a shape that  
6 correspond to "ON" sub-pixels in the mask [S11.6].

7 A practical implementation (such as a bi-tonal printing device) SHOULD create a  
8 pixel mask according to Point 2 of §18.1.4, "Pixel Center Location, Pixel Placement,  
9 and Pixel Inclusion," and subsequently draw only those pixels of a shape that  
10 correspond to "ON" pixels in the mask. In creating the mask and drawing the shape,  
11 the abutment of shapes rule SHOULD be observed so that no pixel of the shape is  
12 drawn that would not have been drawn if the clip geometry were another abutting  
13 shape [S11.7]. Devices MAY use sub-pixel masking instead [O11.3].

---

## 14 18.2 Implementation Limits

15 XPS Document markup does not assume fixed implementation limits. However,  
16 consumers may have specific implementation limits imposed by their operating  
17 environment. XPS Document markup has been designed so that even complex pages  
18 can be represented accurately and with high fidelity.

19 A typical consumer implementation SHOULD be able to process markup with the  
20 characteristics indicated in Table 18–1 [S11.8]. Encountering markup with  
21 characteristics outside of the consumer-specific implementation limits MUST cause an  
22 error condition [M11.5].

23 Table 18–1 provides the RECOMMENDED minimum requirements for individual  
24 elements [S11.8]. Consumers also have limits on the total number of elements, as  
25 imposed by available memory. Producers SHOULD produce only XPS Documents that  
26 stay within these implementation limits [S11.8].

27 In order to process pages that contain a large number of elements, consumers MAY  
28 implement support for the DiscardControl part in order to discard elements that have  
29 already been processed [O10.5].

30 *Table 18–1. Recommended minimum processing requirements*

Characteristic	Type	Limit	Description
Coordinates/transformation matrix elements	Real number	+/- 10 <sup>12</sup>	Largest and smallest coordinate values. Calculations involving numbers close to this limit within a few orders of magnitude are likely to be inaccurate.
Smallest representable non-zero value		+/- 10 <sup>-12</sup>	Coordinate values closest to 0 without rounding to 0. Calculations involving numbers close to this limit within a few orders of magnitude are likely to be inaccurate.
Required precision for		Single	Coordinates are real numbers



Characteristic	Type	Limit	Description
coordinates		floating point	and SHOULD be computed with at least single floating point precision [S11.9].
Nested <Canvas> elements		At least 16	Depth of nested <Canvas> elements.
Nested <VisualBrush> elements		At least 16	Depth of nested <VisualBrush> elements within the Visual property. If the nesting level is higher than the limit, a consumer SHOULD attempt to flatten the nested content to a bitmap representation rather than failing to draw [S11.10].
Total number of points in a path figure		100,000	
Total number of points in a segment		100,000	
Total number of points in a geometry		100,000	
Total number of elements per page		1,000,000	
Number of glyphs per <Glyphs> element		5,000	
Number of elements in a single resource dictionary		10,000	
Total number of elements in all resource dictionaries of an individual page		10,000	
Total number of resource dictionaries in nested canvas scope		Number of nested <Canvas> elements + 1	The <FixedPage> element and each nested <Canvas> element can have at most one associated <ResourceDictionary> element
Number of gradient stops in a gradient brush		100	
Number of fixed documents in a fixed document sequence		1,000	
Number of fixed pages in a fixed document		1,000,000	
Number of dash-gap segments in StrokeDashArray property		No preset limit	Practical number of dash-gap segments depends on the StrokeThickness and the total length of the stroked path.

Characteristic	Type	Limit	Description
Total size of XPS Document markup per page	Bytes	64,000,000	Total size of markup after removing all unnecessary whitespace (according to the schema in B), assuming markup elements are specified in the default namespace without namespace prefixes, and assuming the most compact representation of all attributes using abbreviated syntax where possible.

## 1 18.3 Gradient Computations

2 To ensure the greatest possible consistency among consumers, gradients SHOULD be  
3 rendered according to the guidelines described in this section [S11.11].

### 4 18.3.1 All Gradients

5 Linear gradients and radial gradients share a common set of recommended  
6 operations for pre-processing gradient stops and blending colors. These are  
7 described below.

#### 8 18.3.1.1 Gradient Stop Pre-Processing

9 Consumers SHOULD pre-process gradient stops for all gradients using the following  
10 steps [S11.12]:

- 11 1. Sort all gradient stops by their respective offset values in ascending order.  
12 When two or more gradient stops have the same offset value, preserve their  
13 relative order from the markup while sorting. When more than two gradient  
14 stops have the same offset value, remove all but the first and last gradient  
15 stops having the same offset value.
- 16 2. If no gradient stop with an offset of 0.0 exists,  
17 a. And no gradient stop with an offset less than 0.0 exists, create an artificial  
18 gradient stop having an offset of 0.0 and a color of the gradient stop with  
19 the smallest offset value.  
20 b. And a gradient stop with an offset less than 0.0 exists and a gradient stop  
21 with an offset greater than 0.0 exists, create an artificial gradient stop  
22 having an offset of 0.0 and a color interpolated between the two gradient  
23 stops surrounding 0.0. Discard all gradient stops with an offset less than  
24 0.0.  
25 c. And a gradient stop with an offset less than 0.0 exists and no gradient stop  
26 with an offset greater than 0.0 exists, create an artificial gradient stop  
27 having an offset of 0.0 and a color of the gradient stop with the largest  
28 offset value. Discard all gradient stop elements with an offset less than 0.0.  
29
- 30 3. If no gradient stop with an offset of 1.0 exists,

- 1 a. And no gradient stop with an offset of greater than 1.0 exists, create an
- 2 artificial gradient stop having an offset of 1.0 and a color equal to the color
- 3 of the gradient stop with the largest offset value.
- 4 b. And a gradient stop with an offset greater than 1.0 exists and a gradient
- 5 stop with an offset less than 1.0 exists, create an artificial gradient stop
- 6 having an offset of 1.0 and a color interpolated between the two
- 7 surrounding gradient stops. Discard all gradient stops with an offset greater
- 8 than 1.0.
- 9 c. And a gradient stop with an offset greater than 1.0 exists and no gradient
- 10 stop with an offset less than 1.0 exists, create an artificial gradient stop
- 11 having an offset of 1.0 and a color equal to that of the gradient stop with
- 12 the smallest offset value. Discard all gradient stops with an offset greater
- 13 than 1.0.

#### 14 18.3.1.2 Blending Colors

15 If any gradient stops use an sRGB or scRGB color specification or the consumer does  
 16 not understand the PageBlendColorSpace PrintTicket setting, consumers SHOULD  
 17 blend colors between gradient stops in the color space indicated by the  
 18 **ColorInterpolationMode** attribute of the gradient brush [S11.13]. If none of the  
 19 gradient stop elements uses an sRGB or scRGB color specification and the consumer  
 20 understands the PageBlendColorSpace PrintTicket setting, the PageBlendColorSpace  
 21 PrintTicket setting SHOULD be used [S11.13].

22 The function used for blending is:

```
23 BLEND(offset, clo, chi)
```

24 Where the offset is between 0 and 1. c<sub>lo</sub> and c<sub>hi</sub> designate the color values for an  
 25 offset of 0 and 1, respectively.

26 If a **ColorInterpolationMode** value of SRgbLinearInterpolation is used, the  
 27 BLEND() function SHOULD convert the color values to sRGB first, and then perform a  
 28 linear interpolation between them [S11.14].

29 If a **ColorInterpolationMode** value of ScRgbLinearInterpolation is used, the  
 30 BLEND() function SHOULD convert the color values to scRGB first, and then perform  
 31 a linear interpolation between them [S11.15].

32 If blending is performed in the color space identified by the PageBlendColorSpace  
 33 PrintTicket setting, it SHOULD be a linear, channel-by-channel blend operation  
 34 [S11.13].

35 In the presence of transformations or when individual gradient stops are very close  
 36 (separated by a few pixels or less in the device space), the local color gradient at the  
 37 offset used in the BLEND() function may be large, resulting in a large change over  
 38 the extent of a single device pixel. In this case, it is RECOMMENDED that the  
 39 BLEND() function interpolate the gradient over the extent of each device pixel  
 40 [S11.16]. However, the behavior MAY differ from this recommendation in an  
 41 implementation-specific manner [O11.7] and, therefore, producers SHOULD NOT rely  
 42 on a specific effect for such dense gradient specifications [S11.16].

43 As a consequence of this interpolation, radial gradients that define the gradient  
 44 origin on or outside ellipse create an "outside" area that can be rendered  
 45 inconsistently. The radial gradients that are affected are those that define multiple

1 gradient stops that are of different colors and are very close in Offset value to 0.0 or  
 2 1.0 (the gradient end points), for radial gradients with a **SpreadMethod** value of  
 3 Repeat or Reflect, respectively. For these affected gradients, consumers MAY use an  
 4 interpolated color value for the outside area [O11.8]. Depending on the resolution,  
 5 this can result in different colors than those defined by the gradient end points. The  
 6 closer a gradient stop is to the affected gradient end point, the more the rendering  
 7 results on different consumers and at different display resolutions can differ.  
 8 Producers SHOULD therefore either avoid such close gradient stops to the gradient  
 9 end point when specifying radial gradients where the outside area is visible or avoid  
 10 specifying radial gradients with a gradient origin on or outside the ellipse (in which  
 11 case there is no outside area) to ensure consistent rendering results [S11.17].

### 12 18.3.2 Linear Gradients

13 Consumers SHOULD render an element filled with a linear gradient brush such that  
 14 the appearance is the same as if the following steps had been taken [S11.11]:

- 15 1. Transform the **StartPoint** and **EndPoint** attribute values using the current  
 16 effective render transform (including the render transform for the element  
 17 being filled by the linear gradient brush and the brush's transform itself).
- 18 2. If the **SpreadMethod** value is Pad, the colors of points on the line defined by  
 19 the **StartPoint** and **EndPoint** attributes are defined by interpolating the  
 20 coordinates linearly, and each color component (such as R, G, B for sRGB and  
 21 scRGB) as well as the alpha component is interpolated between the  
 22 component values of the closest enclosing gradient stops:

```
23 For each offset (real number)  $t < 0$ :
24 {
25    $x(t) = (\text{EndPoint}_x - \text{StartPoint}_x) * t + \text{StartPoint}_x$ 
26    $y(t) = (\text{EndPoint}_y - \text{StartPoint}_y) * t + \text{StartPoint}_y$ 
27    $c(t) = c_{\text{first}}$ 
28    $a(t) = a_{\text{first}}$ 
29 }
```

30 Where  $c$  is the color component and  $a$  is the alpha component.  $c_{\text{first}}$  are the  
 31 color component values of the first gradient stop (after sorting) and  $a_{\text{first}}$  is  
 32 the alpha component value at the first gradient stop (after sorting).

```
33 For each offset (real number)  $0 \leq t \leq 1$ :
34 {
35    $x(t) = (\text{EndPoint}_x - \text{StartPoint}_x) * t + \text{StartPoint}_x$ 
36    $y(t) = (\text{EndPoint}_y - \text{StartPoint}_y) * t + \text{StartPoint}_y$ 
37    $c(t) = \text{BLEND}((t - t_{l_0}) / (t_{h_1} - t_{l_0}), c_{l_0}, c_{h_1})$ 
38    $a(t) = [(t - t_{l_0}) / (t_{h_1} - t_{l_0})] * (a_{h_1} - a_{l_0}) + a_{l_0}$ 
39 }
```

40 Where  $t_{l_0}$  and  $t_{h_1}$  are the offsets,  $c_{l_0}$  and  $c_{h_1}$  are the color component values at  
 41 the closest enclosing gradient stops (that is,  $t_{l_0} \leq t \leq t_{h_1}$ ) and  $a_{l_0}$  and  $a_{h_1}$   
 42 are the alpha component values at the closest enclosing gradient stops ( $t_{l_0} \leq$   
 43  $t \leq t_{h_1}$ ).

```
44 For each offset (real number)  $t > 1$ :
45 {
46    $x(t) = (\text{EndPoint}_x - \text{StartPoint}_x) * t + \text{StartPoint}_x$ 
47    $y(t) = (\text{EndPoint}_y - \text{StartPoint}_y) * t + \text{StartPoint}_y$ 
48    $c(t) = c_{\text{last}}$ 
49    $a(t) = a_{\text{last}}$ 
50 }
```

1 Where  $c_{last}$  are the color component values of the last gradient stop (after  
 2 sorting) and  $a_{last}$  is the alpha component value at the last gradient stop (after  
 3 sorting).

4 3. If the **SpreadMethod** value is Repeat, the colors of points on the line defined by  
 5 the **StartPoint** and **EndPoint** attributes are defined by interpolating the  
 6 coordinates linearly, and each color component (such as R, G, B for sRGB and  
 7 scRGB) as well as the alpha component is interpolated between the  
 8 component values of the closest enclosing gradient stops:

```
9 For each repetition (all integers) N:
10 {
11   For each offset (real number)  $0 \leq t < 1$ :
12   {
13      $x(t) = (EndPoint_x - StartPoint_x) * (N+t) + StartPoint_x$ 
14      $y(t) = (EndPoint_y - StartPoint_y) * (N+t) + StartPoint_y$ 
15      $c(t) = BLEND((t-t_{lo}) / (t_{hi}-t_{lo}), c_{lo}, c_{hi})$ 
16      $a(t) = [(t-t_{lo}) / (t_{hi}-t_{lo})] * (a_{hi}-a_{lo}) + a_{lo}$ 
17   }
18 }
```

19 Where  $c$  is the color component and  $a$  is the alpha component.  $t_{lo}$  and  $t_{hi}$  are  
 20 the offsets,  $c_{lo}$  and  $c_{hi}$  are the color component values at the closest enclosing  
 21 gradient stops (that is,  $t_{lo} \leq t \leq t_{hi}$ ) and  $a_{lo}$  and  $a_{hi}$  are the alpha  
 22 component values at the closest enclosing gradient stops ( $t_{lo} \leq t \leq t_{hi}$ ).

23 4. If the **SpreadMethod** value is Reflect, the colors of points on the line defined by  
 24 the **StartPoint** and **EndPoint** attributes are defined by interpolating the  
 25 coordinates linearly, and each color component (such as R, G, B for sRGB and  
 26 scRGB) as well as the alpha component is interpolated between the  
 27 component values of the closest enclosing gradient stops:

```
28 For each repetition (all integers) N:
29 {
30   For each offset (real number)  $0 \leq t \leq 1$ :
31   {
32     If (N is EVEN)
33     {
34        $x(t) = (EndPoint_x - StartPoint_x) * (N+t) + StartPoint_x$ 
35        $y(t) = (EndPoint_y - StartPoint_y) * (N+t) + StartPoint_y$ 
36     }
37     Else
38     {
39        $x(t) = (EndPoint_x - StartPoint_x) * (N+1-t) + StartPoint_x$ 
40        $y(t) = (EndPoint_y - StartPoint_y) * (N+1-t) + StartPoint_y$ 
41     }
42      $c(t) = BLEND((t-t_{lo}) / (t_{hi}-t_{lo}), c_{lo}, c_{hi})$ 
43      $a(t) = [(t-t_{lo}) / (t_{hi}-t_{lo})] * (a_{hi}-a_{lo}) + a_{lo}$ 
44   }
45 }
```

46 Where  $c$  is the color component and  $a$  is the alpha component.  $t_{lo}$  and  $t_{hi}$  are  
 47 the offsets,  $c_{lo}$  and  $c_{hi}$  are the color component values at the closest enclosing  
 48 gradient stops (that is,  $t_{lo} \leq t \leq t_{hi}$ ) and  $a_{lo}$  and  $a_{hi}$  are the alpha  
 49 component values at the closest enclosing gradient stops ( $t_{lo} \leq t \leq t_{hi}$ ).

50 5. The colors of points not on the extended line defined by the **StartPoint** and  
 51 **EndPoint** attributes are the same as the color of the closest point on the line  
 52 defined by the **StartPoint** and **EndPoint** attributes, measured in the  
 53 coordinate space as transformed by the current effective render transform

- 1 (including the render transform for the element being filled by the linear  
2 gradient brush and the brush's transform itself).
- 3 6. Clip the resulting set of points to the intersection of the current clip geometry  
4 and the path or glyphs to be filled. Both the clip and path (or glyphs) must be  
5 transformed according to the current effective render transform, including the  
6 render transform for the element being filled, but *not* including the transform  
7 of the linear gradient brush.
- 8 For purposes of the above steps, the closest enclosing gradient stops mean the  
9 gradient stops that, if the relative sequencing of the gradient stop offsets in the  
10 markup order is respected, are numerically closest to the interpolation point if that  
11 interpolation point were converted to an offset value and inserted in a sorted fashion  
12 into the list of gradient stops. [*Example*: If a gradient contains three gradient stops  
13 at offset values 0.0, 0.0, and 1.0, the closest enclosing gradient stops for any value  
14  $0 \leq \text{value} \leq 1$  are the second gradient stop (offset 0.0) and the third gradient  
15 stop (offset 1.0). *end example*]

### 16 18.3.3 Radial Gradients

17 Consumers SHOULD render an element filled with a radial gradient brush such that  
18 the appearance is the same as if these steps had been followed [S11.11]:

- 19 1. The boundary of the area filled by a radial gradient brush is defined by  
20 interpolating ellipses from the **GradientOrigin** value to the circumference of  
21 the ellipse centered at the point specified by the **Center** attribute with radii  
22 equal to the **RadiusX** and **RadiusY** attribute values (the interpolated ellipses  
23 and point being transformed by the current effective render transform,  
24 including the render transform for the element being filled by the radial  
25 gradient brush and the brush's transform itself). If the gradient origin is  
26 outside the circumference of the ellipse specified, the effect will be as if a cone  
27 were drawn, tapering to the gradient origin.
- 28 2. If the **SpreadMethod** value is Pad, the centers and radii of the interpolated  
29 ellipses are defined by linearly interpolating the center of the ellipse from the  
30 **GradientOrigin** attribute value to the **Center** attribute value, and  
31 simultaneously linearly interpolating the radii of the ellipse from 0 to the  
32 **RadiusX** and **RadiusY** attribute values:

```
33 For each offset (real number)  $0 \leq t \leq 1$ :
34 {
35    $c_x(t) = (\text{Center}_x - \text{GradientOrigin}_x) * t + \text{GradientOrigin}_x$ 
36    $c_y(t) = (\text{Center}_y - \text{GradientOrigin}_y) * t + \text{GradientOrigin}_y$ 
37    $r_x(t) = \text{RadiusX} * t$ 
38    $r_y(t) = \text{RadiusY} * t$ 
39 }
```

40 The ellipses defined by the interpolation are transformed by the current  
41 effective render transform, including the render transform for the element  
42 being filled by the radial gradient brush and the brush's transform itself.

- 43 3. The colors of the points within the boundary of this shape are defined as the  
44 color of the smallest interpolated ellipse containing the point. The color of an  
45 interpolated ellipse is defined by interpolating each color component (such as  
46 R, G, B for sRGB and scRGB) as well as the alpha component between the  
47 component values of the closest enclosing gradient stops:

```

1 For each offset (real number) 0 <= t <= 1:
2 {
3   c(t) = BLEND((t-t10)/(thi-t10), c10, chi)
4   a(t) = [(t-t10)/(thi-t10)]*(ahi-a10)+a10
5 }

```

6 Where  $t_{10}$  and  $t_{hi}$  are the offsets,  $c_{10}$  and  $c_{hi}$  are the color component values at  
7 the closest enclosing gradient stops (that is,  $t_{10} \leq t \leq t_{hi}$ ) and  $a_{10}$  and  $a_{hi}$   
8 are the alpha component values at the closest enclosing gradient stops ( $t_{10} \leq$   
9  $t \leq t_{hi}$ ).

- 10 4. If the **SpreadMethod** value is Repeat, the centers and radii of the  
11 interpolated ellipses are defined by linearly interpolating the center of the  
12 ellipse from the **GradientOrigin** attribute value to the **Center** attribute value,  
13 and simultaneously linearly interpolating the radii of the ellipse from 0 to  
14 **RadiusX** and **RadiusY** attribute values:

```

15 For each repetition (all non-negative integers) N:
16 {
17   For each offset (real number) 0 <= t < 1:
18   {
19     cx(t) = (Centerx-GradientOriginx)*(N+t)+GradientOriginx
20     cy(t) = (Centery-GradientOriginy)*(N+t)+GradientOriginy
21     rx(t) = RadiusX*(N + t)
22     ry(t) = RadiusY*(N + t)
23   }
24 }

```

25 The ellipses defined by the interpolation are transformed by the current  
26 effective render transform, including the render transform for the element  
27 being filled by the radial gradient brush and the brush's transform itself.

- 28 5. The colors of the points within the boundary of this shape are defined as the  
29 color of the smallest interpolated ellipse containing the point. The color of an  
30 interpolated ellipse is defined by interpolating each color component (such as  
31 R, G, B for sRGB and scRGB) as well as the alpha component between the  
32 component values of the closest enclosing gradient stops:

```

33 For each repetition (all non-negative integers) N:
34 {
35   For each offset (real number) 0 <= t < 1:
36   {
37     c(t) = BLEND((t-t10)/(thi-t10), c10, chi)
38     a(t) = [(t-t10)/(thi-t10)]*(ahi-a10)+a10
39   }
40 }

```

41 Where  $t_{10}$  and  $t_{hi}$  are the offsets,  $c_{10}$  and  $c_{hi}$  are the color component values at  
42 the closest enclosing gradient stops (that is,  $t_{10} \leq t \leq t_{hi}$ ) and  $a_{10}$  and  $a_{hi}$   
43 are the alpha component values at the closest enclosing gradient stops ( $t_{10} \leq$   
44  $t \leq t_{hi}$ ).

- 45 6. If the **SpreadMethod** value is Reflect, the centers and radii of the  
46 interpolated ellipses are defined by linearly interpolating the center of the  
47 ellipse from the **GradientOrigin** attribute value to the **Center** attribute value,  
48 and simultaneously linearly interpolating the radii of the ellipse from 0 to the  
49 **RadiusX** and **RadiusY** attribute values:

```

50 For each non-negative integer N:
51 {
52   For each offset (real number) 0 <= t <= 1:

```

```

1  {
2    cx(t) = (Centerx-GradientOriginx) * (N+t)+GradientOriginx
3    cy(t) = (Centery-GradientOriginy) * (N+t)+GradientOriginy
4    rx(t) = RadiusX * (N+t)
5    ry(t) = RadiusY * (N+t)
6  }
7  }

```

The ellipses defined by the interpolation are transformed by the current effective render transform, including the render transform for the element being filled by the radial gradient brush and the brush's transform itself.

- The colors of the points within the boundary of this shape are defined as the color of the smallest interpolated ellipse containing the point. The color of an interpolated ellipse is defined by interpolating each color component (such as R, G, B for sRGB and scRGB) as well as the alpha component between the component values of the closest enclosing gradient stops:

```

16 For each non-negative integer N:
17 {
18   For each offset (real number) 0 <= t <= 1:
19     {
20       If N is ODD
21         t' = 1-t
22       Else
23         t' = t
24
25       c(t) = BLEND((t'-t10) / (thi-t10), c10, chi)
26       a(t) = [(t'-t10) / (thi-t10)] * (ahi-a10) + a10
27     }
28 }

```

Where  $t_{10}$  and  $t_{hi}$  are the offsets,  $c_{10}$  and  $c_{hi}$  are the color component values at the closest enclosing gradient stops (that is,  $t_{10} \leq t \leq t_{hi}$ ) and  $a_{10}$  and  $a_{hi}$  are the alpha component values at the closest enclosing gradient stops ( $t_{10} \leq t \leq t_{hi}$ ).

- The colors of points outside the boundary of this shape (points which cannot be drawn by any combination of non-negative N and t) are defined as having the color and alpha defined in the gradient stop with the offset of 0.0 for radial gradients with a **SpreadMethod** value of Reflect and the color and alpha defined in the gradient stop with the offset of 1.0 for radial gradients with a **SpreadMethod** value of Repeat or Pad. The colors outside of the boundary of this shape may also vary in an implementation-specific manner (see §18.3.1.2, "Blending Colors" for more details).
- Clip the resulting set of points by the intersection of the current clip geometry and the path or glyphs to be filled. Both the clip and path (or glyphs) must be transformed according to the current effective render transform, including the render transform for the element being filled, but *not* including the transform of the radial gradient brush.

For purposes of the above steps, the closest enclosing gradient stops mean the gradient stops that, if the relative sequencing of the gradient stop offsets in the markup order is respected, are numerically closest to the interpolation point if that interpolation point were converted to an offset value and inserted in a sorted fashion into the list of gradient stops. [Example: If a gradient contains three gradient stops at offset values 0.0, 0.0, and 1.0, the closest enclosing gradient stops for any value



1 0 <= value <= 1 are the second gradient stop (offset 0.0) and the third gradient  
2 stop (offset 1.0). *end example*]

### 3 **18.4 Opacity Computations**

4 Opacity is used to transparently blend two elements when rendering, also known as  
5 alpha blending. The value of the Opacity property ranges from 0.0 (fully transparent)  
6 to 1.0 (fully opaque), inclusive. Values outside of this range are invalid.

7 The opacity is applied through the following computations, assuming source and  
8 destination values are not pre-multiplied. All opacity calculations SHOULD be  
9 performed with at least 8-bit precision to provide sufficient quality for nested content  
10 [S11.18].

11 Individual pixels are blended as defined below.

12 *Table 18-2. Opacity computation symbols*

Symbol	Description
$O_E$	Opacity attribute of element
$O_M$	Alpha value at corresponding pixel position in the <b>OpacityMask</b> attribute value
$A_S$	Alpha value present in source color
$R_S$	Red value present in source color
$G_S$	Green value present in source color
$B_S$	Blue value present in source color
$A_D$	Alpha value already present in destination surface
$R_D$	Red value already present in destination surface
$G_D$	Green value already present in destination surface
$B_D$	Blue value already present in destination surface
$A^*$	Resulting Alpha value for destination surface
$R^*$	Resulting Red value for destination surface
$G^*$	Resulting Green value for destination surface
$B^*$	Resulting Blue value for destination surface

13 All values designated with a <sub>T</sub> subscript (as in  $R_{T1}$ ) are temporary values.

14 The opacity is calculated as follows:

15 1. Multiply source alpha value with opacity value and alpha value of opacity  
16 mask.

$$17 \quad A_{S1} = A_S * O_E * O_M$$

18 2. Pre-multiply source alpha.

19 Omit this step if the source data specifies pre-multiplied alpha (see §18.4.1,  
20 "Pre-Multiplied Alpha and Superluminous Colors," on page 281 for details).

$$21 \quad A_{T1} = A_{S1}$$

$$22 \quad R_{T1} = R_S * A_{S1}$$

$$23 \quad G_{T1} = G_S * A_{S1}$$

```
1 BT1 = BS*AS1
```

### 2 3. Pre-multiply destination alpha.

3 Omit this step in consumers supporting superluminous colors (see §18.4.1,  
4 "Pre-Multiplied Alpha and Superluminous Colors," on page 281 for details).

```
5 AT2 = AD  
6 RT2 = RD*AD  
7 GT2 = GD*AD  
8 BT2 = BD*AD
```

### 9 4. Blend.

10 See §18.4.1, "Pre-Multiplied Alpha and Superluminous Colors," on page 281  
11 for special case handling.

```
12 AT3 = (1-AT1)*AT2+AT1  
13 RT3 = (1-AT1)*RT2+RT1  
14 GT3 = (1-AT1)*GT2+GT1  
15 BT3 = (1-AT1)*BT2+BT1
```

### 16 5. Reverse pre-multiplication.

17 Omit this step in consumers supporting superluminous colors. See  
18 also §18.4.1, "Pre-Multiplied Alpha and Superluminous Colors," on page 281.

19 The resulting color channel values are divided by the resulting alpha value. If  
20 the resulting alpha value is 0, all color channels are set to 0 by definition, as  
21 expressed in the If condition below. Each of  $R_{T3}$ ,  $G_{T3}$ ,  $B_{T3}$  is smaller than or  
22 equal to  $A_{T3}$  and, therefore, each of the resulting  $R^*$ ,  $G^*$ ,  $B^*$  is in the valid  
23 interval of  $[0.0,1.0]$  after the pre-multiplication is reversed.

```
24 If AT3 = 0  
25 {  
26   set all A* R* G* B* to 0.  
27 }  
28 Else  
29 {  
30   A* = AT3  
31   R* = RT3/AT3  
32   G* = GT3/AT3  
33   B* = BT3/AT3  
34 }
```

35 When blending colors in a color space other than sRGB, color channels are  
36 independently interpolated in a manner analogous to the RGB channel blending  
37 method described above.

### 1 18.4.1 Pre-Multiplied Alpha and Superluminous Colors

2 The alpha information in TIFF images using an ExtraSamples tag value of 1 and in  
3 Windows Media Photo images using pixel formats WICPixelFormat32bppPBGRA,  
4 WICPixelFormat64bppPRGBA or WICPixelFormat128bppPRGBAFloat MUST be  
5 interpreted as pre-multiplied alpha information [M11.6]. In certain scenarios (such  
6 as when rendering 3D scenes to a bitmap), producers MAY choose to create pre-  
7 multiplied bitmap data specifying "superluminous" colors [O11.9].

8 Superluminous colors are defined as pre-multiplied color values with an alpha value  
9 smaller than the individual color channel values but greater than or equal to 0.

10 The effect of composing superluminous colors on a background is similar to adding  
11 additional light of the source color to the destination, as opposed to regular alpha  
12 composition which works more like a colored filter. One can easily verify this  
13 statement by substituting 0 for  $A_{T1}$  in step 4 of the above opacity computations,  
14 which is simplified as follows (note that  $R_{T1}$ ,  $G_{T1}$ ,  $B_{T1}$  are not 0, because the pre-  
15 multiplication in step 2 has been skipped):

```
16  $A_{T3} = A_{T2}$ 
17  $R_{T3} = R_{T2} + R_{T1}$ 
18  $G_{T3} = G_{T2} + G_{T1}$ 
19  $B_{T3} = B_{T2} + B_{T1}$ 
```

20 Consumers supporting superluminous colors retain all temporary information in pre-  
21 multiplied formats. Note, that throughout the XPS specification non-pre-multiplied  
22 alpha processing is assumed. It is up to the implementer of such a consumer to  
23 identify equivalent composition and rendering rules for processing in pre-multiplied  
24 space.

25 Also note, when composing superluminous colors, management of out-of-gamut  
26 colors SHOULD be deferred until the result is rendered to the final target, at which  
27 point out-of-gamut colors are clipped or color managed [S11.19].

28 Consumers MAY handle superluminous colors or MAY instead choose to convert pre-  
29 multiplied source data containing superluminous colors to non-pre-multiplied data  
30 before composition by ignoring the superluminous portion of each color channel  
31 value [O11.10], as described in the following steps:

```
32 For each superluminous pixel with  $A < R$  or  $A < G$  or  $A < B$ 
33 {
34   If  $A = 0$ 
35   {
36      $R^* = 1$ 
37      $G^* = 1$ 
38      $B^* = 1$ 
39   }
40   Else
41   {
42      $A^* = A$ 
43      $R^* = \min(R/A, 1)$ 
44      $G^* = \min(G/A, 1)$ 
45      $B^* = \min(B/A, 1)$ 
46   }
47 }
```

---

## 18.5 Composition Rules

2 XPS Document page markup uses the painter's model with alpha channel.

3 Composition MUST have the same effect as the application of the following rules, in  
4 sequence [M11.7]:

- 5 1. In order to render a fixed page or canvas, a surface is created to hold the  
6 drawing content as it is composed. The color and appearance of this surface  
7 SHOULD match the destination color and appearance, typically a solid white  
8 background for a fixed page or transparent for a canvas [S11.20]. An  
9 implementation MAY choose to meet this goal by always initializing this  
10 surface's alpha channel to 0.0 (transparent) and the color value to black  
11 [O11.5].
- 12 2. The fixed page or canvas represents a surface onto which child elements are  
13 drawn. The child elements are drawn in the order they appear in markup. In  
14 practice, an implementation might represent the surface by a bitmap buffer  
15 large enough to hold all the drawing content produced when the child  
16 elements are rendered.
- 17 3. The contents appearing on the surface of canvas are transformed using the  
18 affine transform specified by the RenderTransform property of the canvas. (A  
19 fixed page does not have a RenderTransform property.)
- 20 4. All child elements are rendered to the surface and clipped to the imageable  
21 area of the physical display (such as a sheet of paper) of the fixed page or  
22 according to the Clip property of a canvas. The geometry value of the canvas'  
23 Clip property is also transformed using the affine transform specified by the  
24 RenderTransform property of the canvas.
- 25 5. If a path has a Stroke and a Fill property, and also specifies Opacity or  
26 OpacityMask property values, additional composition steps must be followed:
  - 27 a. Create a temporary canvas with the opacity, opacity mask, clip, and render  
28 transform specified by the path.
  - 29 b. Create a copy of the original path, remove all but the Fill property from the  
30 copy, and add the copy to the temporary canvas.
  - 31 c. Create another copy of the original path, remove all but the stroke-related  
32 properties (such as Stroke, StrokeThickness, and StrokeDashArray) from  
33 the copy, and add the copy to the temporary canvas.
  - 34 d. Do not draw the original path.
  - 35 e. Draw the temporary canvas, while recursively applying the composition  
36 rules.
- 37 6. If a grouping element (a <Canvas> element) has an Opacity or OpacityMask  
38 property, additional composition steps must be followed:
  - 39 a. Create a temporary surface and set its alpha channel to 0.0 (transparent)  
40 and its color value to black.
  - 41 b. Compose all child elements of the grouping element onto the temporary  
42 surface, while recursively applying the composition rules.
  - 43 c. Cumulatively apply the opacity of the grouping element and opacity mask  
44 to the alpha channel of the temporary surface.

- 1 d. Draw the contents of the temporary surface onto the containing surface.
- 2 7. If a non-grouping element (a <Path> or <Glyphs> element) has an Opacity
- 3 property, an OpacityMask property, or a fill or stroke using transparency, the
- 4 following additional composition steps must be taken:
  - 5 a. If the element has a RenderTransform property, apply it to the element and
  - 6 its Clip, Fill, Stroke, and OpacityMask properties, if present.
  - 7 b. Create a mask from the set of all painted pixels representing the child
  - 8 element (after the Clip property of the element has been applied).
- 9 8. Combine the Fill or Stroke property with the OpacityMask and the Opacity
- 10 property and apply to the surface through the computed mask. For more
- 11 information, see §14.1, "Opacity," on page 172.

12 The behavior that results from this process is:

- 13 • Opacity is not applied cumulatively to self-overlapping areas created when
- 14 rendering an individual <Glyphs> element.
- 15 • Opacity is not applied cumulatively to self-overlapping areas created by
- 16 <PathFigure> elements within the same path (see Example 18–1 on
- 17 page 284).
- 18 • Opacity is not applied cumulatively if the border of a path has self-
- 19 intersections. When the border of a path is stroked, the area of the path is
- 20 filled by first applying the brush specified by the Fill property. After filling the
- 21 area, the border is drawn using the stroke-related properties including the
- 22 brush specified by the Stroke property, with half the stroke width extending
- 23 outside the filled area and half extending inside (see Example 18–2 on
- 24 page 285). If the path has self-intersections, the opacity is not accumulated.
- 25 • The color of the stroke and the color of the filled area are combined on the
- 26 inside half of a stroked border (overlapping the filled area of the path) if the
- 27 brush specified by the Stroke property is transparent.
- 28 • If a path that has a stroked border has an opacity of less than 1.0 or an
- 29 opacity mask, the path (filled area and stroked border) is first rendered onto a
- 30 temporary surface using an opacity of 1.0 and no opacity mask (while
- 31 preserving any transparency of the fill or the stroked border themselves), and
- 32 the resulting figure is drawn onto the background using the specified opacity
- 33 and opacity mask (see Example 18–3 on page 285).

#### 34 **18.5.1 Optimization Guidelines**

35 The composition rules above describe the behavior of an ideal implementation.  
36 Practical implementations can optimize the processing of the composition rules  
37 according to the following guidelines:

- 38 1. If all elements on a canvas and the canvas itself are opaque (an opacity of
- 39 1.0) and parent or ancestor <Canvas> elements are also opaque, the
- 40 elements MAY be drawn directly to the containing fixed page (or canvas),
- 41 provided all render transform and clip values are observed [O11.12].
- 42 2. If an element is fully transparent (an opacity of 0.0), it MAY be skipped
- 43 [O11.13].

- 1 3. If a canvas has an opacity of 0.0, it and all of its child and descendant  
2 elements MAY be skipped [O11.14].
- 3 4. If a canvas has a Clip property with no contained area, the canvas and all of  
4 its child and descendant elements MAY be skipped [O11.15].
- 5 5. When creating a temporary surface, a consumer MAY further restrict the size  
6 of the temporary surface by the effective extent of the geometry specified by  
7 the Clip property of the canvas [O11.16].
- 8 6. A consumer MAY use methods to achieve transparency other than creating a  
9 temporary surface [O11.17]. Such methods MAY include planar mapping (that  
10 is, computation of intersections of transparent elements and resulting colors)  
11 [O11.17].

### 12 18.5.2 Composition Examples

13 The following examples illustrate the composition rules described above.

14 *Example 18-1. Path opacity behavior for overlapping path figures*

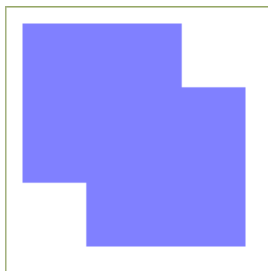
15 In the following markup, opacity is not applied cumulatively to self-overlapping areas  
16 created by path figures within the same path.

```

17 <Path Opacity="0.5">
18   <Path.Fill>
19     <SolidColorBrush Color="#0000FF" />
20   </Path.Fill>
21   <Path.Data>
22     <PathGeometry FillRule="NonZero">
23       <PathFigure StartPoint="10,10" IsClosed="true">
24         <PolyLineSegment Points="110,10 110,110 10,110 10,10" />
25       </PathFigure>
26       <PathFigure StartPoint="50,50" IsClosed="true">
27         <PolyLineSegment Points="150,50 150,150 50,150 50,50" />
28       </PathFigure>
29     </PathGeometry>
30   </Path.Data>
31 </Path>

```

32 This markup is rendered as follows:



33

34 *end example]*

1 *Example 18-2. Opacity behavior of path stroke intersections*

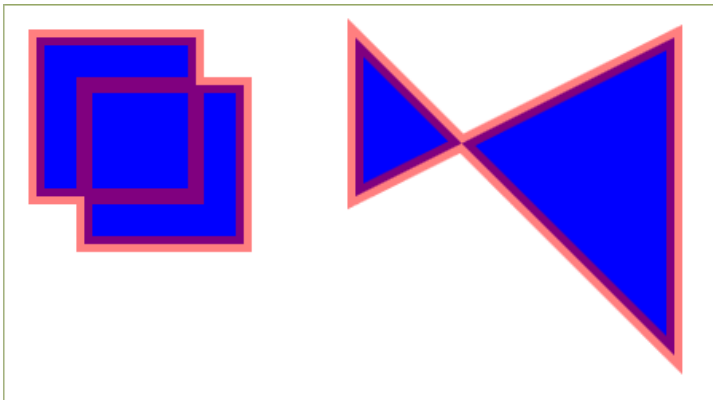
2 In the following markup, opacity is not applied cumulatively if the border of a path  
3 has self-intersections.

```

4 <Path Stroke="#80FF0000" StrokeThickness="10">
5   <Path.Fill>
6     <SolidColorBrush Color="#0000FF" />
7   </Path.Fill>
8   <Path.Data>
9     <PathGeometry FillRule="NonZero">
10      <PathFigure StartPoint="20,20" IsClosed="true">
11        <PolyLineSegment Points="120,20 120,120 20,120 20,20" />
12      </PathFigure>
13      <PathFigure StartPoint="50,50" IsClosed="true">
14        <PolyLineSegment Points="150,50 150,150 50,150 50,50" />
15      </PathFigure>
16    </PathGeometry>
17  </Path.Data>
18 </Path>
19 <Path Stroke="#80FF0000" StrokeThickness="10" StrokeMiterLimit="10">
20   <Path.Fill>
21     <SolidColorBrush Color="#0000FF" />
22   </Path.Fill>
23   <Path.Data>
24     <PathGeometry>
25      <PathFigure StartPoint="220,20" IsClosed="true">
26        <PolyLineSegment Points="420,220 420,20 220,120" />
27      </PathFigure>
28    </PathGeometry>
29  </Path.Data>
30 </Path>

```

31 This markup is rendered as follows:



32

33 *end example]*

34 *Example 18-3. Opacity behavior of paths with stroked edges*

35 The following markup describes a path with a stroked border and an opacity of less  
36 than 1.0:

```

37 <Path>
38   <Path.Fill>

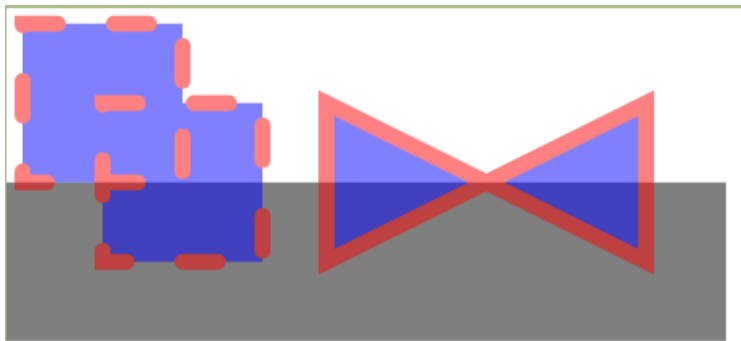
```

```

1      <SolidColorBrush Color="#7F7F7F" />
2      </Path.Fill>
3      <Path.Data>
4          <PathGeometry>
5              <PathFigure StartPoint="0,110" IsClosed="true">
6                  <PolyLineSegment Points="450,110 450,210 0,210" />
7              </PathFigure>
8          </PathGeometry>
9      </Path.Data>
10     </Path>
11     <Path
12         Stroke="#FF0000"
13         StrokeThickness="10"
14         StrokeDashArray="2.2 3.5"
15         StrokeDashCap="Round"
16         Opacity="0.5">
17         <Path.Fill>
18             <SolidColorBrush Color="#0000FF" />
19         </Path.Fill>
20         <Path.Data>
21             <PathGeometry FillRule="NonZero">
22                 <PathFigure StartPoint="10,10" IsClosed="true">
23                     <PolyLineSegment Points="110,10 110,110 10,110 10,10" />
24                 </PathFigure>
25                 <PathFigure StartPoint="60,60" IsClosed="true">
26                     <PolyLineSegment Points="160,60 160,160 60,160 60,60" />
27                 </PathFigure>
28             </PathGeometry>
29         </Path.Data>
30     </Path>
31     <Path Stroke="#FF0000" StrokeThickness="10" Opacity="0.5">
32         <Path.Fill>
33             <SolidColorBrush Color="#0000FF" />
34         </Path.Fill>
35         <Path.Data>
36             <PathGeometry>
37                 <PathFigure StartPoint="200,60" IsClosed="true">
38                     <PolyLineSegment Points="400,160 400,60 200,160" />
39                 </PathFigure>
40             </PathGeometry>
41         </Path.Data>
42     </Path>

```

43 This markup is rendered as follows:



44



1 *end example]*

---

## 18.6 Stroke Rendering

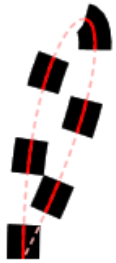
Strokes follow the contours of each segment in a path figure, as specified by the various stroke-related properties.

Contours and dashes SHOULD be rendered so that they have the same appearance as if rendered by sweeping the complete length of the contour or dash with a line segment that is perpendicular to the contour and extends with half its length to each side of the contour. All points covered by the sweep of this perpendicular line are part of the dash or contour [S11.21].

By using this sweeping definition, extreme curvatures may result in line and dash ends that are not flat when specified as flat. If any caps other than flat are specified, the caps are added to the start and end of the stroked contour or dash in the orientation of the first and last position of the line segment used for sweeping. Any render transform is applied after this step.

[Note: Using this definition, any geometry that is less than the value of the stroke thickness across will produce a filled area between these lines if no dashes are employed, or overlapping dashes when they are. *end note*]

Figure 18-2. Extreme curvatures and dash rendering



### 18.6.1 Stroke Edge Parallelization

Consumers SHOULD ensure that parallel edges of strokes appear parallel [S11.22]. Consumers can choose a suitable method to achieve this goal. [Example: Such methods might include anti-aliasing, sub-pixel masking, or appropriate rounding of device coordinates. *end example*]

### 18.6.2 Phase Control

Consumers SHOULD produce a visually consistent appearance of stroke thickness for thin lines, regardless of their orientation or how they fit on the device pixel grid [S11.23].

### 18.6.3 Symmetry of Stroke Drawing Algorithms

Consumers SHOULD select line and curve drawing algorithms that behave symmetrically and result in the same set of device pixels being drawn regardless of the direction of the line or curve (start point and end point exchanged) [S11.24]. In

1 other words, a line from 0,0 to 102,50 should result in the same pixel set as a line  
2 from 102,50 to 0,0.

### 3 **18.6.4 Rules for Dash Cap Rendering**

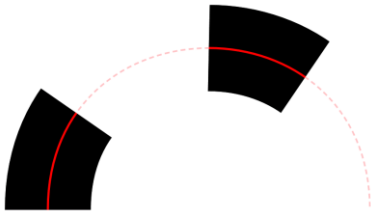
4 The appearance of dash caps is controlled by the **StrokeDashCap** attribute. Valid  
5 values are Flat, Square, Round, and Triangle.

#### 6 **18.6.4.1 Flat Dash Caps**

7 The effective render transform of the path being stroked is used to transform the  
8 control points of the contour of the dash.

9 The length of the dash is the approximate distance on the curve between the two  
10 intersections of the flat lines ending the dash and the contour of the shape. The  
11 distance from the end of one dash to the start of the next dash is the specified dash  
12 gap length. Dashes with a length greater than 0 are drawn, and degenerate dashes  
13 with a length of 0 are not drawn.

14 *Figure 18-3. Flat dash caps*



15

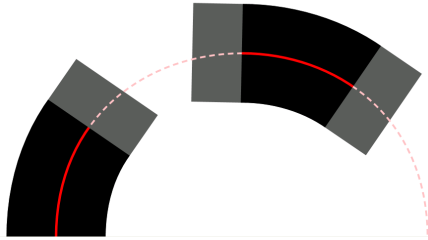
#### 16 **18.6.4.2 Square Dash Caps**

17 The effective render transform of the path being stroked is used to transform the  
18 control points of the contour of the dash.

19 The length of the dash is the approximate distance on the curve between the two  
20 contour intersection points, that is, the intersection of the flat line ending the dash  
21 (without the square caps attached) and the contour of the shape.

22 The caps are drawn as half-squares attached to the ends of the dash. The boundaries  
23 of the square caps are not curved to follow the contour, but are transformed using  
24 the effective render transform.

25 The distance between the contour intersection points of consecutive dashes is the  
26 specified dash gap length. Degenerate dashes with a length of 0 are drawn as  
27 squares.

1 *Figure 18-4. Square dash caps*

2

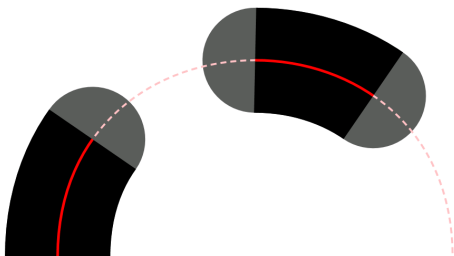
3 **18.6.4.3 Round Dash Caps**

4 The effective render transform of the path being stroked is used to transform the  
5 control points of the contour of the dash.

6 The length of the dash is the approximate distance on the curve between the two  
7 contour intersection points, that is, the intersection of the flat line ending the dash  
8 (without the round caps attached) and the contour of the shape.

9 The caps are drawn as half-circles attached to the ends of the dash. The boundaries  
10 of the round caps are not distorted to follow the contour, but are transformed using  
11 the effective render transform.

12 The distance between the contour intersection points of consecutive dashes is the  
13 specified dash gap length. Degenerate dashes with a length of 0 are drawn as circles.

14 *Figure 18-5. Round dash caps*

15

16 **18.6.4.4 Triangular Dash Caps**

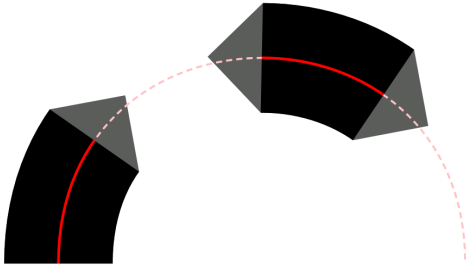
17 The effective render transform of the path being stroked is used to transform the  
18 control points of the contour of the dash.

19 The length of the dash is the approximate distance on the curve between the two  
20 contour intersection points, that is, the intersection of the flat line ending the dash  
21 (without the triangular caps attached) and the contour of the shape.

22 The caps are drawn as triangles attached with their base to the ends of the dash.  
23 The boundaries of the triangular caps are not distorted to follow the contour, but are  
24 transformed using the effective render transform. The height of the triangles is half  
25 of the stroke width.

1 The distance between the contour intersection points of consecutive dashes is the  
 2 specified dash gap length. Degenerate dashes with a length of 0 are drawn as  
 3 diamonds.

4 *Figure 18–6. Triangular dash caps*

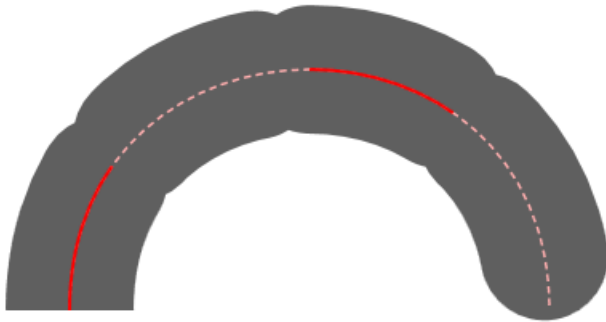


5

#### 6 **18.6.4.5 Overlapping Dashes**

7 It is possible to specify dash sequences with overlapping dash caps. In this  
 8 circumstance, the union of the dash segments (inclusive of dash caps), is used as a  
 9 mask through which the brush is applied as illustrated in Figure 18–7 with a stroke  
 10 dash cap value of Round.

11 *Figure 18–7. Overlapping dash segments*



12

#### 13 **18.6.5 Rules for Line Cap Rendering**

14 The appearance of line caps is controlled by the **StrokeStartLineCap** and  
 15 **StrokeEndLineCap** attribute. Valid values are Flat, Square, Triangle, and Round.  
 16 Every start line cap can be used in combination with any end line cap. Line caps are  
 17 not drawn for closed paths.

18 The rules for line caps on curved lines are analogous to the rules for dash cap  
 19 rendering. For more information, see §18.6, "Stroke Rendering," on page 288  
 20 and §18.6.4, "Rules for Dash Cap Rendering," on page 289.

1 *Figure 18-8. Flat start line cap, flat end line cap*



2

3 *Figure 18-9. Square start line cap, square end line cap*



4

5 *Figure 18-10. Triangular start line cap, triangular end line cap*



6

7 *Figure 18-11. Round start line cap, round end line cap*



8

9 **18.6.6 Line Caps for Dashed Strokes**

10 If the start point of a stroke is within a dash or touches the start or end of a dash, a  
 11 start line cap is appended to the stroke. Similarly, if the end point of a stroke is  
 12 within a dash or touches the start or end of a dash, an end line cap is appended to  
 13 the stroke.

14 *Figure 18-12. Stroke start or end point within a dash*



15

16 [Note: Because the right-most line cap begins at the point exactly coincident with  
 17 the start of the next dash in the sequence, it is rendered. *end note*]

18 However, if the start point of a stroke is within a gap (as can result from a  
 19 **StrokeDashOffset** attribute), no start line cap is appended to the stroke. If the end  
 20 point of a stroke is within a gap, no end line cap is appended to the stroke.

21 *Figure 18-13. Stroke start or end point within a gap*



22

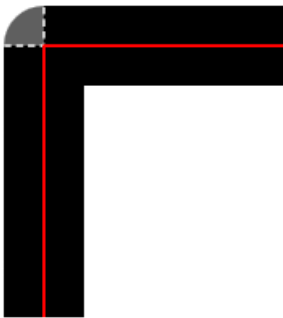
1 **18.6.7 Rules for Line Join Rendering**

2 The appearance of line joins is controlled by the **StrokeLineJoin** attribute. Valid  
3 values are Round, Bevel, and Miter.

4 **18.6.7.1 Round Line Joins**

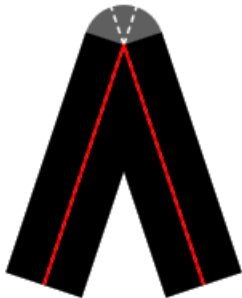
5 A **StrokeLineJoin** attribute value of Round indicates that the outer corner of the  
6 joined lines should be filled by enclosing the rounded region with its center point at  
7 the point of intersection between the two lines and a radius of one-half the stroke  
8 thickness value.

9 *Figure 18-14. Round line join with right angle*



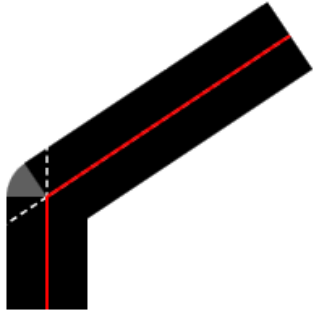
10

11 *Figure 18-15. Round line join with acute angle*



12

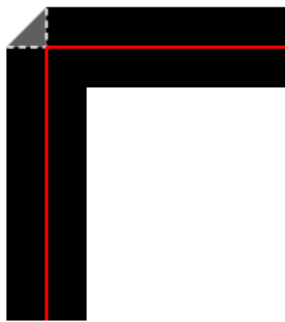
1 *Figure 18–16. Round line join with obtuse angle*



2  
3 **18.6.7.2 Beveled Line Joins**

4 A **StrokeLineJoin** attribute value of Bevel indicates that the outer corner of the  
5 joined lines should be filled by enclosing the triangular region of the corner with a  
6 straight line between the outer corners of each stroke.

7 *Figure 18–17. Beveled line join with right angle*

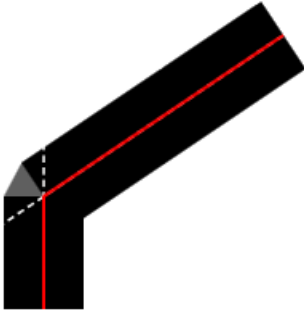


8  
9 *Figure 18–18. Beveled line join with acute angle*



10



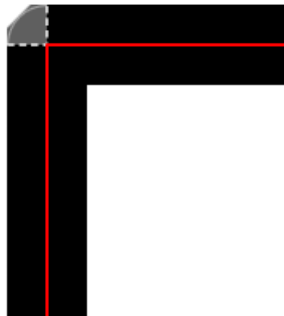
1 *Figure 18–19. Beveled line join with obtuse angle*

2

3 **18.6.7.3 Mitered Line Joins**

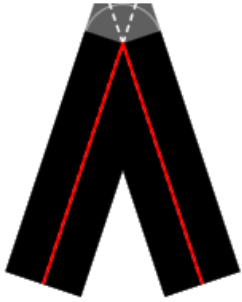
4 If the **StrokeLineJoin** attribute value is Miter, the value of the **StrokeMiterLimit**  
 5 attribute value is also used for rendering these joins. A **StrokeLineJoin** value of  
 6 Miter indicates that the region to be filled includes the intersection of the strokes  
 7 projected to infinity, and then clipped at a specific distance. The intersection of the  
 8 strokes is clipped at a line perpendicular to the bisector of the angle between the  
 9 strokes, at the distance equal to the stroke miter limit value multiplied by half the  
 10 stroke thickness value.

11 When drawing mitered line joins, the presence of one or more degenerate line  
 12 segments between the non-degenerate line segments to be joined results in a  
 13 mitered line join of only the two non-degenerate line segments with an implied  
 14 **StrokeMiterLimit** attribute value of 1.0.

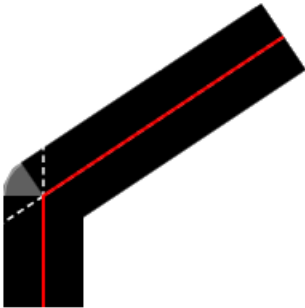
15 *Figure 18–20. Mitered line join with right angle and miter limit of 1.0*

16

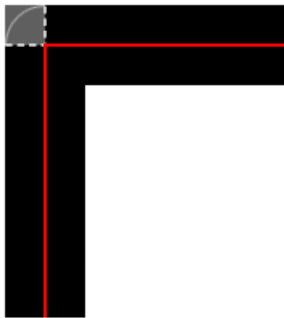
1 *Figure 18–21. Mitered line join with acute angle and miter limit of 1.0*



2  
3 *Figure 18–22. Mitered line join with obtuse angle and miter limit of 1.0*

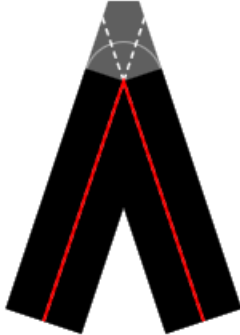


4  
5 *Figure 18–23. Mitered line join with right angle and miter limit of 2.0*



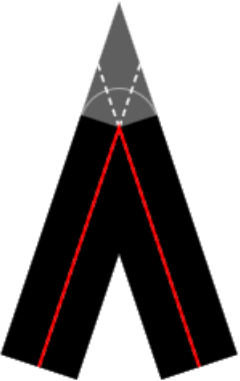
6

1 *Figure 18–24. Mitered line join with acute angle and miter limit of 2.0*



2

3 *Figure 18–25. Mitered line join with acute angle and miter limit of 10.0*



4

#### 5 **18.6.8 Rules for Degenerate Line and Curve Segments**

6 Degenerate line segments (that is, where the start point and end point coincide) are  
7 not drawn.

8 Degenerate curve segments (where the start point, end point, and all control points  
9 coincide) are not drawn, regardless of the stroke start line cap or stroke end line cap  
10 specified.

11 If an open degenerate path (formed from degenerate line or curve segments) with  
12 non-flat start cap and/or non-flat end line cap is stroked, only the start line cap  
13 and/or end line cap is drawn, in the  $x$  direction relative to the current effective  
14 render transform (that is, as if a segment were drawn from  $x,y$  to  $x+d,y$ , with  $d \rightarrow$   
15 0).

16 If a closed degenerate path (formed from degenerate line or curve segments) is  
17 stroked, a circular dot with a diameter of the stroke thickness is drawn instead.

1 If the current render transform is an invertible matrix, consumers SHOULD perform  
2 computations on poly line segments and poly Bézier segments with sufficient  
3 accuracy to avoid producing zero-length segments [S11.25].

#### 4 **18.6.9 Stroking and Fill Rule**

5 Stroking a path is independent of the fill rule. The fill rule affects the filled area only.

#### 6 **18.6.10 Mixing Stroked and Non-Stroked Segments**

7 When a path figure contains multiple segments and one or more of the segments has  
8 an **IsStroked** value of false, the phase for dashes starts anew with the next stroked  
9 segment, including application of the dash offset.

10 When a segment is stroked and the subsequent segment has an **IsStroked** value of  
11 false, thus causing a dash to be truncated, the dash cap is drawn for both ends of  
12 the truncated dash, exactly as it would for a non-truncated dash.

#### 13 **18.6.11 Stroke Behavior with Multiple Path Figures**

14 When a geometry containing multiple path figures is stroked, the phase for dashes  
15 (including application of the dash offsets) starts anew with each new path figure.

16 In general, for any path geometry, each path figure is drawn independently of every  
17 other path figure, so the dash array is reset for each. Dashes are also reset after  
18 every unstroked segment.

#### 19 **18.6.12 Consistent Nominal Stroke Width**

20 For certain scenarios, it is desirable for producers to generate documents targeted at  
21 specific aliasing consumers with particular lines in the document indicated as  
22 hairlines or consistent-width strokes. The following recommendation allows these  
23 producers and consumers to handle these strokes consistently.

24 Producers MAY generate a <Path> element intended to be treated as having a  
25 consistent nominal stroke width by specifying the **StrokeDashArray** attribute and  
26 by specifying a **StrokeDashOffset** attribute value less than -1.0 times the sum of all  
27 the numbers in the **StrokeDashArray** attribute value [O11.25].

28 For a solid line, the producer would set the **StrokeDashArray** to the value "1 0" and  
29 the **StrokeDashOffset** to a value such as "-2". The "-2" value fulfills the restriction  
30 on the **StrokeDashOffset** value in a numerically stable manner, and the phase of  
31 the dash pattern is identical to a **StrokeDashOffset** value of "0". Values less than "-  
32 2" can be used to specify a shifted phase of the dash pattern.

33 A stroke using the consistent nominal stroke width convention SHOULD be rendered  
34 with a width consistent with other strokes using the convention that have the same  
35 **StrokeThickness** attribute value, and consumers aware of this convention SHOULD  
36 render such a stroke no thinner than the thinnest visible line that consumer supports  
37 without dropouts [S11.31]. See section 11.1.4, "Pixel Center Location, Pixel  
38 Placement, and Pixel Inclusion", for further considerations for rendering thin lines.

---

## 18.7 Brushes and Images

Images require the following special considerations for scaling and tile placement.

### 18.7.1 Small Tiles

Tiles for visual brushes and image brushes may be specified with a viewport width or height of a few device pixels, or even less than a single device pixel in size.

If both width and height are nearly zero, implementations SHOULD average the color values of the brush contents, resulting in a constant-color brush [S11.26]. *[Example:*

- A visual brush or image brush that contains a blue and white checkerboard pattern results in a solid light-blue fill as either the width or the height value approaches 0.0.
- A visual brush or image brush whose viewbox is constant-colored produces a constant-colored brush regardless of the width and height values of the viewport.

*end example]*

If only one of the width and height values is nearly zero, the brush should be constant-colored along lines parallel to the narrow side of the viewport. For cases such as these, implementations MAY differ [O11.21]. Producers SHOULD avoid producing such extreme cases and SHOULD NOT rely on any specific behavior when they do [S11.27].

### 18.7.2 Image Scaling

Source sampling SHOULD be done from the center of the pixel and should be mapped to the center of the pixel in the device-space [S11.28]. With one extent of the viewbox zero, sampling SHOULD be done along a line parallel to the non-zero side [S11.28]. With both extents of the viewbox zero, a point sample SHOULD be taken [S11.28].

When up-sampling an image presented at a lower resolution than the device resolution, bilinear filtering SHOULD be used [S11.29]. The precise source coordinates as specified by the viewbox MUST be used to place the up-sampled image tile, which is equivalent to using fractional pixels of the original source image [M11.8].

When down-sampling, at least a bilinear filter SHOULD be used [S11.30]. Consumers MAY choose to implement a more sophisticated algorithm, such as a Fant scaler, to prevent aliasing artifacts [O11.22].

### 18.7.3 Tile Placement

Consumers MUST precisely position the tiles specified by the image brush and visual brush. If the specified values result in fractional device pixels, the consumer MUST calculate a running placement-error delta and adjust the placement of the next tile where the delta reaches a full device pixel in order to keep the tiles from being increasingly out of phase as the expanse of the path is filled [M11.9]. Consumers MAY choose any technique desired to achieve this requirement, such as linear filtering for seams, stretching of the tile (up-sampling or down-sampling), or pre-

1 computing multiple tiles and adjusting behavior according to how the tiles fit on a  
2 grid [O11.23].

### 3 **18.7.4 Tiling Transparent Visual Brushes and Image Brushes**

4 The contents of a visual brush's Visual property are first rendered to a temporary  
5 work canvas with an opacity of 0.0. The viewbox of the visual brush defines the tile  
6 or portion of the temporary canvas that is copied onto the specified geometry,  
7 stroke, or text. Likewise, an image specified by an image brush is also copied to a  
8 temporary work canvas. The viewbox also defines the tile for an image brush. In  
9 either case, the work canvas is scaled to properly match the edges of the tile to the  
10 size specified by the viewport.

11 Each pixel of the resultant tile is separately blended with the background of its  
12 destination, using the alpha of each pixel. This process is repeated for each tile  
13 replication, while respecting the **TileMode** attribute value, although the temporary  
14 work canvas MAY be re-used [O11.24].

15

1 **19. Elements**

2 **ArcSegment**

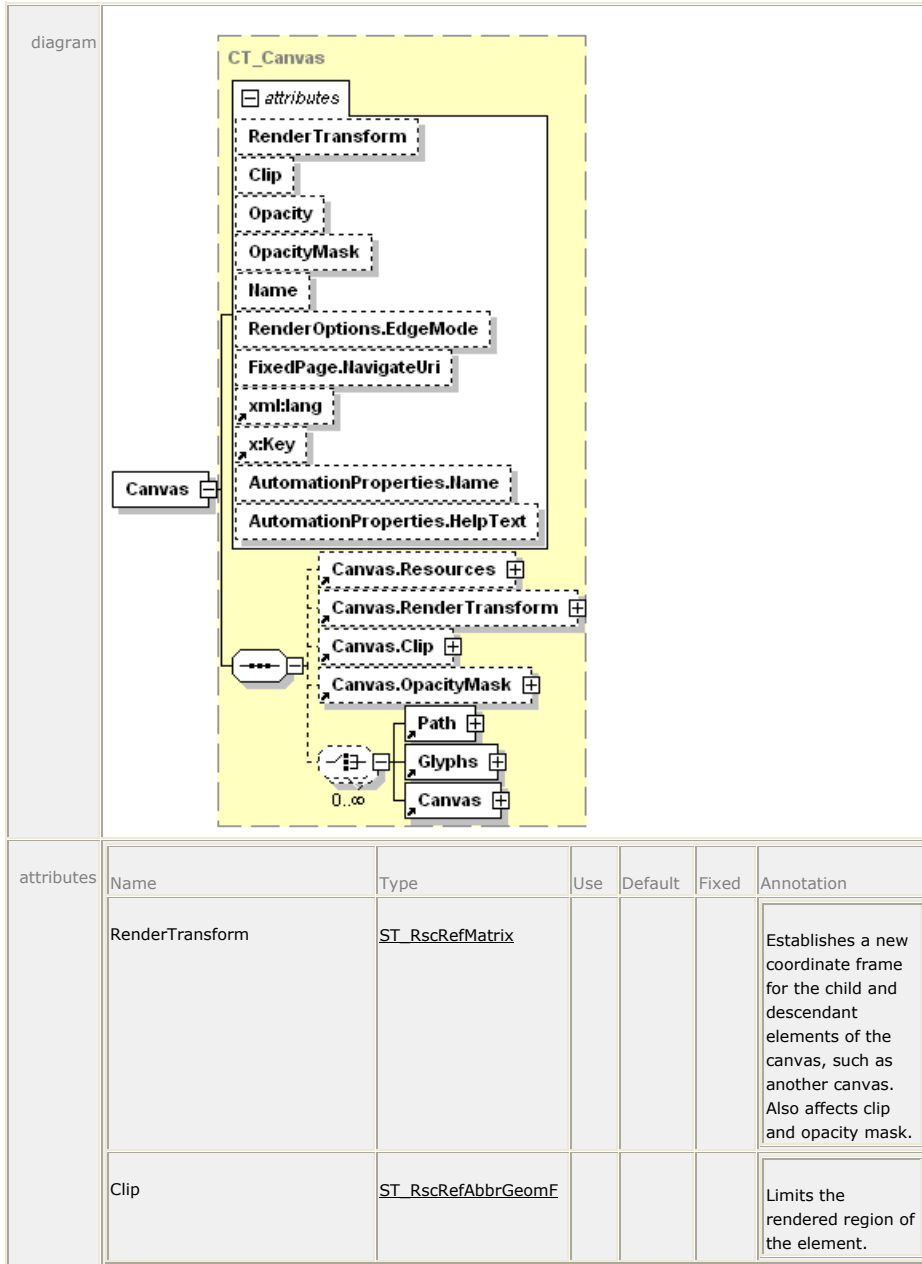
3 element **ArcSegment**

diagram						
attributes	Name	Type	Use	Default	Fixed	Annotation
	Point	<a href="#">ST_Point</a>	required			Specifies the endpoint of the elliptical arc.
	Size	<a href="#">ST_PointGE0</a>	required			Specifies the x and y radius of the elliptical arc as an x,y pair.
	RotationAngle	<a href="#">ST_Double</a>	required			Indicates how the ellipse is rotated relative to the current coordinate system.
	IsLargeArc	<a href="#">ST_Boolean</a>	required			Determines whether the arc is drawn with a sweep of 180 or greater. Can be true or false.
	SweepDirection	<a href="#">ST_SweepDirection</a>	required			Specifies the direction in which the arc is drawn. Valid values are Clockwise and Counterclockwise.
	IsStroked	<a href="#">ST_Boolean</a>		true		Specifies whether the stroke for this segment of the path is drawn. Can be true or false.
annotation	Represents an elliptical arc between two points.					

4

1 **Canvas**

2 element **Canvas**





Opacity	<a href="#">ST_ZeroOne</a>	1.0		Defines the uniform transparency of the canvas. Values range from 0 (fully transparent) to 1 (fully opaque), inclusive. Values outside of this range are invalid.
OpacityMask	<a href="#">ST_RscRef</a>			Specifies a mask of alpha values that is applied to the canvas in the same fashion as the Opacity attribute, but allowing different alpha values for different areas of the element.
Name	<a href="#">ST_Name</a>			Contains a string value that identifies the current element as a named, addressable point in the document for the purpose of hyperlinking.
RenderOptions.EdgeMode	<a href="#">ST_EdgeMode</a>			Controls how edges of paths within the canvas are rendered. The only valid value is Aliased. Omitting this attribute causes the edges to be rendered in the consumer's default manner.
FixedPage.NavigateUri	xs:anyURI			Associates a hyperlink URI with the element. May be a relative reference or a URI that addresses a resource that is

					internal to or external to the package.
xml:lang					Specifies the default language used for the current element and for any child or descendant elements. The language is specified according to RFC 3066.
x:Key					Specifies a name for a resource in a resource dictionary. x:Key MUST be present when the current element is defined in a resource dictionary. x:Key MUST NOT be specified outside of a resource dictionary [M3.20].
AutomationProperties.Name	xs:string				A brief description of the <Canvas> contents for accessibility purposes, particularly if filled with a set of vector graphics and text elements intended to comprise a single vector graphic.
AutomationProperties.HelpText	xs:string				A detailed description of the <Canvas> contents for accessibility purposes, particularly if filled with a set of graphics and text elements intended

						to comprise a single vector graphic.
annotation	Groups <FixedPage> descendant elements together.					

1 For more information, see §10.4, "<Canvas> Element," on page 61.

2 **Canvas.Clip**

3 element **Canvas.Clip**

diagram	
annotation	Limits the rendered region of the element.

4 For more information, see §14.3, "Clipping," on page 182.

5 **Canvas.OpacityMask**

6 element **Canvas.OpacityMask**

diagram	
annotation	Specifies a mask of alpha values that is applied to the canvas in the same fashion as the Opacity attribute, but allowing different alpha values for different areas of the element.

7 For more information, see §14.5.1, "<Canvas.OpacityMask> Element," on page 200.

1 **Canvas.RenderTransform**

2 element **Canvas.RenderTransform**

diagram	
annotation	Establishes a new coordinate frame for the child and descendant elements of the canvas, such as another canvas. Also affects clip and opacity mask.

3 For more information, see §14.4, "Positioning Content," on page 185.

4 **Canvas.Resources**

5 element **Canvas.Resources**

diagram	
annotation	Contains the resource dictionary for the <Canvas> element.

6 For more information, see §14.2, "Resources and Resource References," on page 172.

8 **Discard**

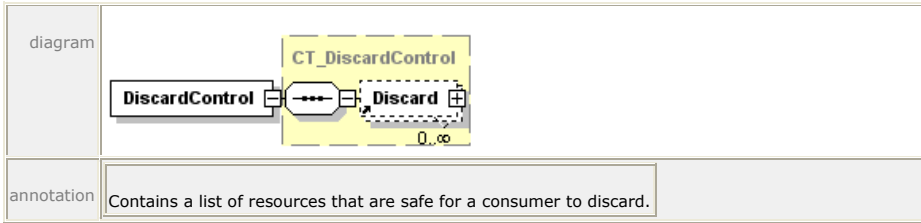
9 element **Discard**

diagram																			
attributes	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> <th>Annotation</th> </tr> </thead> <tbody> <tr> <td>SentinelPage</td> <td>xs:anyURI</td> <td>required</td> <td></td> <td></td> <td>The first fixed page that no longer needs the identified resource in order to be processed.</td> </tr> <tr> <td>Target</td> <td>xs:anyURI</td> <td>required</td> <td></td> <td></td> <td>The resource that can be safely discarded.</td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	Annotation	SentinelPage	xs:anyURI	required			The first fixed page that no longer needs the identified resource in order to be processed.	Target	xs:anyURI	required			The resource that can be safely discarded.
Name	Type	Use	Default	Fixed	Annotation														
SentinelPage	xs:anyURI	required			The first fixed page that no longer needs the identified resource in order to be processed.														
Target	xs:anyURI	required			The resource that can be safely discarded.														
annotation	Identifies a resource that can be safely discarded by a resource-constrained consumer.																		

10 For more information, see §17.1.4.1.2, "<Discard> Element."

1 **DiscardControl**

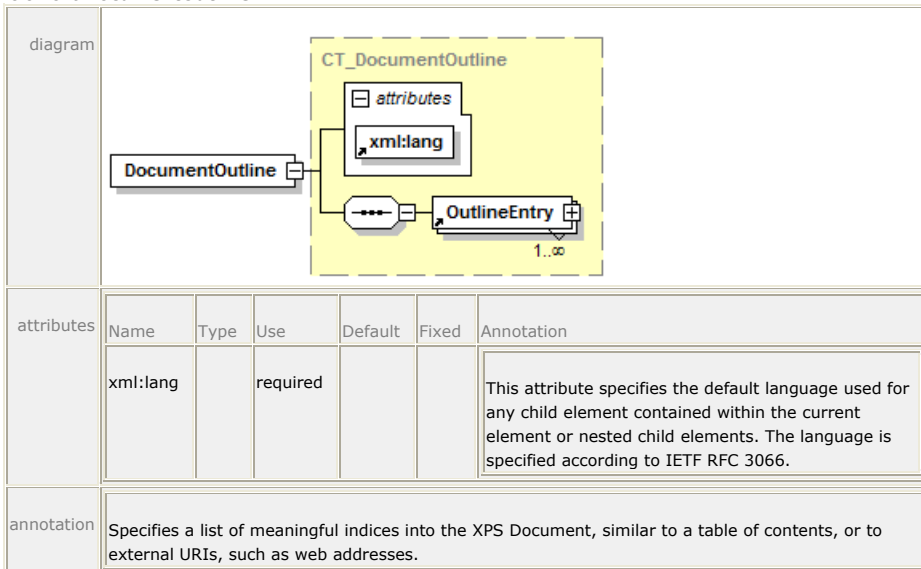
2 element **DiscardControl**



3 For more information, see §17.1.4.1.1, “<DiscardControl> Element.”

4 **DocumentOutline**

5 element **DocumentOutline**



6 For more information, see §16.1.1.3, “<DocumentOutline> Element,” on page 226.

1 **DocumentReference**

2 element **DocumentReference**

diagram						
attributes	Name	Type	Use	Default	Fixed	Annotation
	Source	xs:anyURI	required			Specifies the URI of the fixed document content. The specified URI MUST refer to a FixedDocument part within the XPS Document [M3.2].
annotation	Contains a reference to a FixedDocument part.					

3 For more information, see §10.1.1, "<DocumentReference," on page 49.

4 **DocumentStructure**

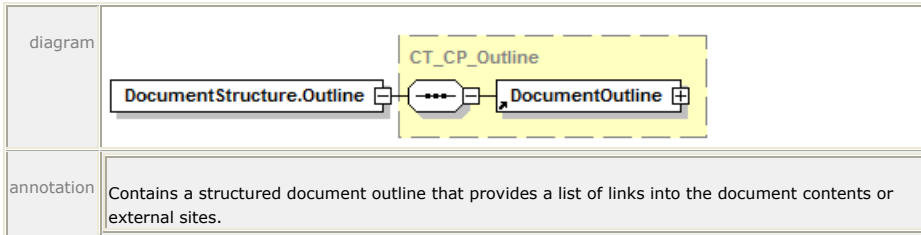
5 element **DocumentStructure**

diagram						
annotation	The root element of the DocumentStructure part.					

6 For more information, see §16.1.1.1, "<DocumentStructure> Element," on page  
 7 224.

1 **DocumentStructure.Outline**

2 element **DocumentStructure.Outline**

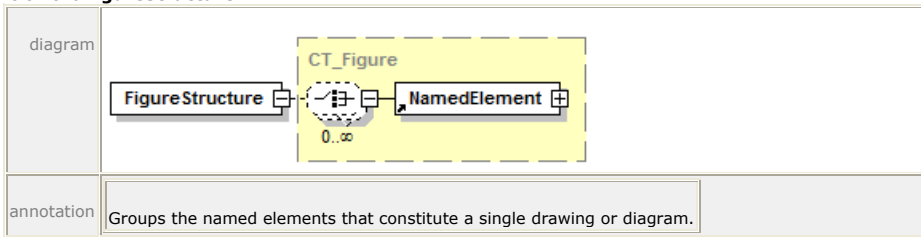


3

4 For more information see §16.1.1.2, "<DocumentStructure.Outline> Element," on  
5 page 226.

6 **FigureStructure**

7 element **FigureStructure**



8 For more information, see §16.1.2.12, "<FigureStructure> Element," on page 242.

9 **FixedDocument**

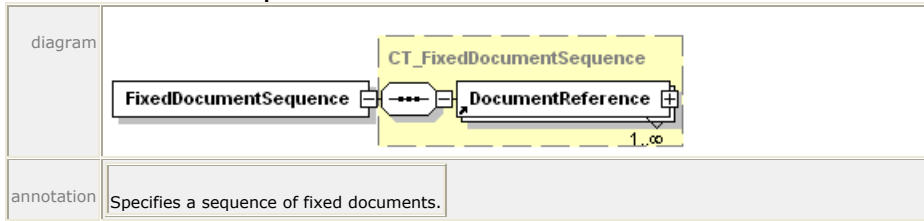
10 element **FixedDocument**



11 For more information, see §10.2, "<FixedDocument> Element," on page 50.

1 **FixedDocumentSequence**

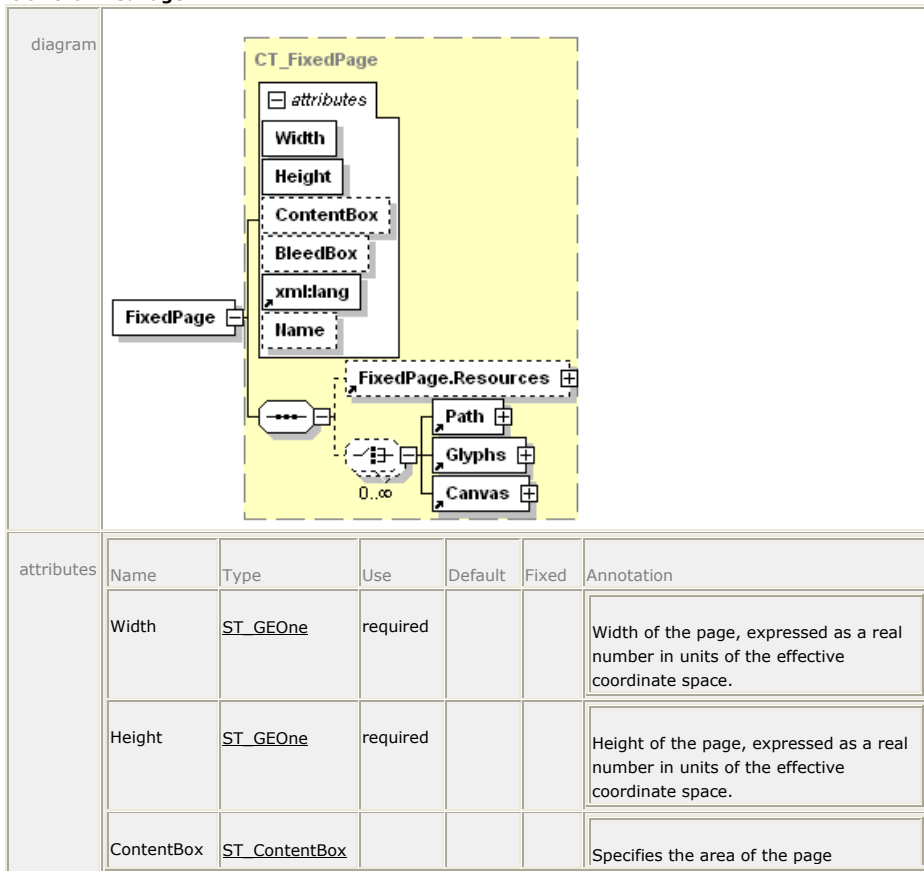
2 element **FixedDocumentSequence**



3 For more information, see §10.1, "<FixedDocumentSequence> Element," on  
 4 page 49.

5 **FixedPage**

6 element **FixedPage**

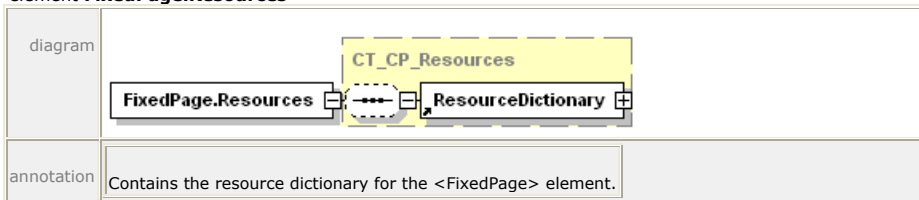




					containing imageable content that is to be fit within the imageable area when printing or viewing. Contains a list of four coordinate values (ContentOriginX, ContentOriginY, ContentWidth, ContentHeight), expressed as comma-separated real numbers. Specifying a value is RECOMMENDED [S3.1]. If omitted, the default value is (0,0,Width,Height).
BleedBox	<u>ST_BleedBox</u>				Specifies the area including crop marks that extends outside of the physical page. Contains a list of four coordinate values (BleedOriginX, BleedOriginY, BleedWidth, BleedHeight), expressed as comma-separated real numbers. If omitted, the default value is (0,0,Width,Height).
xml:lang		required			Specifies the default language used for the current element and for any child or descendant elements. The language is specified according to RFC 3066.
Name	<u>ST_Name</u>				Contains a string value that identifies the current element as a named, addressable point in the document for the purpose of hyperlinking.
annotation	Contains markup that describes the rendering of a single page of content.				

1 For more information, see §10.3, "<FixedPage> Element," on page 53.

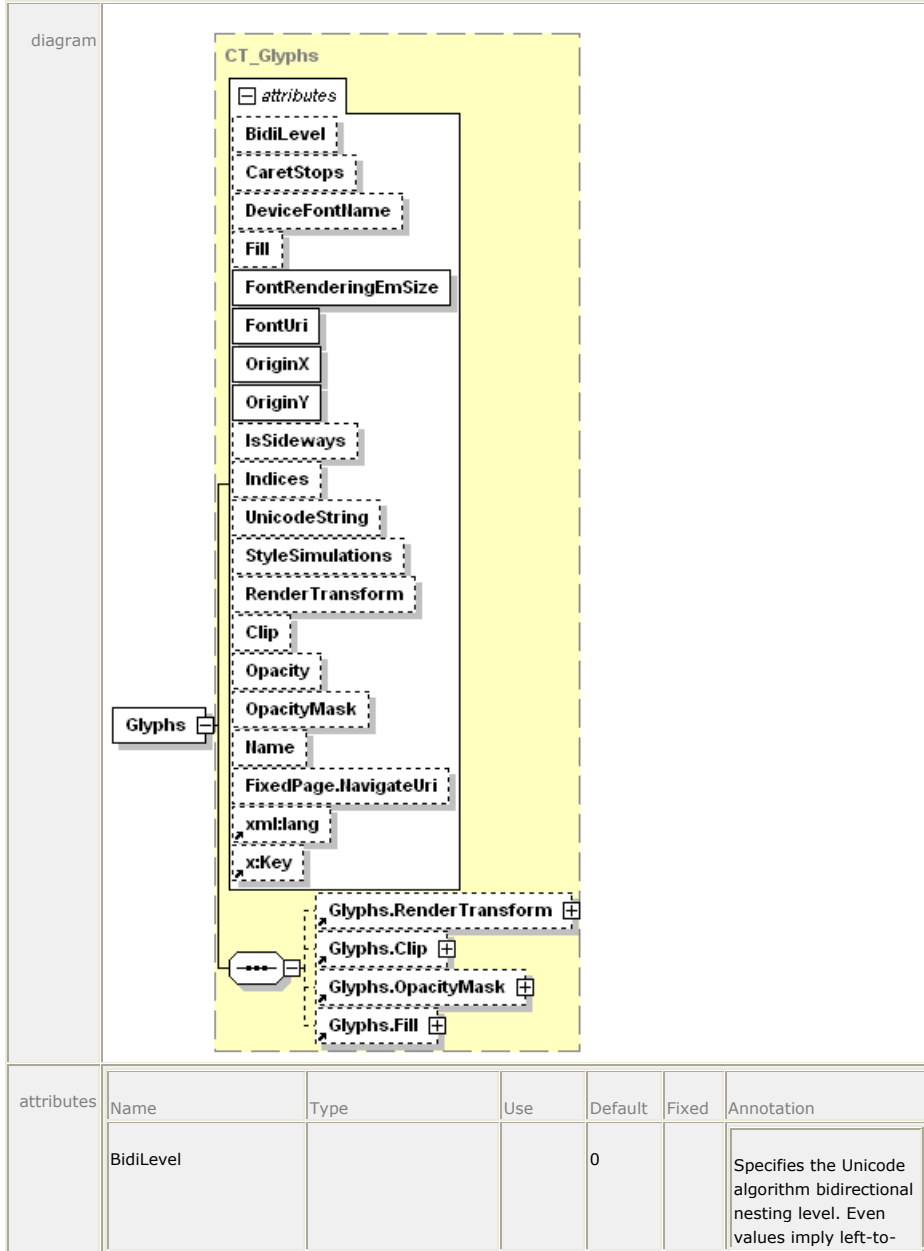
2 **FixedPage.Resources**  
 3 element **FixedPage.Resources**



4 For more information, see §14.2, "Resources and Resource References," on  
 5 page 172.

**Glyphs**

element **Glyphs**



					right layout, odd values imply right-to-left layout. Right-to-left layout places the run origin at the right side of the first glyph, with positive advance widths (representing advances to the left) placing subsequent glyphs to the left of the previous glyph. Valid values range from 0 to 61, inclusive.
CaretStops	<u>ST_CaretStops</u>				Identifies the positions within the sequence of Unicode characters at which a text-selection tool may place a text-editing caret. Potential caret-stop positions are identified by their indices into the UTF-16 code units represented by the UnicodeString attribute value. When this attribute is missing, the text in the UnicodeString attribute value MUST be interpreted as having a caret stop between every Unicode UTF-16 code unit and at the beginning and end of the text [M5.1]. The value SHOULD indicate that the caret cannot stop in front of most combining marks or in front of the second UTF-16 code unit of UTF-16 surrogate pairs [S5.1].
DeviceFontName	<u>ST_UnicodeString</u>				Uniquely identifies a specific device font. The identifier is typically defined by a hardware vendor or

					font vendor.
Fill	<a href="#">ST_RscRefColor</a>				Describes the brush used to fill the shape of the rendered glyphs.
FontRenderingEmSize	<a href="#">ST_GEZero</a>	required			Specifies the font size in drawing surface units, expressed as a float in units of the effective coordinate space. A value of 0 results in no visible text.
FontUri	xs:anyURI	required			The URI of the physical font from which all glyphs in the run are drawn. The URI MUST reference a font contained in the package [M2.1]. If the physical font referenced is a TrueType Collection (containing multiple font faces), the fragment portion of the URI is a 0-based index indicating which font face of the TrueType Collection should be used.
OriginX	<a href="#">ST_Double</a>	required			Specifies the x coordinate of the first glyph in the run, in units of the effective coordinate space. The glyph is placed so that the leading edge of its advance vector and its baseline intersect with the point defined by the OriginX and OriginY attributes.
OriginY	<a href="#">ST_Double</a>	required			Specifies the y coordinate of the first glyph in the run, in units of the effective

					coordinate space. The glyph is placed so that the leading edge of its advance vector and its baseline intersect with the point defined by the OriginX and OriginY attributes.
IsSideways	<a href="#">ST_Boolean</a>		false		Indicates that a glyph is turned on its side, with the origin being defined as the top center of the unturned glyph.
Indices	<a href="#">ST_Indices</a>				Specifies a series of glyph indices and their attributes used for rendering the glyph run. If the UnicodeString attribute specifies an empty string (" or "{}") and the Indices attribute is not specified or is also empty, a consumer MUST generate an error [M5.2].
UnicodeString	<a href="#">ST_UnicodeString</a>				Contains the string of text rendered by the <Glyphs> element. The text is specified as Unicode code points.
StyleSimulations	<a href="#">ST_StyleSimulations</a>		None		Specifies a style simulation. Valid values are None, ItalicSimulation, BoldSimulation, and BoldItalicSimulation.
RenderTransform	<a href="#">ST_RscRefMatrix</a>				Establishes a new coordinate frame for the glyph run specified by the <Glyphs> element. The render transform affects clip, opacity mask, fill, x origin, y origin, the

					actual shape of individual glyphs, and the advance widths. The render transform also affects the font size and values specified in the Indices attribute.
Clip	<a href="#">ST_RscRefAbbrGeomF</a>				Limits the rendered region of the element. Only portions of the <Glyphs> element that fall within the clip region (even partially clipped characters) produce marks on the page.
Opacity	<a href="#">ST_ZeroOne</a>		1.0		Defines the uniform transparency of the glyph element. Values range from 0 (fully transparent) to 1 (fully opaque), inclusive. Values outside of this range are invalid.
OpacityMask	<a href="#">ST_RscRef</a>				Specifies a mask of alpha values that is applied to the glyphs in the same fashion as the Opacity attribute, but allowing different alpha values for different areas of the element.
Name	<a href="#">ST_Name</a>				Contains a string value that identifies the current element as a named, addressable point in the document for the purpose of hyperlinking.
FixedPage.NavigateUri	xs:anyURI				Associates a hyperlink URI with the element. May be a relative reference or a URI that addresses a resource that is

					internal to or external to the package.
	xml:lang				Specifies the default language used for the current element. The language is specified according to RFC 3066.
	x:Key				Specifies a name for a resource in a resource dictionary. x:Key MUST be present when the current element is defined in a resource dictionary. x:Key MUST NOT be specified outside of a resource dictionary [M5.3].
annotation	Represents a run of text from a single font.				

- 1 For more information, see:
- 2     • Section 12.1, "<Glyphs> Element," on page 98
- 3     • Section 9.1.7, "Font Parts," on page 31
- 4     • Section 12.1.3, "Indices Attribute," on page 109

5 **Glyphs.Clip**

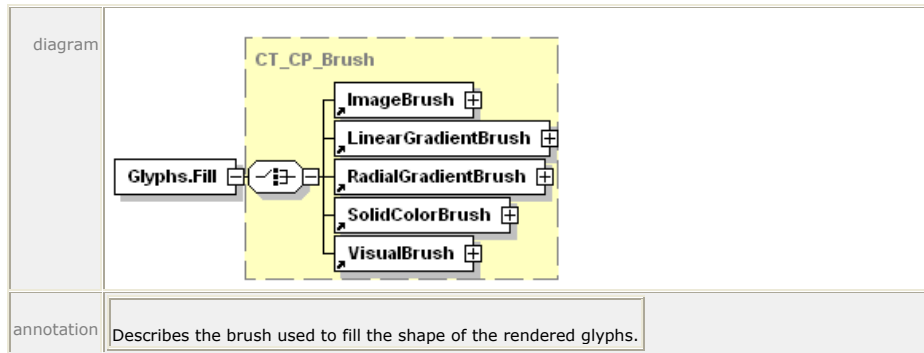
6 element **Glyphs.Clip**

diagram	<p>The diagram illustrates the structure of the Glyphs.Clip element. It shows a box labeled 'Glyphs.Clip' containing a dashed-line rectangle labeled 'CT_CP_Geometry' (the clip region) and a box labeled 'PathGeometry' (the content to be clipped). A dashed arrow points from the clip region to the PathGeometry box, indicating that the content is clipped to fit within the region.</p>
annotation	Limits the rendered region of the element. Only portions of the <Glyphs> element that fall within the clip region (even partially clipped characters) produce marks on the page.

- 7 For more information, see §14.3, "Clipping," on page 182.

8 **Glyphs.Fill**

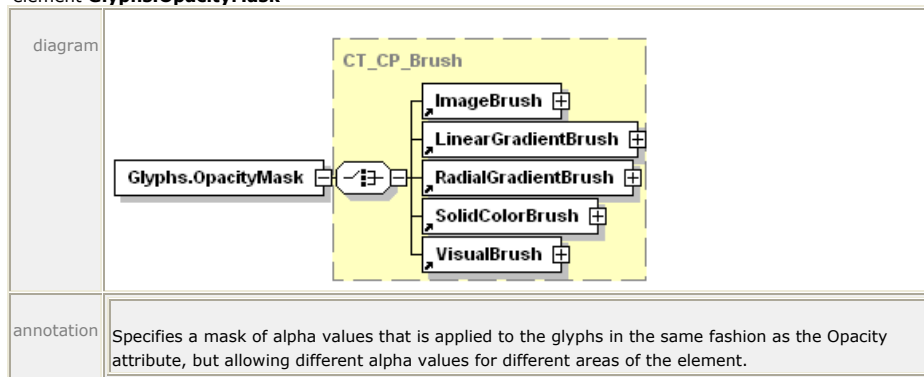
9 element **Glyphs.Fill**



1 For more information, see §12.2, “<Glyphs.Fill> Element.”

## 2 **Glyphs.OpacityMask**

3 element **Glyphs.OpacityMask**



4 For more information, see §14.5.3, “<Glyphs.OpacityMask> Element,” on page 202.



1 **Glyphs.RenderTransform**

2 element **Glyphs.RenderTransform**

diagram	
annotation	<p>Establishes a new coordinate frame for the glyph run specified by the &lt;Glyphs&gt; element. The render transform affects clip, opacity mask, fill, x origin, y origin, the actual shape of individual glyphs, and the advance widths. The render transform also affects the font size and values specified in the Indices attribute.</p>

3 For more information, see §14.4, "Positioning Content," on page 185.

4 **GradientStop**

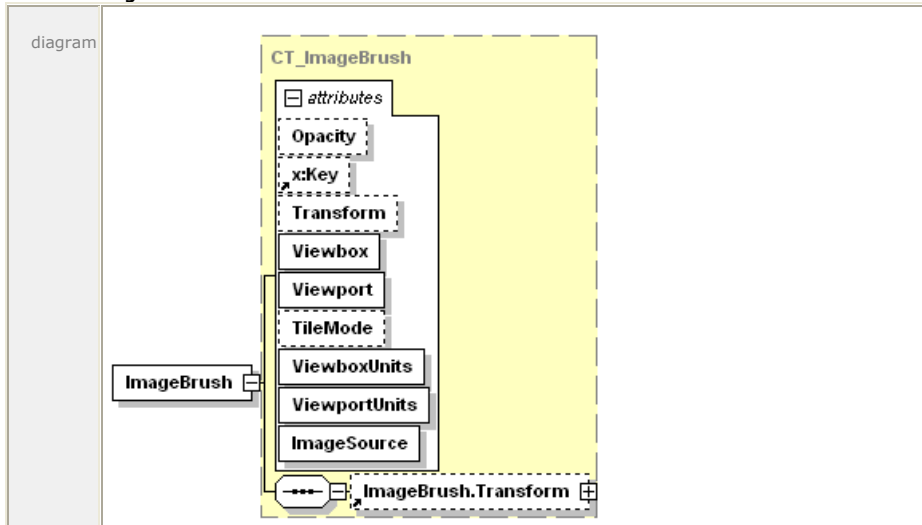
5 element **GradientStop**

diagram																			
attributes	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> <th>Annotation</th> </tr> </thead> <tbody> <tr> <td>Color</td> <td><a href="#">ST_Color</a></td> <td>required</td> <td></td> <td></td> <td>Specifies the gradient stop color. An sRGB color value specified as a 6-digit hexadecimal number (#RRGGBB) or an extended color.</td> </tr> <tr> <td>Offset</td> <td><a href="#">ST_Double</a></td> <td>required</td> <td></td> <td></td> <td>Specifies the gradient offset. The offset indicates a point along the progression of the gradient at which a color is specified. Colors between gradient offsets in the progression are interpolated.</td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	Annotation	Color	<a href="#">ST_Color</a>	required			Specifies the gradient stop color. An sRGB color value specified as a 6-digit hexadecimal number (#RRGGBB) or an extended color.	Offset	<a href="#">ST_Double</a>	required			Specifies the gradient offset. The offset indicates a point along the progression of the gradient at which a color is specified. Colors between gradient offsets in the progression are interpolated.
Name	Type	Use	Default	Fixed	Annotation														
Color	<a href="#">ST_Color</a>	required			Specifies the gradient stop color. An sRGB color value specified as a 6-digit hexadecimal number (#RRGGBB) or an extended color.														
Offset	<a href="#">ST_Double</a>	required			Specifies the gradient offset. The offset indicates a point along the progression of the gradient at which a color is specified. Colors between gradient offsets in the progression are interpolated.														
annotation	<p>Indicates a location and range of color progression for rendering a gradient.</p>																		

6 For more information, see §13.7, "<GradientStop> Element," on page 166.

1 **ImageBrush**

2 element **ImageBrush**



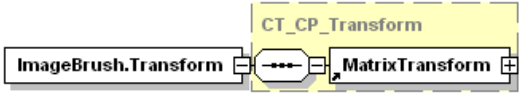
attributes						
Name	Type	Use	Default	Fixed	Annotation	
Opacity	<u>ST_ZeroOne</u>		1.0		Defines the uniform transparency of the brush fill. Values range from 0 (fully transparent) to 1 (fully opaque), inclusive. Values outside of this range are invalid.	
x:Key					Specifies a name for a resource in a resource dictionary. x:Key MUST be present when the current element is defined in a resource dictionary. x:Key MUST NOT be specified outside of a resource dictionary [M6.2].	
Transform	<u>ST_RscRefMatrix</u>				Describes the matrix transformation applied to the coordinate space of the brush. The Transform property is concatenated with the current effective render transform to yield an effective render transform local to the brush. The viewport for the brush is transformed using the local effective render transform.	

	Viewbox	<a href="#">ST_ViewBox</a>	required			Specifies the position and dimensions of the brush's source content. Specifies four comma-separated real numbers (x, y, Width, Height), where width and height are non-negative. The dimensions specified are relative to the image's physical dimensions expressed in units of 1/96". The corners of the viewbox are mapped to the corners of the viewport, thereby providing the default clipping and transform for the brush's source content.
	Viewport	<a href="#">ST_ViewBox</a>	required			Specifies the region in the containing coordinate space of the prime brush tile that is (possibly repeatedly) applied to fill the region to which the brush is applied. Specifies four comma-separated real numbers (x, y, Width, Height), where width and height are non-negative. The alignment of the brush pattern is controlled by adjusting the x and y values.
	TileMode	<a href="#">ST_TileMode</a>		None		Specifies how contents will be tiled in the filled region. Valid values are None, Tile, FlipX, FlipY, and FlipXY.
	ViewboxUnits	<a href="#">ST_ViewUnits</a>	required		Absolute	Specifies the relationship of the viewbox coordinates to the containing coordinate space.
	ViewportUnits	<a href="#">ST_ViewUnits</a>	required		Absolute	Specifies the relationship of the viewport coordinates to the containing coordinate space.
	ImageSource	<a href="#">ST_UriCtxBmp</a>	required			Specifies the URI of an image resource or a combination of the URI of an image resource a color profile resource. See the Color clause for important details. The URI MUST refer to parts in the package [M2.1].
annotation	Fills a region with an image.					

1 For more information, see §13.2, “<ImageBrush> Element,” on page 127.

2 **ImageBrush.Transform**


3 element **ImageBrush.Transform**

diagram	
annotation	<p>Describes the matrix transformation applied to the coordinate space of the brush. The Transform property is concatenated with the current effective render transform to yield an effective render transform local to the brush. The viewport for the brush is transformed using the local effective render transform.</p>

4 For more information, see §14.4, “Positioning Content,” on page 185.

5 **Intent**

6 element **SignatureDefinitionType/Intent**

diagram	
annotation	<p>A string that represents the intent to which the signing party agrees when signing the document.</p>

7 For more information, see §17.2.2.4, “<Intent> Element.”

1 **LinearGradientBrush**

2 element **LinearGradientBrush**

diagram

attributes

Name	Type	Use	Default	Fixed	Annotation
Opacity	ST_ZeroOne		1.0		Defines the uniform transparency of the linear gradient. Values range from 0 (fully transparent) to 1 (fully opaque), inclusive. Values outside of this range are invalid.
x:Key					Specifies a name for a resource in a resource dictionary. x:Key MUST be present when the current element is defined in a resource dictionary. x:Key MUST NOT be specified outside of a resource dictionary [M6.5].
ColorInterpolationMode	ST_ClrIntMode		SRgbLinear Interpolation		Specifies the gamma function for color interpolation. The gamma adjustment should not be applied to

					the alpha component, if specified. Valid values are SRgbLinearInterpolation and ScRgbLinearInterpolation.
SpreadMethod	<a href="#">ST_SpreadMethod</a>		Pad		Describes how the brush should fill the content area outside of the primary, initial gradient area. Valid values are Pad, Reflect and Repeat.
MappingMode	<a href="#">ST_MappingMode</a>	required		Absolute	Specifies that the start point and end point are defined in the effective coordinate space (includes the Transform attribute of the brush).
Transform	<a href="#">ST_RscRefMatrix</a>				Describes the matrix transformation applied to the coordinate space of the brush. The Transform property on a brush is concatenated with the current effective render transform to yield an effective render transform local to the brush. The start point and end point are transformed using the local effective render transform.
StartPoint	<a href="#">ST_Point</a>	required			Specifies the starting point of the linear gradient.
EndPoint	<a href="#">ST_Point</a>	required			Specifies the end point of the linear gradient. The linear gradient brush interpolates the colors from the start point to the end point, where the start point represents an offset of 0, and the EndPoint

						represents an offset of 1. The Offset attribute value specified in a GradientStop element relates to the 0 and 1 offsets defined by the start point and end point.
annotation	Fills a region with a linear gradient.					

- 1 For more information, see
  - 2 • Section 13.5, “<LinearGradientBrush> Element,” on page 151
  - 3 • 15, “Color”

4 **LinearGradientBrush.GradientStops**

5 element **LinearGradientBrush.GradientStops**

diagram	
annotation	Holds a sequence of GradientStop elements.

- 6 For more information, see §13.5.2, “<LinearGradientBrush.GradientStops> Element,” on page 157.

8 **LinearGradientBrush.Transform**

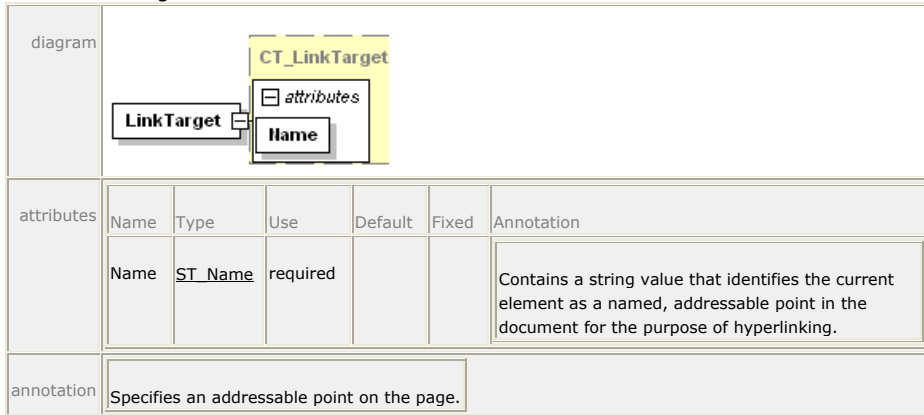
9 element **LinearGradientBrush.Transform**

diagram	
annotation	Describes the matrix transformation applied to the coordinate space of the brush. The Transform property is concatenated with the current effective render transform to yield an effective render transform local to the brush. The start point and end point are transformed using the local effective render transform.

- 10 For more information, see §14.4.8, “<LinearGradientBrush.Transform> Element,” on page 197.

1 **LinkTarget**

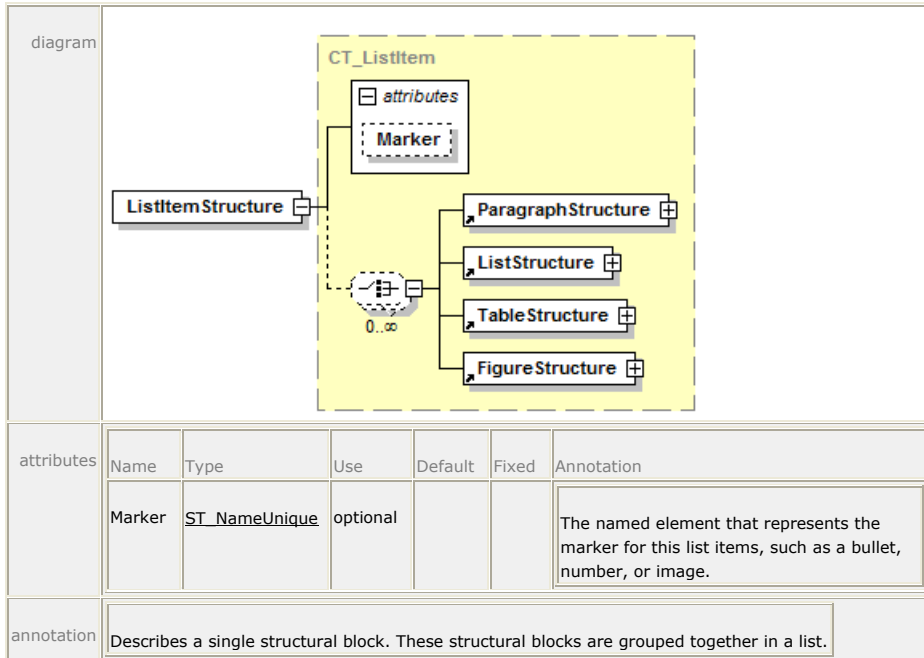
2 element **LinkTarget**



3 For more information, see §10.2.3, “<LinkTarget> Element,” on page 52.

4 **ListItemStructure**

5 element **ListItemStructure**

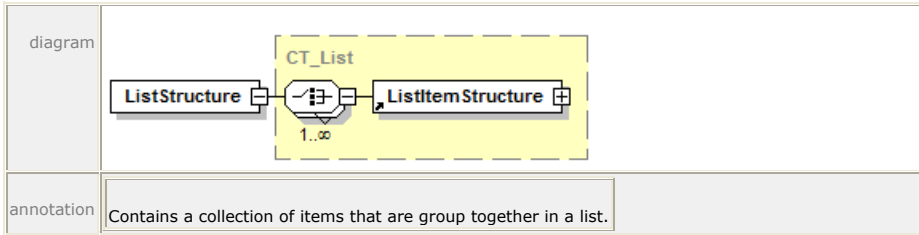


6 For more information, see §16.1.2.11, “<ListItemStructure> Element,” on page 242.



1 **ListStructure**

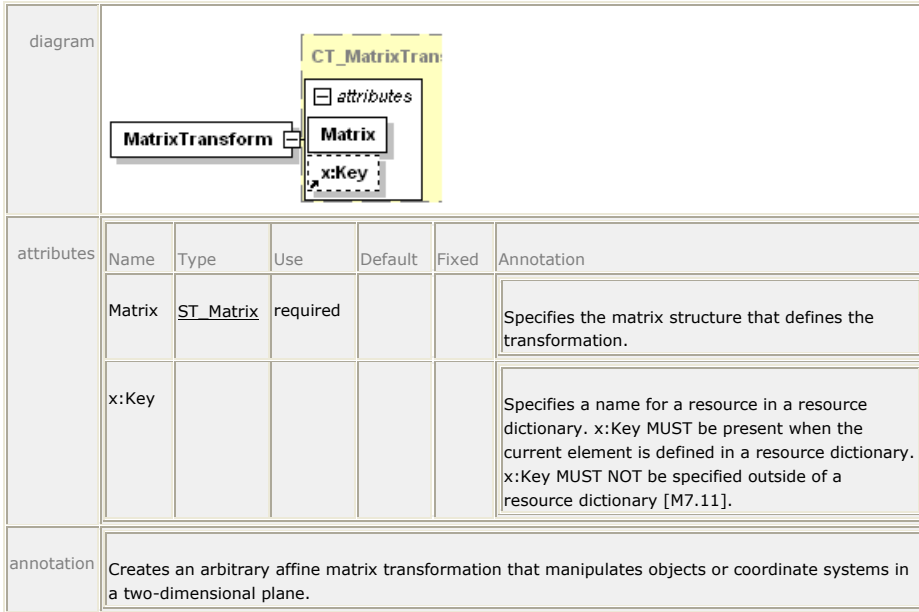
2 element **ListStructure**



3 For more information, see §16.1.2.10, "<ListStructure> Element," on page 241.

4 **MatrixTransform**

5 element **MatrixTransform**



6 For more information, see §14.4.1, "<MatrixTransform> Element," on page 185.

1 **NamedElement**

2 element **NamedElement**

diagram	<p>The diagram shows a box labeled 'NamedElement' with a small square icon on its right side. A line connects this icon to a larger box labeled 'NameReference'. This 'NameReference' box is contained within a larger dashed yellow box labeled 'CT_NamedElement'. Inside the 'CT_NamedElement' box, there is also a smaller box labeled 'attributes' with a minus sign icon on its top-left corner.</p>					
attributes	Name	Type	Use	Default	Fixed	Annotation
	NameReference	ST_Name	required			Identifies the named element in the FixedPage part markup that is referenced by the document structure markup.
annotation	All document structure is related to the fixed page markup using this element. The <NamedElement> points to a single markup element contained in the fixed page markup.					

3 For more information, see §16.1.2.13, “<NamedElement> Element,” on page 243.

4 **OutlineEntry**

5 element **OutlineEntry**

diagram	<p>The diagram shows a box labeled 'OutlineEntry' with a small square icon on its right side. A line connects this icon to a larger dashed yellow box labeled 'CT_OutlineEntry'. Inside the 'CT_OutlineEntry' box, there are three smaller boxes: 'OutlineLevel' (with a dashed border), 'OutlineTarget' (with a solid border), and 'Description' (with a solid border). There is also a box labeled 'xml:lang' (with a dashed border) at the bottom of the 'CT_OutlineEntry' box.</p>					
attributes	Name	Type	Use	Default	Fixed	Annotation
	OutlineLevel	ST_IntGEOne	optional	1		A description of the level where the outline entry exists in the hierarchy. A value of 1 is the root.
	OutlineTarget	xs:anyURI	required			The URI to which the outline entry is linked. This may be a URI to a named element within the document or an

					external URI, such as a website. It can be used as a hyperlink destination.
Description	xs:string	required			The friendly text associated with this outline entry.
xml:lang		optional			This attribute specifies the default language used for any child element contained within the current element or nested child elements. The language is specified according to IETF RFC 3066.
annotation	Represents an index to a specific location in the document.				

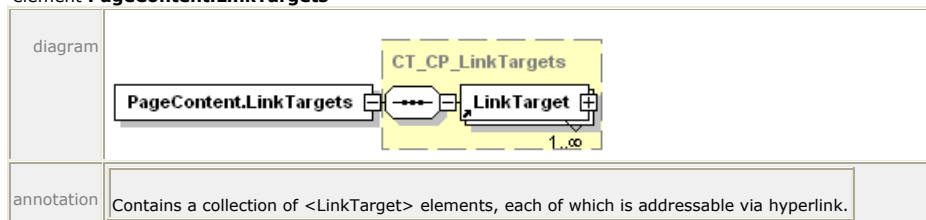
1 For more information, see §16.1.1.4, “<OutlineEntry> Element,” on page 227.

2 **PageContent**

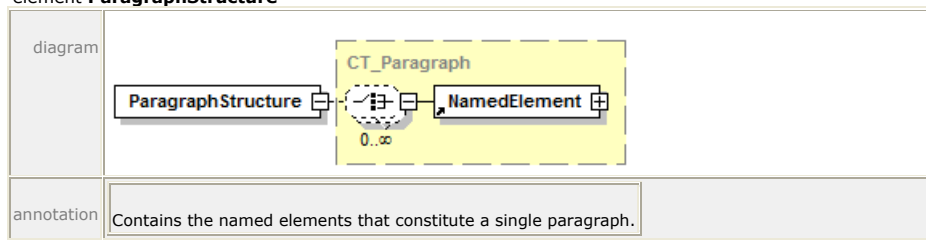
3 element **PageContent**

diagram						
attributes	Name	Type	Use	Default	Fixed	Annotation
	Source	xs:anyURI	required			Specifies a URI that refers to the page content, held in a distinct part within the package. The content identified MUST be a FixedPage part within the XPS Document [M3.5].
	Width	ST_GEOne				Typical width of pages contained in the page content.
	Height	ST_GEOne				Typical height of pages contained in the page content.
annotation	Defines a reference from a fixed document to a part that contains a <FixedPage> element.					

4 For more information, see §10.2.1, “<PageContent> Element,” on page 50.

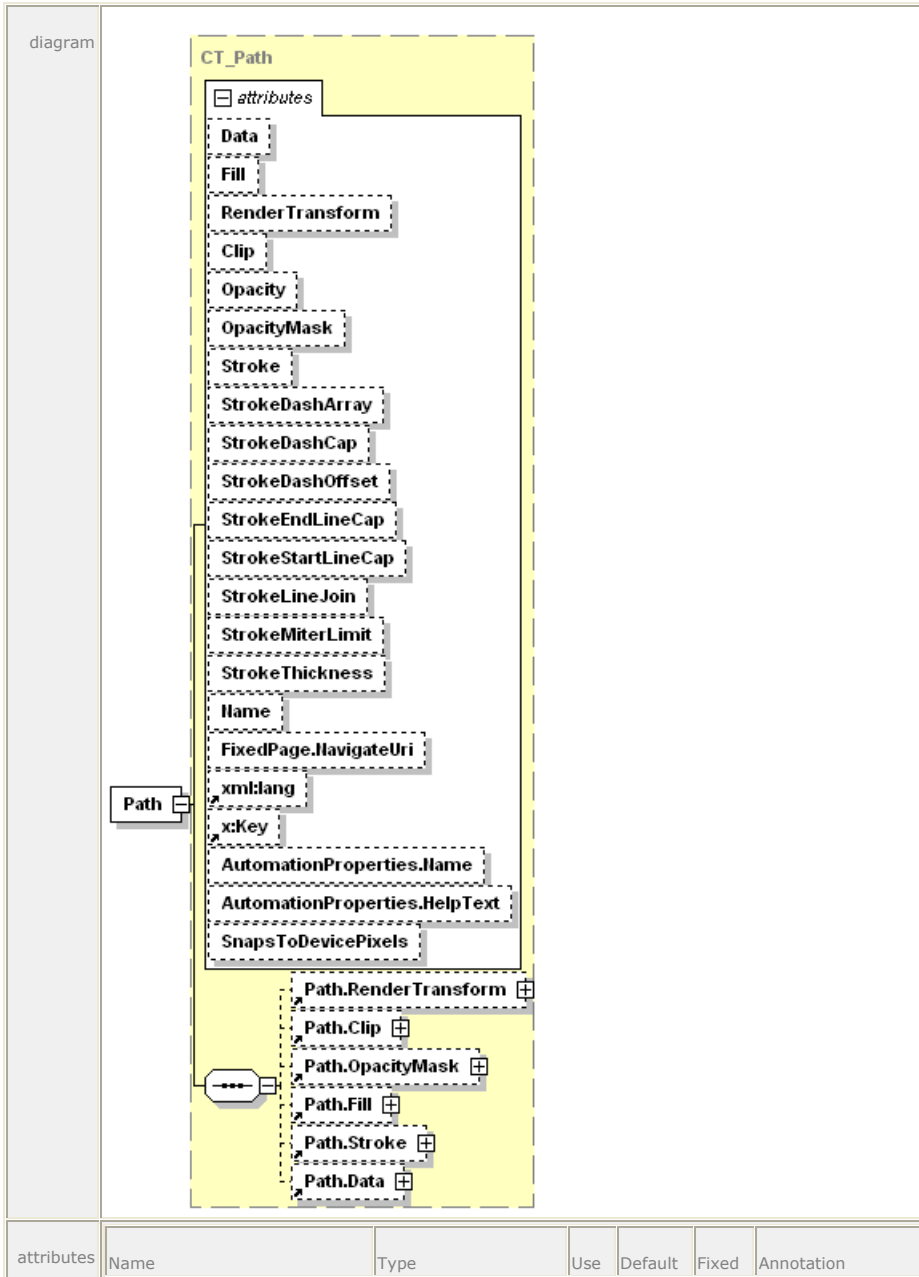
1 **PageContent.LinkTargets**2 element **PageContent.LinkTargets**

3 For more information, see §10.2.2, "<PageContent.LinkTargets> Element," on  
 4 page 51.

5 **ParagraphStructure**6 element **ParagraphStructure**

7 For more information, see §16.1.2.5, "<ParagraphStructure> Element," on page  
 8 239.

- 1 **Path**
- 2 element **Path**



Data	<a href="#">ST_RscRefAbbrGeomF</a>				Describes the geometry of the path.
Fill	<a href="#">ST_RscRefColor</a>				Describes the brush used to paint the geometry specified by the Data property of the path.
RenderTransform	<a href="#">ST_RscRefMatrix</a>				Establishes a new coordinate frame for all attributes of the path and for all child elements of the path, such as the geometry defined by the <Path.Data> property element.
Clip	<a href="#">ST_RscRefAbbrGeomF</a>				Limits the rendered region of the element.
Opacity	<a href="#">ST_ZeroOne</a>		1.0		Defines the uniform transparency of the path element. Values range from 0 (fully transparent) to 1 (fully opaque), inclusive. Values outside of this range are invalid.
OpacityMask	<a href="#">ST_RscRef</a>				Specifies a mask of alpha values that is applied to the path in the same fashion as the Opacity attribute, but allowing different alpha values for different areas of the element.

Stroke	<a href="#">ST_RscRefColor</a>				Specifies the brush used to draw the stroke.
StrokeDashArray	<a href="#">ST_EvenArrayPos</a>				Specifies the length of dashes and gaps of the outline stroke. These values are specified as multiples of the stroke thickness as a space-separated list with an even number of non-negative values. When a stroke is drawn, the dashes and gaps specified by these values are repeated to cover the length of the stroke. If this attribute is omitted, the stroke is drawn solid, without any gaps.
StrokeDashCap	<a href="#">ST_DashCap</a>		Flat		Specifies how the ends of each dash are drawn. Valid values are Flat, Round, Square, and Triangle.
StrokeDashOffset	<a href="#">ST_Double</a>		0.0		Adjusts the start point for repeating the dash array pattern. If this value is omitted, the dash array aligns with the origin of the stroke. Values are specified as multiples of the stroke thickness.
StrokeEndLineCap	<a href="#">ST_LineCap</a>		Flat		Defines the shape of the end of the

					last dash in a stroke. Valid values are Flat, Square, Round, and Triangle.
StrokeStartLineCap	<a href="#">ST_LineCap</a>		Flat		Defines the shape of the beginning of the first dash in a stroke. Valid values are Flat, Square, Round, and Triangle.
StrokeLineJoin	<a href="#">ST_LineJoin</a>		Miter		Specifies how a stroke is drawn at a corner of a path. Valid values are Miter, Bevel, and Round. If Miter is selected, the value of StrokeMiterLimit is used in drawing the stroke.
StrokeMiterLimit	<a href="#">ST_GEOne</a>		10.0		The ratio between the maximum miter length and half of the stroke thickness. This value is significant only if the StrokeLineJoin attribute specifies Miter.
StrokeThickness	<a href="#">ST_GEZero</a>		1.0		Specifies the thickness of a stroke, in units of the effective coordinate space (includes the path's render transform). The stroke is drawn on top of the boundary of the geometry specified by the <Path> element's Data property. Half of the



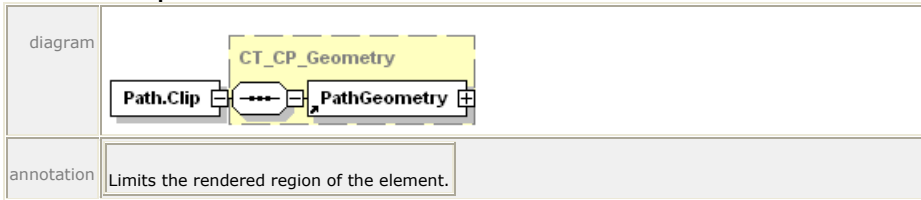
					StrokeThickness extends outside of the geometry specified by the Data property and the other half extends inside of the geometry.
Name	<u>ST_Name</u>				Contains a string value that identifies the current element as a named, addressable point in the document for the purpose of hyperlinking.
FixedPage.NavigateUri	xs:anyURI				Associates a hyperlink URI with the element. Can be a relative reference or a URI that addresses a resource that is internal to or external to the package.
xml:lang					Specifies the default language used for the current element and for any child or descendant elements. The language is specified according to RFC 3066.
x:Key					Specifies a name for a resource in a resource dictionary. x:Key MUST be present when the current element is defined in a resource dictionary. x:Key MUST NOT be specified outside of a resource

					dictionary [M4.1].
	AutomationProperties.Name	xs:string			A brief description of the <Path> for accessibility purposes, particularly if filled with an <ImageBrush>.
	AutomationProperties.HelpText	xs:string			A detailed description of the <Path> for accessibility purposes, particularly if filled with an <ImageBrush>.
	SnapsToDevicePixels	ST_Boolean			On Anti-aliasing consumers controls if control points snap to the nearest device pixels. Valid values are 'false' and 'true'. Consumers MAY ignore this attribute [O4.1].
annotation	Defines a single graphical effect to be rendered to the page. It paints a geometry with a brush and draws a stroke around it.				

- 1 For more information, see
- 2
  - Section 11.1, "<Path> Element," on page 68
- 3
  - Section **11.2.3**, "**Abbreviated Geometry Syntax**," on page 90

1 **Path.Clip**

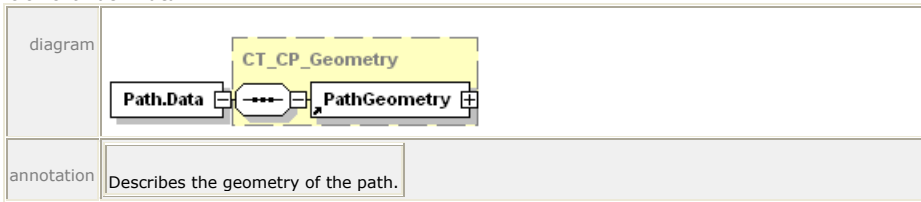
2 element **Path.Clip**



3 For more information, see §14.3, “Clipping,” on page 182.

4 **Path.Data**

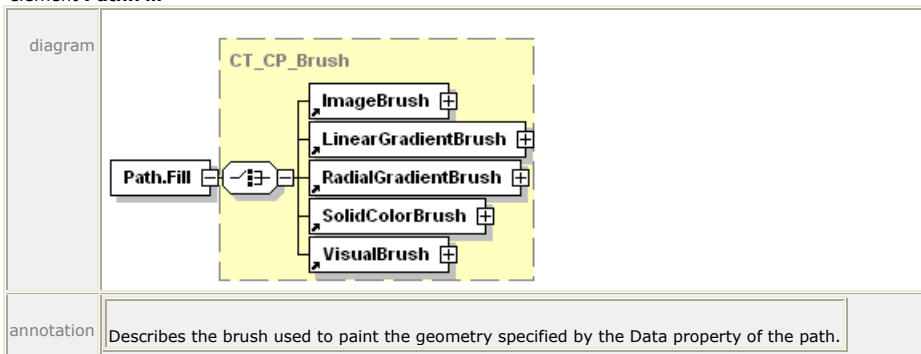
5 element **Path.Data**



6 For more information, see §11.1.1, “<Path.Data> Element,” on page 74.

7 **Path.Fill**

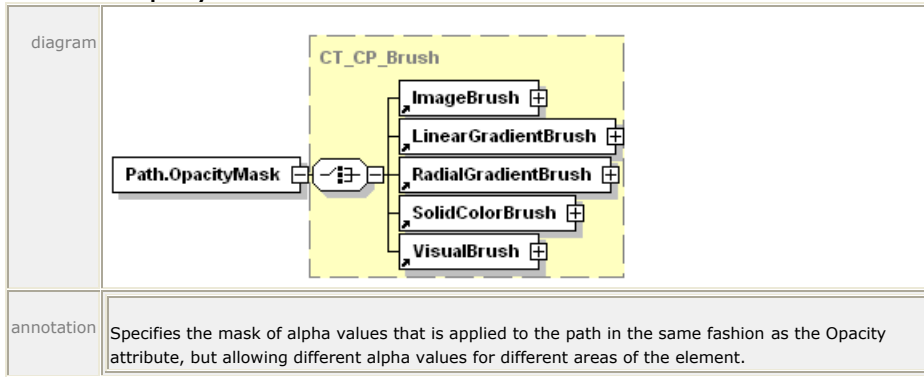
8 element **Path.Fill**



9 For more information, see §11.1.2, “<Path.Fill> Element,” on page 75.

1 **Path.OpacityMask**

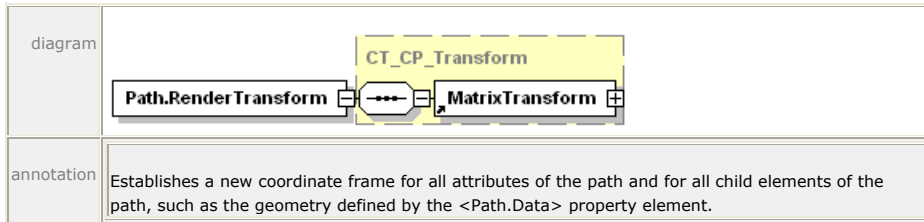
2 element **Path.OpacityMask**



3 For more information, see §14.5.2, “<Path.OpacityMask,” on page 201.

4 **Path.RenderTransform**

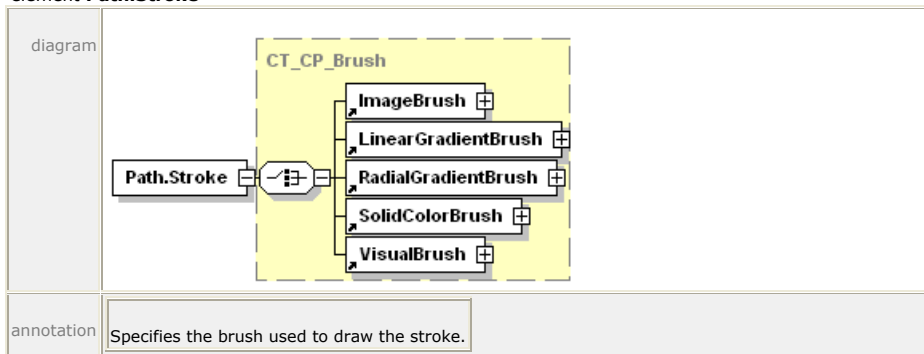
5 element **Path.RenderTransform**



6 For more information, see §14.4, “Positioning Content,” on page 185.

7 **Path.Stroke**

8 element **Path.Stroke**



9

1 For more information, see §11.1.3, "<Path.Stroke> Element," on page 76.

2 **PathFigure**

3 element **PathFigure**

diagram

attributes

Name	Type	Use	Default	Fixed	Annotation
IsClosed	<a href="#">ST_Boolean</a>		false		Specifies whether the path is closed. If set to true, the stroke is drawn "closed," that is, the last point in the last segment of the path figure is connected with the point specified in the StartPoint attribute, otherwise the stroke is drawn "open," and the last point is not connected to the start point. Only applicable if the path figure is used in a <Path> element that specifies a stroke.
StartPoint	<a href="#">ST_Point</a>	required			Specifies the starting point for the first segment of the path figure.
IsFilled	<a href="#">ST_Boolean</a>		true		Specifies whether the path figure is used in computing the area of the containing path geometry. Can be true or false. When set to false, the path figure is considered only for stroking.

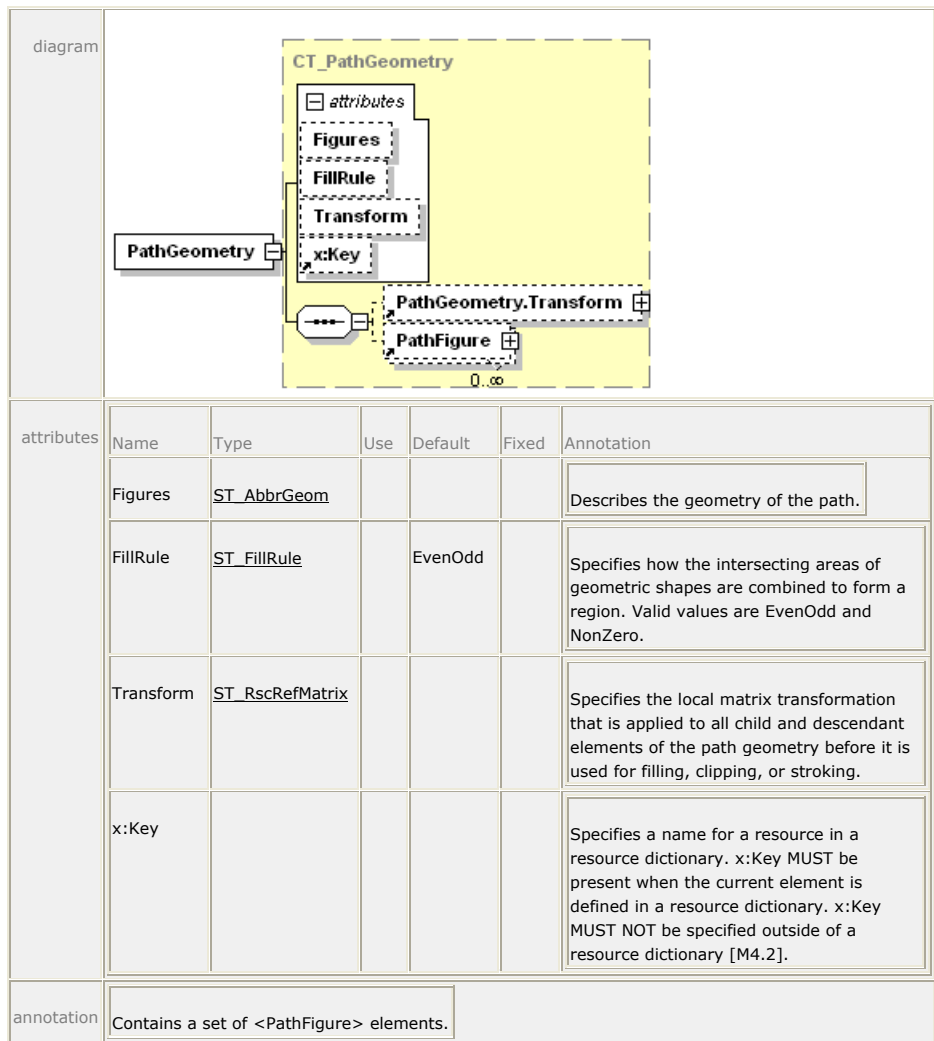
annotation

Specifies a set of one or more segment elements defining a closed region.

4 For more information, see §11.2.2.1, "<PathFigure> Element."

5 **PathGeometry**

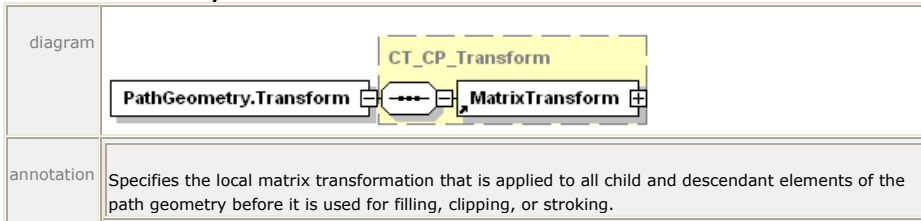
6 element **PathGeometry**



1 For more information, see §11.2.1.1, "<PathGeometry> Element."

1 **PathGeometry.Transform**

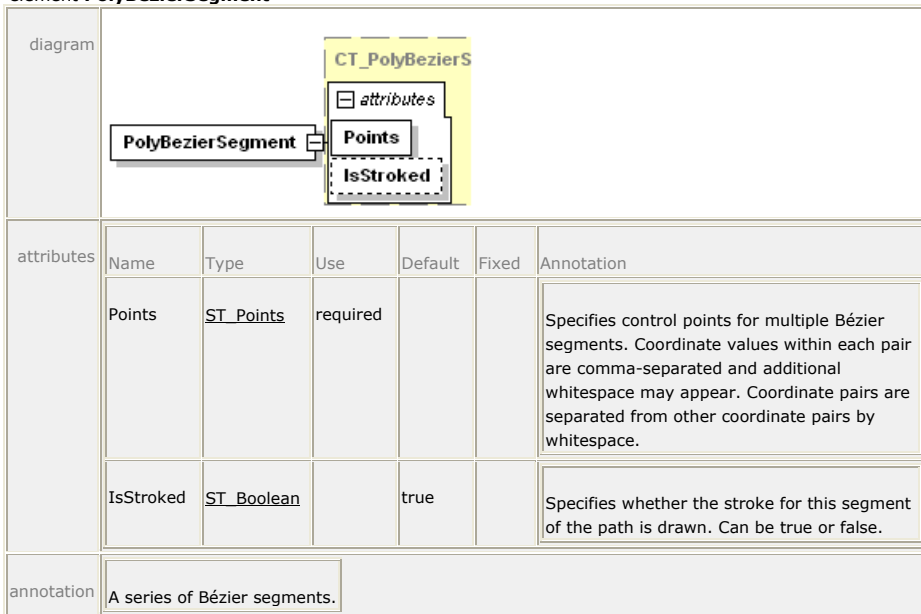
2 element **PathGeometry.Transform**



3 For more information, see §14.4.5, “<PathGeometry.Transform> Element,” on  
 4 page 192.

5 **PolyBezierSegment**

6 element **PolyBezierSegment**



7

1 **PolyLineSegment**

2 element **PolyLineSegment**

diagram						
attributes	Name	Type	Use	Default	Fixed	Annotation
	Points	<a href="#">ST_Points</a>	required			Specifies a set of coordinates for the multiple segments that define the poly line segment. Coordinate values within each pair are comma-separated and additional whitespace may appear. Coordinate pairs are separated from other coordinate pairs by whitespace.
	IsStroked	<a href="#">ST_Boolean</a>		true		Specifies whether the stroke for this segment of the path is drawn. Can be true or false.
annotation	Specifies a set of points between which lines are drawn.					

3

4 **PolyQuadraticBezierSegment**

5 element **PolyQuadraticBezierSegment**

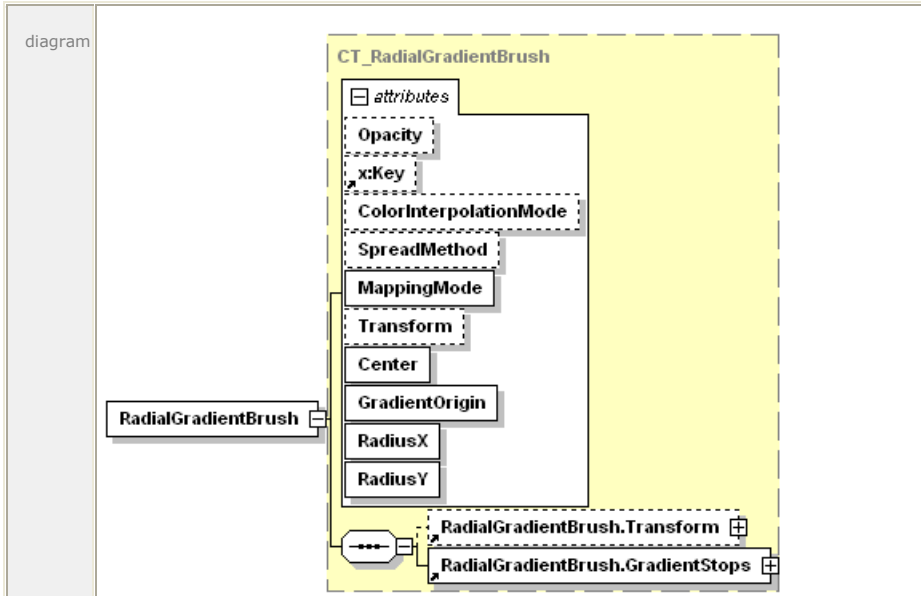
diagram						
attributes	Name	Type	Use	Default	Fixed	Annotation
	Points	<a href="#">ST_Points</a>	required			Specifies control points for multiple quadratic Bézier segments. Coordinate values within each pair are comma-separated and additional whitespace may appear. Coordinate pairs are separated from other coordinate pairs by whitespace.
	IsStroked	<a href="#">ST_Boolean</a>		true		Specifies whether the stroke for this segment of the path is drawn. Can be true or false.



1 annotation A series of quadratic Bézier segments.

2 **RadialGradientBrush**

3 element **RadialGradientBrush**



attributes	Name	Type	Use	Default	Fixed	Annotation
	Opacity	ST_ZeroOne		1.0		Defines the uniform transparency of the radial gradient. Values range from 0 (fully transparent) to 1 (fully opaque), inclusive. Values outside of this range are invalid.
	x:Key					Specifies a name for a resource in a resource dictionary. x:Key MUST be present when the current element is defined in a resource dictionary. x:Key MUST NOT be specified outside of a resource dictionary [M6.6].

ColorInterpolationMode	<a href="#">ST_ClrIntMode</a>		SRgbLinear Interpolation		Specifies the gamma function for color interpolation for sRGB colors. The gamma adjustment should not be applied to the alpha component, if specified. Valid values are SRgbLinearInterpolation and ScRgbLinearInterpolation.
SpreadMethod	<a href="#">ST_SpreadMethod</a>		Pad		Describes how the brush should fill the content area outside of the primary, initial gradient area. Valid values are Pad, Reflect and Repeat.
MappingMode	<a href="#">ST_MappingMode</a>	required		Absolute	Specifies that center, x radius, and y radius are defined in the effective coordinate space (includes the Transform attribute of the brush).
Transform	<a href="#">ST_RscRefMatrix</a>				Describes the matrix transformation applied to the coordinate space of the brush. The Transform property is concatenated with the current effective render transform to yield an effective render transform local to the brush. The ellipse defined by the center, gradient origin, x radius, and y radius values is transformed using the local effective render transform.
Center	<a href="#">ST_Point</a>	required			Specifies the center point of the radial gradient (that is, the center of the ellipse). The radial gradient brush interpolates the colors from the gradient

					origin to the circumference of the ellipse. The circumference is determined by the center and the radii.
	GradientOrigin	<a href="#">ST_Point</a>	required		Specifies the origin point of the radial gradient.
	RadiusX	<a href="#">ST_GZero</a>	required		Specifies the radius in the x dimension of the ellipse which defines the radial gradient.
	RadiusY	<a href="#">ST_GZero</a>	required		Specifies the radius in the y dimension of the ellipse which defines the radial gradient.
annotation	Fills a region with a radial gradient.				

- 1 For more information, see:
- 2     • Section 13.6, “<RadialGradientBrush> Element,” on page 158
- 3     • 15, “Color”

4 **RadialGradientBrush.GradientStops**

5 element **RadialGradientBrush.GradientStops**

diagram	
annotation	Holds a sequence of <GradientStop> elements.

- 6 For more information, see §13.6.2, “<RadialGradientBrush.GradientStops>
- 7 Element,” on page 165.

1 **RadialGradientBrush.Transform**

2 element **RadialGradientBrush.Transform**

diagram	
annotation	<p>Describes the matrix transformation applied to the coordinate space of the brush. The Transform property is concatenated with the current effective render transform to yield an effective render transform local to the brush. The center, gradient origin, x radius, and y radius are transformed using the local effective render transform.</p>

3 For more information, see §14.4.9, “<RadialGradientBrush.Transform> Element,” on  
 4 page 198.

5 **ResourceDictionary**

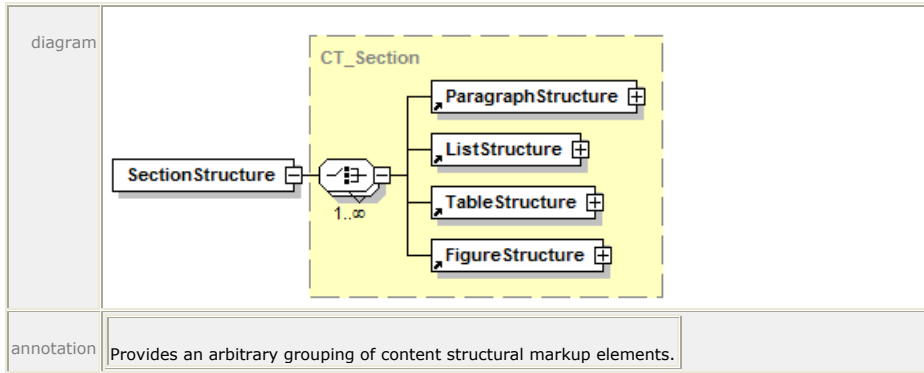
6 element **ResourceDictionary**

diagram													
attributes	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> <th>Annotation</th> </tr> </thead> <tbody> <tr> <td>Source</td> <td>xs:anyURI</td> <td></td> <td></td> <td></td> <td>Specifies the URI of a part containing markup for a resource dictionary. The URI MUST refer to a part in the package [M2.1].</td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	Annotation	Source	xs:anyURI				Specifies the URI of a part containing markup for a resource dictionary. The URI MUST refer to a part in the package [M2.1].
Name	Type	Use	Default	Fixed	Annotation								
Source	xs:anyURI				Specifies the URI of a part containing markup for a resource dictionary. The URI MUST refer to a part in the package [M2.1].								
annotation	<p>Defines a set of reusable resource definitions that can be used as property values in the fixed page markup.</p>												

- 1 For more information, see §14.2, "Resources and Resource References," on
- 2 page 172.

1 **SectionStructure**

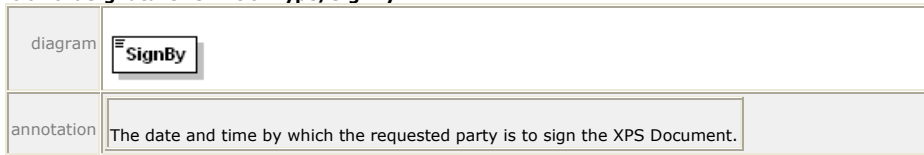
2 element **SectionStructure**



3 For more information, see §16.1.2.4, “<SectionStructure> Element,” on page 239.

4 **SignBy**

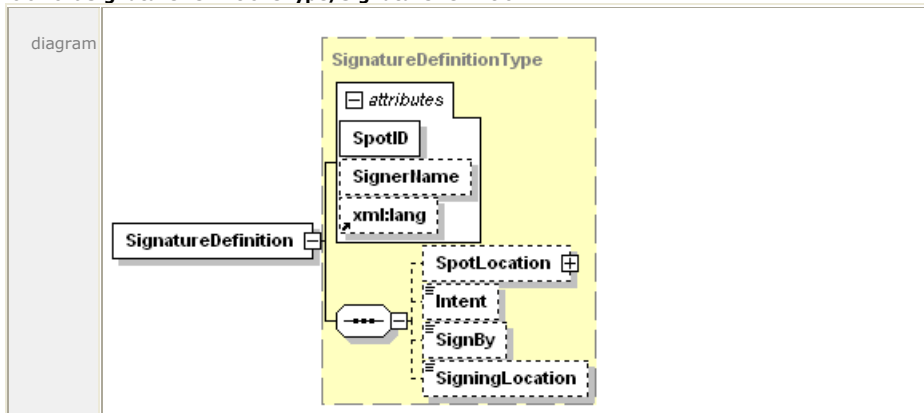
5 element **SignatureDefinitionType/SignBy**



6 For more information, see §17.2.2.5, “<SignBy> Element.”

7 **SignatureDefinition**

8 element **SignatureDefinitionsType/SignatureDefinition**

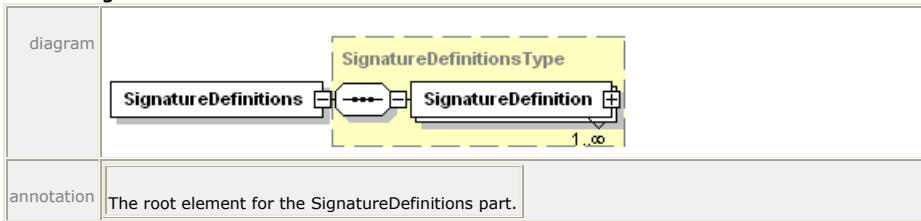


attributes	Name	Type	Use	Default	Fixed	Annotation
	SpotID	xs:ID	required			A globally unique identifier for this signature spot.
	SignerName	xs:string				A string representing the identity of the individual who is requested to sign the XPS Document, or the name of the individual who has signed the XPS Document.
	xml:lang					Specifies the language used for the current element and its descendants. The language is specified according to RFC 3066.
annotation	A single signature definition.					

1 For more information, see §17.2.2.2, “<SignatureDefinition> Element.”

2 **SignatureDefinitions**

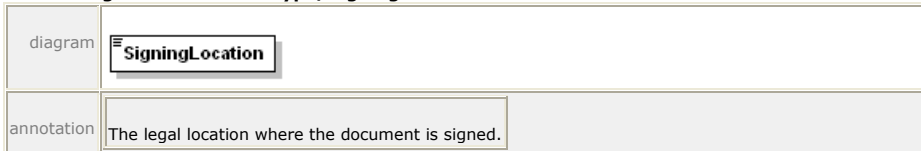
3 element **SignatureDefinitions**



4 For more information, see §17.2.2.1, “<SignatureDefinitions> Element.”

5 **SigningLocation**

6 element **SignatureDefinitionType/SigningLocation**



7 For more information, see §17.2.2.6, “<SigningLocation> Element.”

1 **SolidColorBrush**

2 element **SolidColorBrush**

diagram						
attributes	Name	Type	Use	Default	Fixed	Annotation
	Opacity	ST_ZeroOne		1.0		Defines the uniform transparency of the brush fill. Values range from 0 (fully transparent) to 1 (fully opaque), inclusive. Values outside of this range are invalid.
	x:Key					Specifies a name for a resource in a resource dictionary. x:Key MUST be present when the current element is defined in a resource dictionary. x:Key MUST NOT be specified outside of a resource dictionary [M6.1].
	Color	ST_Color	required			Specifies the color for filled elements. An sRGB color value specified as a 6-digit hexadecimal number (#RRGGBB) or an extended color.
annotation	Fills defined geometric regions with a solid color.					

3 For more information, see §13.1, "<SolidColorBrush," on page 126.

4 **SpotLocation**

5 element **SignatureDefinitionType/SpotLocation**

diagram						
---------	--	--	--	--	--	--



attributes	Name	Type	Use	Default	Fixed	Annotation
	PageURI	xs:anyURI	required			Specifies the page on which the signature spot should be displayed.
	StartX	xs:double	required			Specifies the x coordinate of the origin point (upper-left corner) on the page where the signature spot should be displayed.
	StartY	xs:double	required			Specifies the y coordinate of the origin point (upper-left corner) on the page where the signature spot should be displayed.
annotation	Specifies where a consumer should place a signature spot.					

1 For more information, see §17.2.2.3, “<SpotLocation> Element.”

2 **Story**

3 element **Story**

attributes	Name	Type	Use	Default	Fixed	Annotation
	StoryName	xs:string	required			The name used by story fragments to identify they belong to this story.
	annotation					
	Defines a single story and where each of its story fragments appear in the XPS Document.					

diagram

```

classDiagram
    class Story {
        +attributes
        +StoryName
        +StoryFragmentReference 1..∞
    }
    class attributes {
        +StoryName
        +StoryFragmentReference
    }
    Story "1" -- "1..∞" attributes
    
```

4 For more information, see §16.1.1.5, “<Story> Element,” on page 228.

1 **StoryBreak**

2 element **StoryBreak**

diagram	
annotation	If located at the beginning of a <StoryFragment> definition, indicates that the following markup elements should not be merged with the markup from the previous <StoryFragment>. If located at the end of a <StoryFragment> definition, indicates that the preceding markup elements should not be merged with the subsequent <StoryFragment>.

3 For more information, see §16.1.2.3, “<StoryBreak> Element,” on page 239.

4 **StoryFragment**

5 element **StoryFragment**

diagram						
attributes	Name	Type	Use	Default	Fixed	Annotation
	StoryName	xs:string	optional			Identifies the story that this story fragment belongs to. If omitted, the story fragment is not associated with any story.

	FragmentName	xs:string	optional			Used to uniquely identify the story fragment.
	FragmentType	<u>ST_FragmentType</u>	required			Specifies the type of content included in the story fragment. Valid values are Content, Header, and Footer.
annotation	Specifies the document structural markup that appears on the current page for a single story block.					

1 For more information, see §16.1.2.2, “<StoryFragment> Element,” on page 235.

2 **StoryFragments**

3 element **StoryFragments**

diagram	
annotation	The root of a StoryFragments part. Contains all story fragments that appear on a specific page.

4 For more information, see §16.1.2.1, “<StoryFragments> Element,” on page 234.

5 **StoryFragmentReference**

6 element **StoryFragmentReference**

diagram																			
attributes	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> <th>Annotation</th> </tr> </thead> <tbody> <tr> <td>FragmentName</td> <td>xs:string</td> <td>optional</td> <td></td> <td></td> <td>Used to distinguish between multiple story fragments from the same story on a single page.</td> </tr> <tr> <td>Page</td> <td><u>ST_IntGEOne</u></td> <td>required</td> <td></td> <td></td> <td>Identifies the page number of the document that the story fragment is related to. Page numbers start at 1</td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	Annotation	FragmentName	xs:string	optional			Used to distinguish between multiple story fragments from the same story on a single page.	Page	<u>ST_IntGEOne</u>	required			Identifies the page number of the document that the story fragment is related to. Page numbers start at 1
Name	Type	Use	Default	Fixed	Annotation														
FragmentName	xs:string	optional			Used to distinguish between multiple story fragments from the same story on a single page.														
Page	<u>ST_IntGEOne</u>	required			Identifies the page number of the document that the story fragment is related to. Page numbers start at 1														

						and correspond to the order of <PageContent> elements in the FixedDocument part.
annotation	Identifies the StoryFragments part where this individual story fragment is defined.					

1 For more information, see §16.1.1.6, "<StoryFragmentReference> Element," on  
 2 page 229.

3 **TableCellStructure**

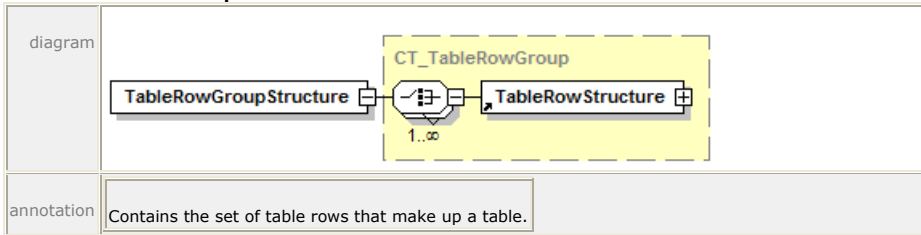
4 element **TableCellStructure**

diagram						
attributes	Name	Type	Use	Default	Fixed	Annotation
	RowSpan	ST_TableSpan	optional	1		Indicates the number of rows this cell spans, or merges into a single cell.
	ColumnSpan	ST_TableSpan	optional	1		Indicates the number of columns this cell spans, or merges into a single cell.
annotation	Contains the elements that occupy a single cell of a table.					

5 For more information, see §16.1.2.9, "<TableCellStructure> Element," on page 241.

1 **TableRowGroupStructure**

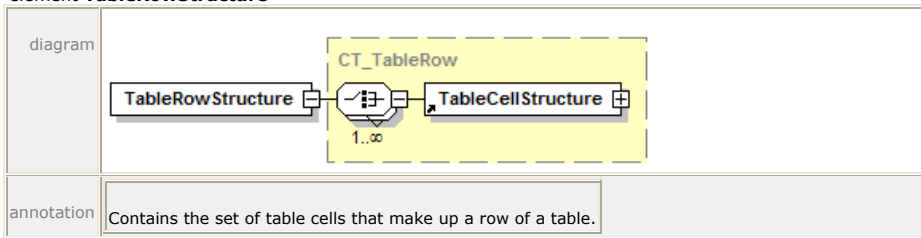
2 element **TableRowGroupStructure**



3 For more information, see §16.1.2.7, “<TableRowGroupStructure> Element,” on  
 4 page 240.

5 **TableRowStructure**

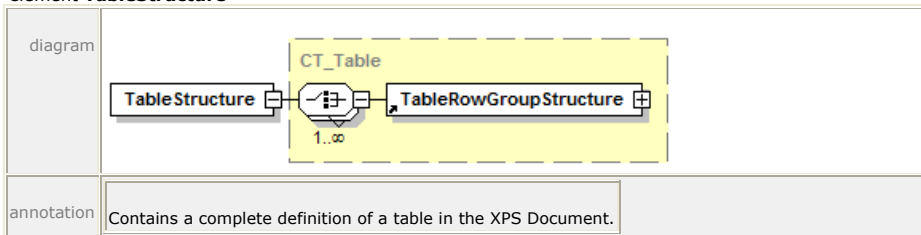
6 element **TableRowStructure**



7 For more information, see §16.1.2.8, “<TableRowStructure> Element,” on page 240.

8 **TableStructure**

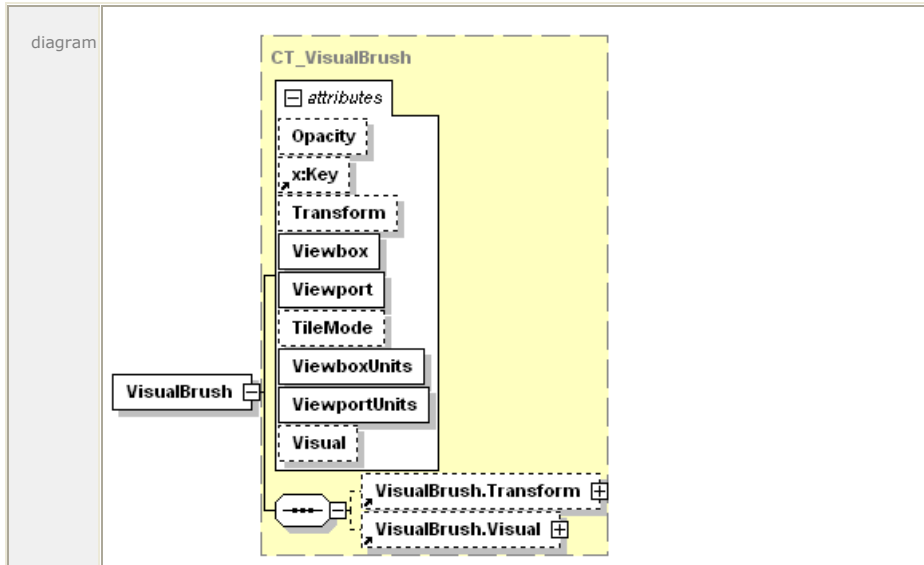
9 element **TableStructure**



10 For more information, see §16.1.2.6, “<TableStructure> Element,” on page 240.

1 **VisualBrush**

2 element **VisualBrush**



attributes	Name	Type	Use	Default	Fixed	Annotation
	Opacity	<a href="#">ST_ZeroOne</a>		1.0		Defines the uniform transparency of the brush fill. Values range from 0 (fully transparent) to 1 (fully opaque), inclusive. Values outside of this range are invalid.
	x:Key					Specifies a name for a resource in a resource dictionary. x:Key MUST be present when the current element is defined in a resource dictionary. x:Key MUST NOT be specified outside of a resource dictionary [M6.4].
	Transform	<a href="#">ST_RscRefMatrix</a>				Describes the matrix transformation applied to the coordinate space of the brush. The Transform property is concatenated with the current effective render transform to yield an effective render transform local to the brush. The viewport for the brush is transformed using that local

					effective render transform.
Viewbox	<a href="#">ST_ViewBox</a>	required			Specifies the position and dimensions of the brush's source content. Specifies four comma-separated real numbers (x, y, Width, Height), where width and height are non-negative. The viewbox defines the default coordinate system for the element specified in the <VisualBrush.Visual> property element. The corners of the viewbox are mapped to the corners of the viewport, thereby providing the default clipping and transform for the brush's source content.
Viewport	<a href="#">ST_ViewBox</a>	required			Specifies the region in the containing coordinate space of the prime brush tile that is (possibly repeatedly) applied to fill the region to which the brush is applied. Specifies four comma-separated real numbers (x, y, Width, Height), where width and height are non-negative. The alignment of the brush pattern is controlled by adjusting the x and y values.
TileMode	<a href="#">ST_TileMode</a>		None		Specifies how contents will be tiled in the filled region. Valid values are None, Tile, FlipX, FlipY, and FlipXY.
ViewboxUnits	<a href="#">ST_ViewUnits</a>	required		Absolute	Specifies the relationship of the viewbox coordinates to the containing coordinate space.
ViewportUnits	<a href="#">ST_ViewUnits</a>	required		Absolute	Specifies the relationship of the viewport coordinates to the containing coordinate space.
Visual	<a href="#">ST_RscRef</a>				Specifies resource reference to a <Path>, <Glyphs>, or <Canvas> element defined in a resource dictionary and used to draw the brush's source content.

annotation	Fills a region with a drawing. The drawing may be specified as either a child of the <VisualBrush> element, or as a resource reference. Drawing content is expressed using <Canvas>, <Path>, and <Glyphs> elements.
------------	---

1 For more information, see §13.3, “<VisualBrush> Element,” on page 131.

2 **VisualBrush.Transform**

3 element **VisualBrush.Transform**

diagram	<p>The diagram shows a box labeled 'VisualBrush.Transform' connected by an arrow to a box labeled 'MatrixTransform'. The 'MatrixTransform' box is enclosed within a dashed yellow box labeled 'CT_CP_Transform'.</p>
annotation	Describes the matrix transformation applied to the coordinate space of the brush. The Transform property is concatenated with the current effective render transform to yield an effective render transform local to the brush. The viewport for the brush is transformed using the local effective render transform.

4 For more information, see §14.4, “Positioning Content,” on page 185.

5 **VisualBrush.Visual**

6 element **VisualBrush.Visual**

diagram	<p>The diagram shows a box labeled 'VisualBrush.Visual' connected by an arrow to a dashed yellow box labeled 'CT_CP_Visual'. Inside the 'CT_CP_Visual' box, there are three stacked boxes: 'Path', 'Glyphs', and 'Canvas', each with an arrow pointing from the 'CT_CP_Visual' container to it.</p>
annotation	Specifies a <Path> element, <Glyphs> element, or <Canvas> element used to draw the brush's source contents.

7  
8  
9



## 1 A. Signature Definitions Schema

```

2 <?xml version="1.0" encoding="utf-8"?>
3 <xs:schema targetNamespace="http://schemas.microsoft.com/xps/2005/06/signature-
4 definitions" xmlns="http://schemas.microsoft.com/xps/2005/06/signature-
5 definitions" xmlns:xs="http://www.w3.org/2001/XMLSchema"
6 elementFormDefault="qualified" blockDefault="#all">
7
8   <xs:import namespace="http://www.w3.org/XML/1998/namespace" />
9
10  <xs:complexType name="SignatureDefinitionsType">
11    <xs:sequence>
12      <xs:element name="SignatureDefinition"
13 type="SignatureDefinitionType" minOccurs="1" maxOccurs="unbounded">
14        </xs:element>
15      </xs:sequence>
16    </xs:complexType>
17
18  <xs:complexType name="SpotLocationType">
19    <xs:attribute name="PageURI" type="xs:anyURI" use="required">
20      </xs:attribute>
21    <xs:attribute name="StartX" type="xs:double" use="required">
22      </xs:attribute>
23    <xs:attribute name="StartY" type="xs:double" use="required">
24      </xs:attribute>
25    </xs:complexType>
26
27  <xs:complexType name="SignatureDefinitionType">
28    <xs:sequence>
29      <xs:element name="SpotLocation" type="SpotLocationType"
30 minOccurs="0">
31        </xs:element>
32      <xs:element name="Intent" type="xs:string" minOccurs="0">
33        </xs:element>
34      <xs:element name="SignBy" type="xs:dateTime" minOccurs="0">
35        </xs:element>
36      <xs:element name="SigningLocation" type="xs:string" minOccurs="0">
37        </xs:element>
38      </xs:sequence>
39      <xs:attribute name="SpotID" type="xs:ID" use="required">
40        </xs:attribute>
41      <xs:attribute name="SignerName" type="xs:string">
42        </xs:attribute>
43      <xs:attribute ref="xml:lang">
44        </xs:attribute>
45      </xs:complexType>
46
47    <xs:element name="SignatureDefinitions" type="SignatureDefinitionsType">
48      </xs:element>
49  </xs:schema>
50

```



## 1 B. XPS Document Schema

```

2 <?xml version="1.0" encoding="utf-8"?>
3 <xs:schema targetNamespace="http://schemas.microsoft.com/xps/2005/06"
4 xmlns="http://schemas.microsoft.com/xps/2005/06"
5 xmlns:mstns="http://tempuri.org/XMLSchema.xsd"
6 xmlns:xs="http://www.w3.org/2001/XMLSchema"
7 xmlns:x="http://schemas.microsoft.com/xps/2005/06/resourcedictionary-key"
8 elementFormDefault="qualified" blockDefault="#all">
9
10 <xs:import
11 namespace="http://schemas.microsoft.com/xps/2005/06/resourcedictionary-key" />
12 <xs:import namespace="http://www.w3.org/XML/1998/namespace" />
13
14 <!-- Names used for types and groups:
15
16     ST_*         simpleType
17     CT_*         complexType
18     G_*          group
19     AG_*         attributeGroup
20
21 -->
22
23 <!-- Individual real number patterns
24 All patterns using numbers now use <whitespace value="collapse">.
25 As a result, any whitespace in the pattern can be expressed as:
26 mandatory whitespace, one or more: " "
27 optional whitespace, zero or more: " ?"
28
29 For better readability, each pattern using numbers is also described in a
30 comment using
31 one of the following pattern designators.
32
33 The actual patterns are generated by replacement by the schema publication
34 process.
35 -->
36 <!--DEFINE [pint]          "([1-9][0-9]*)" -->
37 <!--DEFINE [uint]        "([0-9]+)" -->
38 <!--DEFINE [dec]         "(\\-?(([0-9]+(\\. [0-9]+)?)|\\. [0-9]+))" -->
39 <!--DEFINE [rn]          "((\\-|\\+)?((([0-9]+(\\. [0-9]+)?)|\\. [0-9]+))((e|E)(\\-
40 |\\+)?[0-9]+)?)" -->
41 <!--DEFINE [prn]         "(\\+?(([0-9]+(\\. [0-9]+)?)|\\. [0-9]+))((e|E)(\\-
42 |\\+)?[0-9]+)?" -->
43 <!--DEFINE [scs]        "( ( ?, ? )" -->
44
45
46
47 <!-- Complex Types -->
48 <xs:complexType name="CT_MatrixTransform">
49 <xs:attributeGroup ref="AG_MatrixTransform" />
50 </xs:complexType>
51
52 <xs:complexType name="CT_SolidColorBrush">
53 <xs:attributeGroup ref="AG_Brush" />
54 <xs:attributeGroup ref="AG_SolidColorBrush" />
55 </xs:complexType>
56
57 <xs:complexType name="CT_ImageBrush">
58 <xs:sequence>

```

## B XPS Document Schema

```
1         <xs:element ref="ImageBrush.Transform" minOccurs="0" />
2     </xs:sequence>
3     <xs:attributeGroup ref="AG_Brush" />
4     <xs:attributeGroup ref="AG_TileBrush" />
5     <xs:attributeGroup ref="AG_ImageBrush" />
6 </xs:complexType>
7
8 <xs:complexType name="CT_VisualBrush">
9     <xs:sequence>
10         <xs:element ref="VisualBrush.Transform" minOccurs="0" />
11         <xs:element ref="VisualBrush.Visual" minOccurs="0" />
12     </xs:sequence>
13     <xs:attributeGroup ref="AG_Brush" />
14     <xs:attributeGroup ref="AG_TileBrush" />
15     <xs:attributeGroup ref="AG_VisualBrush" />
16 </xs:complexType>
17
18 <xs:complexType name="CT_LinearGradientBrush">
19     <xs:sequence>
20         <xs:element ref="LinearGradientBrush.Transform" minOccurs="0" />
21         <xs:element ref="LinearGradientBrush.GradientStops" />
22     </xs:sequence>
23     <xs:attributeGroup ref="AG_Brush" />
24     <xs:attributeGroup ref="AG_GradientBrush" />
25     <xs:attributeGroup ref="AG_LinearGradientBrush" />
26 </xs:complexType>
27
28 <xs:complexType name="CT_RadialGradientBrush">
29     <xs:sequence>
30         <xs:element ref="RadialGradientBrush.Transform" minOccurs="0" />
31         <xs:element ref="RadialGradientBrush.GradientStops" />
32     </xs:sequence>
33     <xs:attributeGroup ref="AG_Brush" />
34     <xs:attributeGroup ref="AG_GradientBrush" />
35     <xs:attributeGroup ref="AG_RadialGradientBrush" />
36 </xs:complexType>
37
38 <xs:complexType name="CT_GradientStop">
39     <xs:attributeGroup ref="AG_GradientStop" />
40 </xs:complexType>
41
42 <xs:complexType name="CT_PathGeometry">
43     <xs:sequence>
44         <xs:element ref="PathGeometry.Transform" minOccurs="0" />
45         <xs:element ref="PathFigure" minOccurs="0" maxOccurs="unbounded" />
46     </xs:sequence>
47     <xs:attributeGroup ref="AG_PathGeometry" />
48 </xs:complexType>
49
50 <xs:complexType name="CT_Glyphs">
51     <xs:sequence>
52         <xs:element ref="Glyphs.RenderTransform" minOccurs="0" />
53         <xs:element ref="Glyphs.Clip" minOccurs="0" />
54         <xs:element ref="Glyphs.OpacityMask" minOccurs="0" />
55         <xs:element ref="Glyphs.Fill" minOccurs="0" />
56     </xs:sequence>
57     <xs:attributeGroup ref="AG_Glyphs" />
58 </xs:complexType>
59
60 <xs:complexType name="CT_Path">
61     <xs:sequence>
62         <xs:element ref="Path.RenderTransform" minOccurs="0" />
```

```

1         <xs:element ref="Path.Clip" minOccurs="0" />
2         <xs:element ref="Path.OpacityMask" minOccurs="0" />
3         <xs:element ref="Path.Fill" minOccurs="0" />
4         <xs:element ref="Path.Stroke" minOccurs="0" />
5         <xs:element ref="Path.Data" minOccurs="0" />
6     </xs:sequence>
7     <xs:attributeGroup ref="AG_Path" />
8     <xs:attributeGroup ref="AG_AutomationProvider" />
9     <xs:attributeGroup ref="AG_SnapsToDevicePixels" />
10 </xs:complexType>
11
12 <xs:complexType name="CT_PathFigure">
13     <xs:sequence>
14         <xs:choice maxOccurs="unbounded">
15             <xs:element ref="PolyLineSegment" />
16             <xs:element ref="PolyBezierSegment" />
17             <xs:element ref="ArcSegment" />
18             <xs:element ref="PolyQuadraticBezierSegment" />
19         </xs:choice>
20     </xs:sequence>
21     <xs:attributeGroup ref="AG_PathFigure" />
22 </xs:complexType>
23
24 <xs:complexType name="CT_ArcSegment">
25     <xs:attributeGroup ref="AG_ArcSegment" />
26 </xs:complexType>
27
28 <xs:complexType name="CT_PolyQuadraticBezierSegment">
29     <xs:attributeGroup ref="AG_PolyQuadraticBezierSegment" />
30 </xs:complexType>
31
32 <xs:complexType name="CT_PolyLineSegment">
33     <xs:attributeGroup ref="AG_PolyLineSegment" />
34 </xs:complexType>
35
36 <xs:complexType name="CT_PolyBezierSegment">
37     <xs:attributeGroup ref="AG_PolyBezierSegment" />
38 </xs:complexType>
39
40 <xs:complexType name="CT_Canvas">
41     <xs:sequence>
42         <xs:element ref="Canvas.Resources" minOccurs="0" />
43         <xs:element ref="Canvas.RenderTransform" minOccurs="0" />
44         <xs:element ref="Canvas.Clip" minOccurs="0" />
45         <xs:element ref="Canvas.OpacityMask" minOccurs="0" />
46         <xs:choice minOccurs="0" maxOccurs="unbounded">
47             <xs:element ref="Path" />
48             <xs:element ref="Glyphs" />
49             <xs:element ref="Canvas" />
50         </xs:choice>
51     </xs:sequence>
52     <xs:attributeGroup ref="AG_Canvas" />
53     <xs:attributeGroup ref="AG_AutomationProvider" />
54 </xs:complexType>
55
56 <xs:complexType name="CT_ResourceDictionary">
57     <xs:choice minOccurs="0" maxOccurs="unbounded">
58         <xs:element ref="ImageBrush" />
59         <xs:element ref="LinearGradientBrush" />
60         <xs:element ref="RadialGradientBrush" />
61         <xs:element ref="VisualBrush" />
62         <xs:element ref="SolidColorBrush" />
63         <xs:element ref="MatrixTransform" />

```

## B XPS Document Schema

```
1         <xs:element ref="PathGeometry" />
2         <xs:element ref="Path" />
3         <xs:element ref="Glyphs" />
4         <xs:element ref="Canvas" />
5     </xs:choice>
6     <xs:attributeGroup ref="AG_ResourceDictionary" />
7 </xs:complexType>
8
9 <xs:complexType name="CT_FixedPage">
10     <xs:sequence>
11         <xs:element ref="FixedPage.Resources" minOccurs="0" />
12         <xs:choice minOccurs="0" maxOccurs="unbounded">
13             <xs:element ref="Path" />
14             <xs:element ref="Glyphs" />
15             <xs:element ref="Canvas" />
16         </xs:choice>
17     </xs:sequence>
18     <xs:attributeGroup ref="AG_FixedPage" />
19 </xs:complexType>
20
21 <xs:complexType name="CT_FixedDocument">
22     <xs:sequence>
23         <xs:element ref="PageContent" maxOccurs="unbounded" />
24     </xs:sequence>
25 </xs:complexType>
26
27 <xs:complexType name="CT_PageContent">
28     <xs:sequence>
29         <xs:element ref="PageContent.LinkTargets" minOccurs="0" />
30     </xs:sequence>
31     <xs:attributeGroup ref="AG_PageContent" />
32 </xs:complexType>
33
34 <xs:complexType name="CT_FixedDocumentSequence">
35     <xs:sequence>
36         <xs:element ref="DocumentReference" maxOccurs="unbounded" />
37     </xs:sequence>
38 </xs:complexType>
39
40 <xs:complexType name="CT_DocumentReference">
41     <xs:attributeGroup ref="AG_DocumentReference" />
42 </xs:complexType>
43
44 <xs:complexType name="CT_LinkTarget">
45     <xs:attributeGroup ref="AG_LinkTarget" />
46 </xs:complexType>
47
48 <xs:complexType name="CT_CP_LinkTargets">
49     <xs:sequence>
50         <xs:element ref="LinkTarget" maxOccurs="unbounded" />
51     </xs:sequence>
52 </xs:complexType>
53
54 <xs:complexType name="CT_CP_Transform">
55     <xs:sequence>
56         <xs:element ref="MatrixTransform" />
57     </xs:sequence>
58 </xs:complexType>
59
60 <xs:complexType name="CT_CP_Visual">
61     <xs:choice>
62         <xs:element ref="Path" />
```

```

1         <xs:element ref="Glyphs" />
2         <xs:element ref="Canvas" />
3     </xs:choice>
4 </xs:complexType>
5
6 <xs:complexType name="CT_CP_GradientStops">
7     <xs:sequence>
8         <xs:element ref="GradientStop" minOccurs="2" maxOccurs="unbounded"
9 />
10    </xs:sequence>
11 </xs:complexType>
12
13 <xs:complexType name="CT_CP_Geometry">
14     <xs:sequence>
15         <xs:element ref="PathGeometry" />
16     </xs:sequence>
17 </xs:complexType>
18
19 <xs:complexType name="CT_CP_Brush">
20     <xs:choice>
21         <xs:element ref="ImageBrush" />
22         <xs:element ref="LinearGradientBrush" />
23         <xs:element ref="RadialGradientBrush" />
24         <xs:element ref="SolidColorBrush" />
25         <xs:element ref="VisualBrush" />
26     </xs:choice>
27 </xs:complexType>
28
29 <xs:complexType name="CT_CP_Resources">
30     <xs:sequence minOccurs="0">
31         <xs:element ref="ResourceDictionary" />
32     </xs:sequence>
33 </xs:complexType>
34
35 <!-- Root elements -->
36 <xs:element name="MatrixTransform" type="CT_MatrixTransform">
37     </xs:element>
38
39 <xs:element name="SolidColorBrush" type="CT_SolidColorBrush">
40     </xs:element>
41
42 <xs:element name="ImageBrush" type="CT_ImageBrush">
43     </xs:element>
44
45 <xs:element name="VisualBrush" type="CT_VisualBrush">
46     </xs:element>
47
48 <xs:element name="LinearGradientBrush" type="CT_LinearGradientBrush">
49     </xs:element>
50
51 <xs:element name="RadialGradientBrush" type="CT_RadialGradientBrush">
52     </xs:element>
53
54 <xs:element name="Glyphs" type="CT_Glyphs">
55     </xs:element>
56
57 <xs:element name="Path" type="CT_Path">
58     </xs:element>
59
60 <xs:element name="Canvas" type="CT_Canvas">
61     </xs:element>
62
63 <xs:element name="GradientStop" type="CT_GradientStop">

```

## B XPS Document Schema

```
1         </xs:element>
2
3     <xs:element name="ResourceDictionary" type="CT_ResourceDictionary">
4         </xs:element>
5
6     <xs:element name="PathGeometry" type="CT_PathGeometry">
7         </xs:element>
8
9     <xs:element name="PathFigure" type="CT_PathFigure">
10        </xs:element>
11
12    <xs:element name="PolyLineSegment" type="CT_PolyLineSegment">
13        </xs:element>
14
15    <xs:element name="ArcSegment" type="CT_ArcSegment">
16        </xs:element>
17
18    <xs:element name="PolyBezierSegment" type="CT_PolyBezierSegment">
19        </xs:element>
20
21    <xs:element name="PolyQuadraticBezierSegment"
22    type="CT_PolyQuadraticBezierSegment">
23        </xs:element>
24
25    <xs:element name="FixedPage" type="CT_FixedPage">
26        </xs:element>
27
28    <xs:element name="FixedDocument" type="CT_FixedDocument">
29        </xs:element>
30
31    <xs:element name="PageContent" type="CT_PageContent">
32        </xs:element>
33
34    <xs:element name="FixedDocumentSequence" type="CT_FixedDocumentSequence">
35        </xs:element>
36
37    <xs:element name="DocumentReference" type="CT_DocumentReference">
38        </xs:element>
39
40    <xs:element name="LinkTarget" type="CT_LinkTarget">
41        </xs:element>
42
43    <xs:element name="PageContent.LinkTargets" type="CT_CP_LinkTargets">
44        </xs:element>
45
46    <xs:element name="ImageBrush.Transform" type="CT_CP_Transform">
47        </xs:element>
48
49    <xs:element name="VisualBrush.Transform" type="CT_CP_Transform">
50        </xs:element>
51
52    <xs:element name="LinearGradientBrush.Transform" type="CT_CP_Transform">
53        </xs:element>
54
55    <xs:element name="RadialGradientBrush.Transform" type="CT_CP_Transform">
56        </xs:element>
57
58    <xs:element name="PathGeometry.Transform" type="CT_CP_Transform">
59        </xs:element>
60
61    <xs:element name="Glyphs.RenderTransform" type="CT_CP_Transform">
62        </xs:element>
```



```

1
2     <xs:element name="Path.RenderTransform" type="CT_CP_Transform">
3         </xs:element>
4
5     <xs:element name="Canvas.RenderTransform" type="CT_CP_Transform">
6         </xs:element>
7
8     <xs:element name="VisualBrush.Visual" type="CT_CP_Visual">
9         </xs:element>
10
11     <xs:element name="LinearGradientBrush.GradientStops"
12 type="CT_CP_GradientStops">
13         </xs:element>
14
15     <xs:element name="RadialGradientBrush.GradientStops"
16 type="CT_CP_GradientStops">
17         </xs:element>
18
19     <xs:element name="Glyphs.Clip" type="CT_CP_Geometry">
20         </xs:element>
21
22     <xs:element name="Path.Clip" type="CT_CP_Geometry">
23         </xs:element>
24
25     <xs:element name="Canvas.Clip" type="CT_CP_Geometry">
26         </xs:element>
27
28     <xs:element name="Glyphs.OpacityMask" type="CT_CP_Brush">
29         </xs:element>
30
31     <xs:element name="Path.OpacityMask" type="CT_CP_Brush">
32         </xs:element>
33
34     <xs:element name="Canvas.OpacityMask" type="CT_CP_Brush">
35         </xs:element>
36
37     <xs:element name="Glyphs.Fill" type="CT_CP_Brush">
38         </xs:element>
39
40     <xs:element name="Path.Fill" type="CT_CP_Brush">
41         </xs:element>
42
43     <xs:element name="Path.Data" type="CT_CP_Geometry">
44         </xs:element>
45
46     <xs:element name="Path.Stroke" type="CT_CP_Brush">
47         </xs:element>
48
49     <xs:element name="Canvas.Resources" type="CT_CP_Resources">
50         </xs:element>
51
52     <xs:element name="FixedPage.Resources" type="CT_CP_Resources">
53         </xs:element>
54
55     <xs:attributeGroup name="AG_GradientStop">
56         <xs:attribute name="Color" type="ST_Color" use="required">
57             </xs:attribute>
58         <xs:attribute name="Offset" type="ST_Double" use="required">
59             </xs:attribute>
60     </xs:attributeGroup>
61
62     <xs:attributeGroup name="AG_Brush">
63         <xs:attribute name="Opacity" type="ST_ZeroOne" default="1.0">

```

## B XPS Document Schema

```
1         </xs:attribute>
2         <xs:attribute ref="x:Key" />
3     </xs:attributeGroup>
4
5     <xs:attributeGroup name="AG_TileBrush">
6         <xs:attribute name="Transform" type="ST_RscRefMatrix">
7             </xs:attribute>
8         <xs:attribute name="Viewbox" type="ST_ViewBox" use="required">
9             </xs:attribute>
10        <xs:attribute name="Viewport" type="ST_ViewBox" use="required">
11            </xs:attribute>
12        <xs:attribute name="TileMode" type="ST_TileMode" default="None">
13            </xs:attribute>
14        <xs:attribute name="ViewboxUnits" type="ST_ViewUnits" use="required"
15        fixed="Absolute">
16            </xs:attribute>
17        <xs:attribute name="ViewportUnits" type="ST_ViewUnits" use="required"
18        fixed="Absolute">
19            </xs:attribute>
20    </xs:attributeGroup>
21
22    <xs:attributeGroup name="AG_VisualBrush">
23        <xs:attribute name="Visual" type="ST_RscRef">
24            </xs:attribute>
25    </xs:attributeGroup>
26
27    <xs:attributeGroup name="AG_GradientBrush">
28        <xs:attribute name="ColorInterpolationMode" type="ST_ClrIntMode"
29        default="SRgbLinearInterpolation">
30            </xs:attribute>
31        <xs:attribute name="SpreadMethod" type="ST_SpreadMethod" default="Pad">
32            </xs:attribute>
33        <xs:attribute name="MappingMode" type="ST_MappingMode" use="required"
34        fixed="Absolute">
35            </xs:attribute>
36    </xs:attributeGroup>
37
38    <xs:attributeGroup name="AG_SolidColorBrush">
39        <xs:attribute name="Color" type="ST_Color" use="required">
40            </xs:attribute>
41    </xs:attributeGroup>
42
43    <xs:attributeGroup name="AG_ImageBrush">
44        <xs:attribute name="ImageSource" type="ST_UriCtxBmp" use="required">
45            </xs:attribute>
46    </xs:attributeGroup>
47
48    <xs:attributeGroup name="AG_LinearGradientBrush">
49        <xs:attribute name="Transform" type="ST_RscRefMatrix">
50            </xs:attribute>
51        <xs:attribute name="StartPoint" type="ST_Point" use="required">
52            </xs:attribute>
53        <xs:attribute name="EndPoint" type="ST_Point" use="required">
54            </xs:attribute>
55    </xs:attributeGroup>
56
57    <xs:attributeGroup name="AG_RadialGradientBrush">
58        <xs:attribute name="Transform" type="ST_RscRefMatrix">
59            </xs:attribute>
60        <xs:attribute name="Center" type="ST_Point" use="required">
61            </xs:attribute>
62        <xs:attribute name="GradientOrigin" type="ST_Point" use="required">
```

```

1         </xs:attribute>
2         <xs:attribute name="RadiusX" type="ST_GEZero" use="required">
3         </xs:attribute>
4         <xs:attribute name="RadiusY" type="ST_GEZero" use="required">
5         </xs:attribute>
6     </xs:attributeGroup>
7
8     <xs:attributeGroup name="AG_PathGeometry">
9         <xs:attribute name="Figures" type="ST_AbbrGeom">
10        </xs:attribute>
11        <xs:attribute name="FillRule" type="ST_FillRule" default="EvenOdd">
12        </xs:attribute>
13        <xs:attribute name="Transform" type="ST_RscRefMatrix">
14        </xs:attribute>
15        <xs:attribute ref="x:Key" />
16    </xs:attributeGroup>
17
18    <xs:attributeGroup name="AG_ResourceDictionary">
19        <xs:attribute name="Source" type="xs:anyURI">
20        </xs:attribute>
21    </xs:attributeGroup>
22
23    <xs:attributeGroup name="AG_PolyLineSegment">
24        <xs:attribute name="Points" type="ST_Points" use="required">
25        </xs:attribute>
26        <xs:attribute name="IsStroked" type="ST_Boolean" default="true">
27        </xs:attribute>
28    </xs:attributeGroup>
29
30    <xs:attributeGroup name="AG_ArcSegment">
31        <xs:attribute name="Point" type="ST_Point" use="required">
32        </xs:attribute>
33        <xs:attribute name="Size" type="ST_PointGE0" use="required">
34        </xs:attribute>
35        <xs:attribute name="RotationAngle" type="ST_Double" use="required">
36        </xs:attribute>
37        <xs:attribute name="IsLargeArc" type="ST_Boolean" use="required">
38        </xs:attribute>
39        <xs:attribute name="SweepDirection" type="ST_SweepDirection"
40        use="required">
41        </xs:attribute>
42        <xs:attribute name="IsStroked" type="ST_Boolean" default="true">
43        </xs:attribute>
44    </xs:attributeGroup>
45
46    <xs:attributeGroup name="AG_PolyBezierSegment">
47        <xs:attribute name="Points" type="ST_Points" use="required">
48        </xs:attribute>
49        <xs:attribute name="IsStroked" type="ST_Boolean" default="true">
50        </xs:attribute>
51    </xs:attributeGroup>
52
53    <xs:attributeGroup name="AG_PolyQuadraticBezierSegment">
54        <xs:attribute name="Points" type="ST_Points" use="required">
55        </xs:attribute>
56        <xs:attribute name="IsStroked" type="ST_Boolean" default="true">
57        </xs:attribute>
58    </xs:attributeGroup>
59
60    <xs:attributeGroup name="AG_Glyphs">
61        <xs:attribute name="BidiLevel" default="0">
62        <xs:simpleType>
63            <xs:restriction base="xs:integer">

```

## B XPS Document Schema

```
1         <xs:minInclusive value="0" />
2         <xs:maxInclusive value="61" />
3     </xs:restriction>
4 </xs:simpleType>
5 </xs:attribute>
6 <xs:attribute name="CaretStops" type="ST_CaretStops">
7 </xs:attribute>
8 <xs:attribute name="DeviceFontName" type="ST_UnicodeString">
9 </xs:attribute>
10 <xs:attribute name="Fill" type="ST_RscRefColor">
11 </xs:attribute>
12 <xs:attribute name="FontRenderingEmSize" type="ST_GEZero"
13 use="required">
14 </xs:attribute>
15 <xs:attribute name="FontUri" type="xs:anyURI" use="required">
16 </xs:attribute>
17 <xs:attribute name="OriginX" type="ST_Double" use="required">
18 </xs:attribute>
19 <xs:attribute name="OriginY" type="ST_Double" use="required">
20 </xs:attribute>
21 <xs:attribute name="IsSideways" type="ST_Boolean" default="false">
22 </xs:attribute>
23 <xs:attribute name="Indices" type="ST_Indices">
24 </xs:attribute>
25 <xs:attribute name="UnicodeString" type="ST_UnicodeString">
26 </xs:attribute>
27 <xs:attribute name="StyleSimulations" type="ST_StyleSimulations"
28 default="None">
29 </xs:attribute>
30 <xs:attribute name="RenderTransform" type="ST_RscRefMatrix">
31 </xs:attribute>
32 <xs:attribute name="Clip" type="ST_RscRefAbbrGeomF">
33 </xs:attribute>
34 <xs:attribute name="Opacity" type="ST_ZeroOne" default="1.0">
35 </xs:attribute>
36 <xs:attribute name="OpacityMask" type="ST_RscRef">
37 </xs:attribute>
38 <xs:attribute name="Name" type="ST_Name">
39 </xs:attribute>
40 <xs:attribute name="FixedPage.NavigateUri" type="xs:anyURI">
41 </xs:attribute>
42 <xs:attribute ref="xml:lang">
43 </xs:attribute>
44 <xs:attribute ref="x:Key" />
45 </xs:attributeGroup>
46
47 <xs:attributeGroup name="AG_Path">
48 <xs:attribute name="Data" type="ST_RscRefAbbrGeomF">
49 </xs:attribute>
50 <xs:attribute name="Fill" type="ST_RscRefColor">
51 </xs:attribute>
52 <xs:attribute name="RenderTransform" type="ST_RscRefMatrix">
53 </xs:attribute>
54 <xs:attribute name="Clip" type="ST_RscRefAbbrGeomF">
55 </xs:attribute>
56 <xs:attribute name="Opacity" type="ST_ZeroOne" default="1.0">
57 </xs:attribute>
58 <xs:attribute name="OpacityMask" type="ST_RscRef">
59 </xs:attribute>
60 <xs:attribute name="Stroke" type="ST_RscRefColor">
61 </xs:attribute>
62 <xs:attribute name="StrokeDashArray" type="ST_EvenArrayPos">
```

```

1      </xs:attribute>
2      <xs:attribute name="StrokeDashCap" type="ST_DashCap" default="Flat">
3      </xs:attribute>
4      <xs:attribute name="StrokeDashOffset" type="ST_Double" default="0.0">
5      </xs:attribute>
6      <xs:attribute name="StrokeEndLineCap" type="ST_LineCap" default="Flat">
7      </xs:attribute>
8      <xs:attribute name="StrokeStartLineCap" type="ST_LineCap"
9      default="Flat">
10     </xs:attribute>
11     <xs:attribute name="StrokeLineJoin" type="ST_LineJoin" default="Miter">
12     </xs:attribute>
13     <xs:attribute name="StrokeMiterLimit" type="ST_GEOne" default="10.0">
14     </xs:attribute>
15     <xs:attribute name="StrokeThickness" type="ST_GEZero" default="1.0">
16     </xs:attribute>
17     <xs:attribute name="Name" type="ST_Name">
18     </xs:attribute>
19     <xs:attribute name="FixedPage.NavigateUri" type="xs:anyURI">
20     </xs:attribute>
21     <xs:attribute ref="xml:lang">
22     </xs:attribute>
23     <xs:attribute ref="x:Key" />
24 </xs:attributeGroup>
25
26 <xs:attributeGroup name="AG_PathFigure">
27   <xs:attribute name="IsClosed" type="ST_Boolean" default="false">
28   </xs:attribute>
29   <xs:attribute name="StartPoint" type="ST_Point" use="required">
30   </xs:attribute>
31   <xs:attribute name="IsFilled" type="ST_Boolean" default="true">
32   </xs:attribute>
33 </xs:attributeGroup>
34
35 <xs:attributeGroup name="AG_Canvas">
36   <xs:attribute name="RenderTransform" type="ST_RscRefMatrix">
37   </xs:attribute>
38   <xs:attribute name="Clip" type="ST_RscRefAbbrGeomF">
39   </xs:attribute>
40   <xs:attribute name="Opacity" type="ST_ZeroOne" default="1.0">
41   </xs:attribute>
42   <xs:attribute name="OpacityMask" type="ST_RscRef">
43   </xs:attribute>
44   <xs:attribute name="Name" type="ST_Name">
45   </xs:attribute>
46   <xs:attribute name="RenderOptions.EdgeMode" type="ST_EdgeMode">
47   </xs:attribute>
48   <xs:attribute name="FixedPage.NavigateUri" type="xs:anyURI">
49   </xs:attribute>
50   <xs:attribute ref="xml:lang">
51   </xs:attribute>
52   <xs:attribute ref="x:Key" />
53 </xs:attributeGroup>
54
55 <xs:attributeGroup name="AG_PageContent">
56   <xs:attribute name="Source" type="xs:anyURI" use="required">
57   </xs:attribute>
58   <xs:attribute name="Width" type="ST_GEOne">
59   </xs:attribute>
60   <xs:attribute name="Height" type="ST_GEOne">
61   </xs:attribute>
62 </xs:attributeGroup>
63

```

## B XPS Document Schema

```
1 <xs:attributeGroup name="AG_LinkTarget">
2   <xs:attribute name="Name" type="ST_NUName" use="required">
3     </xs:attribute>
4 </xs:attributeGroup>
5
6 <xs:attributeGroup name="AG_DocumentReference">
7   <xs:attribute name="Source" type="xs:anyURI" use="required">
8     </xs:attribute>
9 </xs:attributeGroup>
10
11 <xs:attributeGroup name="AG_MatrixTransform">
12   <xs:attribute name="Matrix" type="ST_Matrix" use="required">
13     </xs:attribute>
14   <xs:attribute ref="x:Key" />
15 </xs:attributeGroup>
16
17 <xs:attributeGroup name="AG_FixedPage">
18   <xs:attribute name="Width" type="ST_GEOne" use="required">
19     </xs:attribute>
20   <xs:attribute name="Height" type="ST_GEOne" use="required">
21     </xs:attribute>
22   <xs:attribute name="ContentBox" type="ST_ContentBox">
23     </xs:attribute>
24   <xs:attribute name="BleedBox" type="ST_BleedBox">
25     </xs:attribute>
26   <xs:attribute ref="xml:lang" use="required">
27     </xs:attribute>
28   <xs:attribute name="Name" type="ST_Name">
29     </xs:attribute>
30 </xs:attributeGroup>
31
32 <xs:attributeGroup name="AG_AutomationProvider">
33   <xs:attribute name="AutomationProperties.Name" type="xs:string">
34     </xs:attribute>
35   <xs:attribute name="AutomationProperties.HelpText" type="xs:string">
36     </xs:attribute>
37 </xs:attributeGroup>
38
39 <xs:attributeGroup name="AG_SnapsToDevicePixels">
40   <xs:attribute name="SnapsToDevicePixels" type="ST_Boolean">
41     </xs:attribute>
42 </xs:attributeGroup>
43
44 <!-- Simple data types -->
45 <!-- A unique Name (ID with pattern restriction according to XPS spec) -->
46 <xs:simpleType name="ST_Name">
47   <xs:restriction base="xs:ID">
48     <xs:pattern
49 value="(\p{Lu}|\p{Ll}|\p{Lt}|\p{Lo}|\p{Nl}|_)(\p{Lu}|\p{Ll}|\p{Lt}|\p{Lo}|\p{Nl}
50 )|\p{Mn}|\p{Mc}|\p{Nd}|_)*" />
51   </xs:restriction>
52 </xs:simpleType>
53
54 <!-- A non-unique Name (ID with pattern restriction according to XPS spec)
55 -->
56 <xs:simpleType name="ST_NUName">
57   <xs:restriction base="xs:string">
58     <xs:pattern
59 value="(\p{Lu}|\p{Ll}|\p{Lt}|\p{Lo}|\p{Nl}|_)(\p{Lu}|\p{Ll}|\p{Lt}|\p{Lo}|\p{Nl}
60 )|\p{Mn}|\p{Mc}|\p{Nd}|_)*" />
61   </xs:restriction>
62 </xs:simpleType>
```

```

1
2     <!-- Boolean with true and false only (no 0 or 1) -->
3     <xs:simpleType name="ST_Boolean">
4         <xs:restriction base="xs:boolean">
5             <xs:pattern value="true|false" />
6         </xs:restriction>
7     </xs:simpleType>
8
9     <!-- real number from 0.0 to 1.0 inclusive -->
10    <xs:simpleType name="ST_ZeroOne">
11        <xs:restriction base="ST_Double">
12            <xs:minInclusive value="0.0" />
13            <xs:maxInclusive value="1.0" />
14        </xs:restriction>
15    </xs:simpleType>
16
17    <!-- positive real number -->
18    <xs:simpleType name="ST_GEZero">
19        <xs:restriction base="ST_Double">
20            <xs:minInclusive value="0.0" />
21        </xs:restriction>
22    </xs:simpleType>
23
24    <!-- positive real number, equal or greater than one -->
25    <xs:simpleType name="ST_GEOne">
26        <xs:restriction base="ST_Double">
27            <xs:minInclusive value="1.0" />
28        </xs:restriction>
29    </xs:simpleType>
30
31    <!-- Double -->
32    <xs:simpleType name="ST_Double">
33        <xs:restriction base="xs:double">
34            <xs:whiteSpace value="collapse" />
35    <!--
36        <xs:pattern value="[rn]" />
37    -->
38        <xs:pattern value="((\-|\+)?((([0-9]+)(\.[0-9]+)?)|(\.[0-
39    9]+)))(e|E)(\-|\+)?[0-9]+)?" />
40        </xs:restriction>
41    </xs:simpleType>
42
43    <!-- Point: 2 numbers, separated by , and arbitrary whitespace -->
44    <xs:simpleType name="ST_Point">
45        <xs:restriction base="xs:string">
46            <xs:whiteSpace value="collapse" />
47    <!--
48        <xs:pattern value="[rn][scs][rn]" />
49    -->
50        <xs:pattern value="((\-|\+)?((([0-9]+)(\.[0-9]+)?)|(\.[0-
51    9]+)))(e|E)(\-|\+)?[0-9]+)?( ?, ?)((\-|\+)?((([0-9]+)(\.[0-9]+)?)|(\.[0-
52    9]+)))(e|E)(\-|\+)?[0-9]+)" />
53        </xs:restriction>
54    </xs:simpleType>
55
56    <!-- PointGE0: 2 non-negative numbers, separated by , and arbitrary
57    whitespace -->
58    <xs:simpleType name="ST_PointGE0">
59        <xs:restriction base="xs:string">
60            <xs:whiteSpace value="collapse" />
61    <!--
62        <xs:pattern value="[prn][scs][prn]" />
63    -->

```

## B XPS Document Schema

```

1         <xs:pattern value="(\+?([0-9]+(\.[0-9]+)?)|\.[0-9]+)((e|E)(\+
2 |\+)?[0-9]+)?( ? , ?)(\+?([0-9]+(\.[0-9]+)?)|\.[0-9]+)((e|E)(\+|\+)?[0-
3 9]+)?)" />
4     </xs:restriction>
5 </xs:simpleType>
6
7     <!-- Points: List of ST_Point, separated by arbitrary whitespace -->
8     <xs:simpleType name="ST_Points">
9         <xs:restriction base="xs:string">
10            <xs:whiteSpace value="collapse" />
11 <!--
12            <xs:pattern value="[rn][scs][rn]([rn][scs][rn])*/>
13 -->
14            <xs:pattern value="((\+|\+)?([0-9]+(\.[0-9]+)?)|\.[0-
15 9]+)((e|E)(\+|\+)?[0-9]+)?( ? , ?)((\+|\+)?([0-9]+(\.[0-9]+)?)|\.[0-
16 9]+)((e|E)(\+|\+)?[0-9]+)?( (\+|\+)?([0-9]+(\.[0-9]+)?)|\.[0-
17 9]+)((e|E)(\+|\+)?[0-9]+)?( ? , ?)((\+|\+)?([0-9]+(\.[0-9]+)?)|\.[0-
18 9]+)((e|E)(\+|\+)?[0-9]+)?)" />
19        </xs:restriction>
20    </xs:simpleType>
21
22    <!-- EvenArray: List with even number of entries of non-negative numbers.
23 -->
24    <xs:simpleType name="ST_EvenArrayPos">
25        <xs:restriction base="xs:string">
26            <xs:whiteSpace value="collapse" />
27 <!--
28            <xs:pattern value="[prn] [prn]([prn] [prn])*/>
29 -->
30            <xs:pattern value="(\+?([0-9]+(\.[0-9]+)?)|\.[0-9]+)((e|E)(\+
31 |\+)?[0-9]+)? (\+?([0-9]+(\.[0-9]+)?)|\.[0-9]+)((e|E)(\+|\+)?[0-9]+)? (
32 \+?([0-9]+(\.[0-9]+)?)|\.[0-9]+)((e|E)(\+|\+)?[0-9]+)? (\+?([0-9]+(\.[0-
33 9]+)?)|\.[0-9]+)((e|E)(\+|\+)?[0-9]+)?)" />
34        </xs:restriction>
35    </xs:simpleType>
36
37    <!-- Array: List of numbers. -->
38    <xs:simpleType name="ST_Array">
39        <xs:restriction base="xs:string">
40            <xs:whiteSpace value="collapse" />
41 <!--
42            <xs:pattern value="([rn] ?)*/>
43 -->
44            <xs:pattern value="((\+|\+)?([0-9]+(\.[0-9]+)?)|\.[0-
45 9]+)((e|E)(\+|\+)?[0-9]+)? ?)" />
46        </xs:restriction>
47    </xs:simpleType>
48
49    <!-- ViewBox: 4 numbers, separated by , and arbitrary whitespace. Second
50 number pair must be non-negative -->
51    <xs:simpleType name="ST_ViewBox">
52        <xs:restriction base="xs:string">
53            <xs:whiteSpace value="collapse" />
54 <!--
55            <xs:pattern value="[rn][scs][rn][scs][prn][scs][prn]*/>
56 -->
57            <xs:pattern value="((\+|\+)?([0-9]+(\.[0-9]+)?)|\.[0-
58 9]+)((e|E)(\+|\+)?[0-9]+)?( ? , ?)((\+|\+)?([0-9]+(\.[0-9]+)?)|\.[0-
59 9]+)((e|E)(\+|\+)?[0-9]+)?( ? , ?)(\+?([0-9]+(\.[0-9]+)?)|\.[0-
60 9]+)((e|E)(\+|\+)?[0-9]+)?( ? , ?)(\+?([0-9]+(\.[0-9]+)?)|\.[0-
61 9]+)((e|E)(\+|\+)?[0-9]+)?)" />
62        </xs:restriction>

```



```

1      </xs:simpleType>
2
3      <!-- ContentBox: 4 non-negative numbers, separated by commas and arbitrary
4      whitespace -->
5      <xs:simpleType name="ST_ContentBox">
6          <xs:restriction base="xs:string">
7              <xs:whiteSpace value="collapse" />
8      <!--
9          <xs:pattern value="[prn][scs][prn][scs][prn][scs][prn]" />
10     -->
11         <xs:pattern value="(\+?(((0-9)+(\.[0-9]+)?)|(\.[0-9]+))((e|E)(\-|
12     |\\+)?[0-9]+)?( ? , ?)(\+?(((0-9)+(\.[0-9]+)?)|(\.[0-9]+))((e|E)(\-|\\+)?[0-
13     9]+)?( ? , ?)(\+?(((0-9)+(\.[0-9]+)?)|(\.[0-9]+))((e|E)(\-|\\+)?[0-9]+)?( ? ,
14     ?)(\+?(((0-9)+(\.[0-9]+)?)|(\.[0-9]+))((e|E)(\-|\\+)?[0-9]+)?)" />
15     </xs:restriction>
16 </xs:simpleType>
17
18 <!-- BleedBox: 4 numbers, separated by , and arbitrary whitespace. Second
19 number pair must be non-negative -->
20 <xs:simpleType name="ST_BleedBox">
21     <xs:restriction base="xs:string">
22         <xs:whiteSpace value="collapse" />
23 <!--
24     <xs:pattern value="[rn][scs][rn][scs][prn][scs][prn]" />
25 -->
26     <xs:pattern value="((\-|\\+)?(((0-9)+(\.[0-9]+)?)|(\.[0-
27     9]+))((e|E)(\-|\\+)?[0-9]+)?( ? , ?)(\-|\\+)?(((0-9)+(\.[0-9]+)?)|(\.[0-
28     9]+))((e|E)(\-|\\+)?[0-9]+)?( ? , ?)(\+?(((0-9)+(\.[0-9]+)?)|(\.[0-
29     9]+))((e|E)(\-|\\+)?[0-9]+)?( ? , ?)(\+?(((0-9)+(\.[0-9]+)?)|(\.[0-
30     9]+))((e|E)(\-|\\+)?[0-9]+)" />
31 </xs:restriction>
32 </xs:simpleType>
33
34 <!-- Bare Matrix form: 6 numbers separated by , and arbitrary whitespace --
35 >
36 <xs:simpleType name="ST_Matrix">
37     <xs:restriction base="xs:string">
38         <xs:whiteSpace value="collapse" />
39 <!--
40     <xs:pattern
41     value="[rn][scs][rn][scs][rn][scs][rn][scs][rn][scs][rn]" />
42 -->
43     <xs:pattern value="((\-|\\+)?(((0-9)+(\.[0-9]+)?)|(\.[0-
44     9]+))((e|E)(\-|\\+)?[0-9]+)?( ? , ?)(\-|\\+)?(((0-9)+(\.[0-9]+)?)|(\.[0-
45     9]+))((e|E)(\-|\\+)?[0-9]+)?( ? , ?)(\-|\\+)?(((0-9)+(\.[0-9]+)?)|(\.[0-
46     9]+))((e|E)(\-|\\+)?[0-9]+)?( ? , ?)(\-|\\+)?(((0-9)+(\.[0-9]+)?)|(\.[0-
47     9]+))((e|E)(\-|\\+)?[0-9]+)?( ? , ?)(\-|\\+)?(((0-9)+(\.[0-9]+)?)|(\.[0-
48     9]+))((e|E)(\-|\\+)?[0-9]+)?( ? , ?)(\-|\\+)?(((0-9)+(\.[0-9]+)?)|(\.[0-
49     9]+))((e|E)(\-|\\+)?[0-9]+)" />
50 </xs:restriction>
51 </xs:simpleType>
52
53 <!-- Color: 6 or 8 hex digits -->
54 <xs:simpleType name="ST_Color">
55     <xs:restriction base="xs:string">
56         <!-- The pattern restriction does not check for sRGB gamut -->
57         <!-- The pattern restriction does not check for color profile URI
58     validity -->
59         <xs:whiteSpace value="collapse" />
60 <!--
61     <xs:pattern value="(#[0-9a-fA-F]{2})?[0-9a-fA-F]{6})|\\
62         (sc# ?[dec][scs][dec][scs][dec]([scs][dec])?)|\\

```

## B XPS Document Schema

```

1             (ContextColor +[\S]+
2 ?[dec] ([scs] [dec]) {3,8})"/>
3 -->
4         <xs:pattern value="(#[0-9a-fA-F]{2})?[0-9a-fA-F]{6}|(sc# ?(\-
5 ?((([0-9]+\.[0-9]+)?) | (\.[0-9]+)) ( ? , ?) (\-?((([0-9]+\.[0-9]+)?) | (\.[0-9]+)) (
6 ? , ?) (\-?((([0-9]+\.[0-9]+)?) | (\.[0-9]+)) ( ( ? , ?) (\-?((([0-9]+\.[0-9]+)?) | (\.[0-
7 9]+\.[0-9]+)?) | (\.[0-9]+)?)?) | (ContextColor +[\S]+ ?(\-?((([0-9]+\.[0-9]+)?) | (\.[0-
8 9]+\.[0-9]+)?) ( ( ? , ?) (\-?((([0-9]+\.[0-9]+)?) | (\.[0-9]+)?)?) {3,8})" />
9         </xs:restriction>
10        </xs:simpleType>
11
12        <!-- Indices grammar for Glyphs.CaretStops -->
13        <xs:simpleType name="ST_CaretStops">
14            <xs:restriction base="xs:string">
15                <xs:whiteSpace value="collapse" />
16                <xs:pattern value="[0-9A-Fa-f]*" />
17            </xs:restriction>
18        </xs:simpleType>
19
20        <!-- Indices grammar for Glyphs.Indices -->
21        <xs:simpleType name="ST_Indices">
22            <xs:restriction base="xs:string">
23                <xs:whiteSpace value="collapse" />
24        <!--
25            <xs:pattern value="(\
26                ((\([[:pint]](:[[:pint]]?)\))?[[:uint]]?)\
27                (, [[:prn]]?(, [[:rn]]?(, [[:rn]]?)?)?)\
28                )\
29                (:\
30                    ((\([[:pint]](:[[:pint]]?)\))?[[:uint]]?)\
31                    (, [[:prn]]?(, [[:rn]]?(, [[:rn]]?)?)?)\
32                )*" />
33        -->
34            <xs:pattern value="(((\([[:1-9]] [0-9]*)(:([[:1-9]] [0-9]*))?)\))?([0-
35 9]+\.)?((\+?((([0-9]+\.[0-9]+)?) | (\.[0-9]+)) ((e|E) (\-|\+)?[0-9]+)?)?((\-
36 |\+)?((([0-9]+\.[0-9]+)?) | (\.[0-9]+)) ((e|E) (\-|\+)?[0-9]+)?)?((\-|\+)?((([0-
37 9]+\.[0-9]+)?) | (\.[0-9]+)) ((e|E) (\-|\+)?[0-9]+)?)?)?)?((\([[:1-9]] [0-
38 9]*)(:([[:1-9]] [0-9]*))?)\))?([0-9]+)?)?((\+?((([0-9]+\.[0-9]+)?) | (\.[0-
39 9]+\.[0-9]+)) ((e|E) (\-|\+)?[0-9]+)?)?((\-|\+)?((([0-9]+\.[0-9]+)?) | (\.[0-
40 9]+\.[0-9]+)) ((e|E) (\-|\+)?[0-9]+)?)?((\-|\+)?((([0-9]+\.[0-9]+)?) | (\.[0-
41 9]+\.[0-9]+)) ((e|E) (\-|\+)?[0-9]+)?)?)?)?" />
42            </xs:restriction>
43        </xs:simpleType>
44
45        <!-- UnicodeString grammar -->
46        <xs:simpleType name="ST_UnicodeString">
47            <xs:restriction base="xs:string">
48                <xs:pattern value="([\^\{\}|\{\})\.*)" />
49            </xs:restriction>
50        </xs:simpleType>
51
52        <!-- Abbreviated Geometry grammar for Path.Data , clip and Geometries -->
53        <xs:simpleType name="ST_AbbrGeomF">
54            <xs:restriction base="xs:string">
55                <xs:whiteSpace value="collapse" />
56        <!--
57            <xs:pattern value="(F ?(0|1))?\
58                ( ?(M|m) ( ?[[:rn]] [scs] [[:rn]]) )\
59                (\
60                    ( ?(M|m) ( ?[[:rn]] [scs] [[:rn]]) ) | \
61                    ( ?(L|l) ( ?[[:rn]] [scs] [[:rn]]) (
62 [[:rn]] [scs] [[:rn]])*) | \

```

```

1           ( ?(H|h|V|v) ( ?[rn]) ( [rn])*) | \
2           ( ?(Q|q|S|s) ( ?[rn][scs][rn]
3 [rn][scs][rn]) ( ( [rn][scs][rn]){2})*) | \
4           ( ?(C|c) ( ?[rn][scs][rn] (
5 [rn][scs][rn]){2}) ( ( [rn][scs][rn]){3})*) | \
6           ( ?(A|a) ( ?[rn][scs][rn] [rn] [0-1] [0-1]
7 [rn][scs][rn]) \
8           ( [rn][scs][rn] [rn] [0-1]
9 [0-1] [rn][scs][rn])*) | \
10          ( ?(Z|z)) \
11          )*/>
12 -->
13 <xs:pattern value="(F ?(0|1))?( ?(M|m) ( ?(\-|\+)?([0-9]+\.[0-
14 9]+)? | (\.[0-9]+)) ((e|E) (\-|\+)?[0-9]+)? ( ? , ?) (\-|\+)?([0-9]+\.[0-
15 9]+)? | (\.[0-9]+)) ((e|E) (\-|\+)?[0-9]+)?)) ( ?(M|m) ( ?(\-|\+)?([0-9]+\.[0-
16 9]+)? | (\.[0-9]+)) ((e|E) (\-|\+)?[0-9]+)? ( ? , ?) (\-|\+)?([0-9]+\.[0-
17 9]+)? | (\.[0-9]+)) ((e|E) (\-|\+)?[0-9]+)?)) | ( ?(L|l) ( ?(\-|\+)?([0-9]+\.[0-
18 9]+)? | (\.[0-9]+)) ((e|E) (\-|\+)?[0-9]+)? ( ? , ?) (\-|\+)?([0-9]+\.[0-
19 9]+)? | (\.[0-9]+)) ((e|E) (\-|\+)?[0-9]+)?)) ( (\-|\+)?([0-9]+\.[0-
20 9]+)? | (\.[0-9]+)) ((e|E) (\-|\+)?[0-9]+)? ( ? , ?) (\-|\+)?([0-9]+\.[0-
21 9]+)? | (\.[0-9]+)) ((e|E) (\-|\+)?[0-9]+)?))* | ( ?(H|h|V|v) ( ?(\-|\+)?([0-
22 9]+\.[0-9]+)? | (\.[0-9]+)) ((e|E) (\-|\+)?[0-9]+)?)) ( (\-|\+)?([0-9]+\.[0-
23 9]+)? | (\.[0-9]+)) ((e|E) (\-|\+)?[0-9]+)?))*) | ( ?(Q|q|S|s) ( ?(\-|\+)?([0-
24 9]+\.[0-9]+)? | (\.[0-9]+)) ((e|E) (\-|\+)?[0-9]+)? ( ? , ?) (\-|\+)?([0-
25 9]+\.[0-9]+)? | (\.[0-9]+)) ((e|E) (\-|\+)?[0-9]+)?)) ( (\-|\+)?([0-9]+\.[0-
26 9]+)? | (\.[0-9]+)) ((e|E) (\-|\+)?[0-9]+)? ( ? , ?) (\-|\+)?([0-9]+\.[0-
27 9]+)? | (\.[0-9]+)) ((e|E) (\-|\+)?[0-9]+)?)) ( ( (\-|\+)?([0-9]+\.[0-
28 9]+)? | (\.[0-9]+)) ((e|E) (\-|\+)?[0-9]+)? ( ? , ?) (\-|\+)?([0-9]+\.[0-
29 9]+)? | (\.[0-9]+)) ((e|E) (\-|\+)?[0-9]+)?)) {2})* | ( ?(C|c) ( ?(\-|\+)?([0-
30 9]+\.[0-9]+)? | (\.[0-9]+)) ((e|E) (\-|\+)?[0-9]+)? ( ? , ?) (\-|\+)?([0-
31 9]+\.[0-9]+)? | (\.[0-9]+)) ((e|E) (\-|\+)?[0-9]+)?)) ( (\-|\+)?([0-9]+\.[0-
32 9]+)? | (\.[0-9]+)) ((e|E) (\-|\+)?[0-9]+)? ( ? , ?) (\-|\+)?([0-9]+\.[0-
33 9]+)? | (\.[0-9]+)) ((e|E) (\-|\+)?[0-9]+)?)) {2}) ( ( (\-|\+)?([0-9]+\.[0-
34 9]+)? | (\.[0-9]+)) ((e|E) (\-|\+)?[0-9]+)? ( ? , ?) (\-|\+)?([0-9]+\.[0-
35 9]+)? | (\.[0-9]+)) ((e|E) (\-|\+)?[0-9]+)?)) {3})* | ( ?(A|a) ( ?(\-|\+)?([0-
36 9]+\.[0-9]+)? | (\.[0-9]+)) ((e|E) (\-|\+)?[0-9]+)? ( ? , ?) (\-|\+)?([0-
37 9]+\.[0-9]+)? | (\.[0-9]+)) ((e|E) (\-|\+)?[0-9]+)?)) ( (\-|\+)?([0-9]+\.[0-
38 9]+)? | (\.[0-9]+)) ((e|E) (\-|\+)?[0-9]+)?) [0-1] [0-1] (\-|\+)?([0-9]+\.[0-
39 9]+)? | (\.[0-9]+)) ((e|E) (\-|\+)?[0-9]+)? ( ? , ?) (\-|\+)?([0-9]+\.[0-
40 9]+)? | (\.[0-9]+)) ((e|E) (\-|\+)?[0-9]+)?)) ( (\-|\+)?([0-9]+\.[0-9]+)? | (\.[0-
41 9]+)? | (\.[0-9]+)) ((e|E) (\-|\+)?[0-9]+)? ( ? , ?) (\-|\+)?([0-9]+\.[0-
42 9]+)? | (\.[0-9]+)) ((e|E) (\-|\+)?[0-9]+)?)) ( (\-|\+)?([0-9]+\.[0-9]+)? | (\.[0-
43 9]+)) ((e|E) (\-|\+)?[0-9]+)?) [0-1] [0-1] (\-|\+)?([0-9]+\.[0-9]+)? | (\.[0-
44 9]+)) ((e|E) (\-|\+)?[0-9]+)? ( ? , ?) (\-|\+)?([0-9]+\.[0-9]+)? | (\.[0-
45 9]+)) ((e|E) (\-|\+)?[0-9]+)?))* | ( ?(Z|z))*/>
46 </xs:restriction>
47 </xs:simpleType>
48
49 <!-- Abbreviated Geometry grammar for PatGeometry.Figures -->
50 <xs:simpleType name="ST_AbbrGeom">
51 <xs:restriction base="xs:string">
52 <xs:whiteSpace value="collapse" />
53 <!--
54 <xs:pattern value="( ?(M|m) ( ?[rn][scs][rn])) \
55 \
56 ( ?(M|m) ( ?[rn][scs][rn])) | \
57 ( ?(L|l) ( ?[rn][scs][rn]) (
58 [rn][scs][rn])*) | \
59 ( ?(H|h|V|v) ( ?[rn]) ( [rn])*) | \
60 ( ?(Q|q|S|s) ( ?[rn][scs][rn]
61 [rn][scs][rn]) ( ( [rn][scs][rn]){2})*) | \
62 ( ?(C|c) ( ?[rn][scs][rn] (
63 [rn][scs][rn]){2}) ( ( [rn][scs][rn]){3})*) | \

```



```

1
2     <!-- Resource reference -->
3     <xs:simpleType name="ST_RscRef">
4         <xs:restriction base="xs:string">
5             <xs:pattern value="\{StaticResource[\s]+[\S]+\}[\s]*" />
6         </xs:restriction>
7     </xs:simpleType>
8
9     <!-- Resource reference OR Color -->
10    <xs:simpleType name="ST_RscRefColor">
11        <xs:union memberTypes="ST_Color ST_RscRef" />
12    </xs:simpleType>
13
14    <!-- Resource reference OR Compact Matrix-->
15    <xs:simpleType name="ST_RscRefMatrix">
16        <xs:union memberTypes="ST_Matrix ST_RscRef" />
17    </xs:simpleType>
18
19    <!-- Resource reference OR AbbrGeomF-->
20    <xs:simpleType name="ST_RscRefAbbrGeomF">
21        <xs:union memberTypes="ST_AbbrGeomF ST_RscRef" />
22    </xs:simpleType>
23
24    <!-- Sweep Direction enumeration -->
25    <xs:simpleType name="ST_SweepDirection">
26        <xs:restriction base="xs:string">
27            <xs:enumeration value="Clockwise" />
28            <xs:enumeration value="Counterclockwise" />
29        </xs:restriction>
30    </xs:simpleType>
31
32    <!-- Dash Cap enumeration -->
33    <xs:simpleType name="ST_DashCap">
34        <xs:restriction base="xs:string">
35            <xs:enumeration value="Flat" />
36            <xs:enumeration value="Round" />
37            <xs:enumeration value="Square" />
38            <xs:enumeration value="Triangle" />
39        </xs:restriction>
40    </xs:simpleType>
41
42    <!-- Line Cap enumeration -->
43    <xs:simpleType name="ST_LineCap">
44        <xs:restriction base="xs:string">
45            <xs:enumeration value="Flat" />
46            <xs:enumeration value="Round" />
47            <xs:enumeration value="Square" />
48            <xs:enumeration value="Triangle" />
49        </xs:restriction>
50    </xs:simpleType>
51
52    <!-- Line Join enumeration -->
53    <xs:simpleType name="ST_LineJoin">
54        <xs:restriction base="xs:string">
55            <xs:enumeration value="Miter" />
56            <xs:enumeration value="Bevel" />
57            <xs:enumeration value="Round" />
58        </xs:restriction>
59    </xs:simpleType>
60
61    <!-- Tile Mode enumeration -->
62    <xs:simpleType name="ST_TileMode">
63        <xs:restriction base="xs:string">

```

## B XPS Document Schema

```
1         <xs:enumeration value="None" />
2         <xs:enumeration value="Tile" />
3         <xs:enumeration value="FlipX" />
4         <xs:enumeration value="FlipY" />
5         <xs:enumeration value="FlipXY" />
6     </xs:restriction>
7 </xs:simpleType>
8
9 <!-- Color Interpolation Mode enumeration -->
10 <xs:simpleType name="ST_ClrIntMode">
11     <xs:restriction base="xs:string">
12         <xs:enumeration value="ScRgbLinearInterpolation" />
13         <xs:enumeration value="SRgbLinearInterpolation" />
14     </xs:restriction>
15 </xs:simpleType>
16
17 <!-- SpreadMethod Mode enumeration -->
18 <xs:simpleType name="ST_SpreadMethod">
19     <xs:restriction base="xs:string">
20         <xs:enumeration value="Pad" />
21         <xs:enumeration value="Reflect" />
22         <xs:enumeration value="Repeat" />
23     </xs:restriction>
24 </xs:simpleType>
25
26 <!-- FillRule Mode enumeration -->
27 <xs:simpleType name="ST_FillRule">
28     <xs:restriction base="xs:string">
29         <xs:enumeration value="EvenOdd" />
30         <xs:enumeration value="NonZero" />
31     </xs:restriction>
32 </xs:simpleType>
33
34 <!-- Edge Mode enumeration -->
35 <xs:simpleType name="ST_EdgeMode">
36     <xs:restriction base="xs:string">
37         <xs:enumeration value="Aliased" />
38     </xs:restriction>
39 </xs:simpleType>
40
41 <!-- Style Simulation Enumeration -->
42 <xs:simpleType name="ST_StyleSimulations">
43     <xs:restriction base="xs:string">
44         <xs:enumeration value="None" />
45         <xs:enumeration value="ItalicSimulation" />
46         <xs:enumeration value="BoldSimulation" />
47         <xs:enumeration value="BoldItalicSimulation" />
48     </xs:restriction>
49 </xs:simpleType>
50
51 <!-- ViewUnits Enumeration -->
52 <xs:simpleType name="ST_ViewUnits">
53     <xs:restriction base="xs:string">
54         <xs:enumeration value="Absolute" />
55     </xs:restriction>
56 </xs:simpleType>
57
58 <!-- MappingMode Enumeration -->
59 <xs:simpleType name="ST_MappingMode">
60     <xs:restriction base="xs:string">
61         <xs:enumeration value="Absolute" />
62     </xs:restriction>
```

```
1     </xs:simpleType>
2 </xs:schema>
```





## 1 C. Resource Dictionary Key Schema

```
2 <?xml version="1.0" encoding="utf-8"?>
3 <xs:schema
4 targetNamespace="http://schemas.microsoft.com/xps/2005/06/resourcedictionary-
5 key" xmlns="http://schemas.microsoft.com/xps/2005/06/resourcedictionary-key"
6 xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
7 blockDefault="#all">
8
9     <xs:attribute name="Key">
10         <xs:simpleType>
11             <xs:restriction base="xs:string">
12                 <!-- A Key (pattern restriction according to XPS spec) -->
13                 <xs:pattern
14 value="(\p{Lu}|\p{Ll}|\p{Lt}|\p{Lo}|\p{Nl}|_)(\p{Lu}|\p{Ll}|\p{Lt}|\p{Lo}|\p{Nl}
15 )|\p{Mn}|\p{Mc}|\p{Nd}|_)*" />
16                 </xs:restriction>
17             </xs:simpleType>
18         </xs:attribute>
19
20 </xs:schema>
```

21



## 1 D. Document Structure Schema

```

2 <?xml version="1.0" encoding="UTF-8"?>
3 <xs:schema
4 targetNamespace="http://schemas.microsoft.com/xps/2005/06/documentstructure"
5 xmlns="http://schemas.microsoft.com/xps/2005/06/documentstructure"
6 xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
7 blockDefault="#all">
8
9     <xs:import namespace="http://www.w3.org/XML/1998/namespace" />
10
11     <!-- =====DocumentStructure Part===== -->
12     <!-- Complex Types -->
13     <xs:complexType name="CT_DocumentStructure">
14         <xs:sequence>
15             <xs:element ref="DocumentStructure.Outline" minOccurs="0"
16 />
17             <xs:element ref="Story" minOccurs="0" maxOccurs="unbounded"
18 />
19         </xs:sequence>
20     </xs:complexType>
21     <xs:complexType name="CT_CP_Outline">
22         <xs:sequence>
23             <xs:element ref="DocumentOutline" />
24         </xs:sequence>
25     </xs:complexType>
26     <xs:complexType name="CT_DocumentOutline">
27         <xs:sequence>
28             <xs:element ref="OutlineEntry" maxOccurs="unbounded" />
29         </xs:sequence>
30         <xs:attributeGroup ref="AG_DocumentOutline" />
31     </xs:complexType>
32     <xs:complexType name="CT_OutlineEntry">
33         <xs:attributeGroup ref="AG_OutlineEntry" />
34     </xs:complexType>
35     <xs:complexType name="CT_Story">
36         <xs:sequence>
37             <xs:element ref="StoryFragmentReference"
38 maxOccurs="unbounded" />
39         </xs:sequence>
40         <xs:attributeGroup ref="AG_Story" />
41     </xs:complexType>
42     <xs:complexType name="CT_StoryFragmentReference">
43         <xs:attributeGroup ref="AG_StoryFragmentReference" />
44     </xs:complexType>
45     <!-- Simple Types -->
46     <!-- A Name (ID with pattern restriction according to XPS spec) -->
47     <xs:simpleType name="ST_Name">
48         <xs:restriction base="xs:string">
49             <xs:pattern
50 value="(\p{Lu}|\p{Ll}|\p{Lo}|\p{Lt}|\p{Nl}) (\p{Lu}|\p{Ll}|\p{Lo}|\p{Lt}|\p{Nl}|\p{Mn}|\p{Mc}|\p{Nd}|\p{Lm}|_)*" />
51         </xs:restriction>
52     </xs:simpleType>
53     <!-- A Unique Name (ID with pattern restriction according to XPS spec) -->
54     <xs:simpleType name="ST_NameUnique">
55         <xs:restriction base="xs:ID">
56
57

```

## D Document Structure Schema

```
1         <xs:pattern
2 value="(\{Lu\}|\{Ll\}|\{Lo\}|\{Lt\}|\{Nl\}) (\{Lu\}|\{Ll\}|\{Lo\}|\{Lt\}|\{Nl\}|
3 \{Mn\}|\{Mc\}|\{Nd\}|\{Lm\}|_)*" />
4     </xs:restriction>
5 </xs:simpleType>
6 <!-- integer greater than or equal to 1 inclusive -->
7 <xs:simpleType name="ST_IntGEOne">
8     <xs:restriction base="xs:int">
9         <xs:minInclusive value="1" />
10    </xs:restriction>
11 </xs:simpleType>
12 <!-- Elements -->
13 <xs:element name="DocumentStructure" type="CT_DocumentStructure">
14     </xs:element>
15 <xs:element name="DocumentStructure.Outline" type="CT_CP_Outline">
16     </xs:element>
17 <xs:element name="DocumentOutline" type="CT_DocumentOutline">
18     </xs:element>
19 <xs:element name="OutlineEntry" type="CT_OutlineEntry">
20     </xs:element>
21 <xs:element name="Story" type="CT_Story">
22     </xs:element>
23 <xs:element name="StoryFragmentReference"
24 type="CT_StoryFragmentReference">
25     </xs:element>
26 <!-- Attribute Groups -->
27 <xs:attributeGroup name="AG_DocumentOutline">
28     <xs:attribute ref="xml:lang" use="required">
29         </xs:attribute>
30 </xs:attributeGroup>
31 <xs:attributeGroup name="AG_OutlineEntry">
32     <xs:attribute name="OutlineLevel" type="ST_IntGEOne"
33 use="optional" default="1">
34         </xs:attribute>
35     <xs:attribute name="OutlineTarget" type="xs:anyURI"
36 use="required">
37         </xs:attribute>
38     <xs:attribute name="Description" type="xs:string" use="required">
39         </xs:attribute>
40     <xs:attribute ref="xml:lang" use="optional">
41         </xs:attribute>
42 </xs:attributeGroup>
43 <xs:attributeGroup name="AG_Story">
44     <xs:attribute name="StoryName" type="xs:string" use="required">
45         </xs:attribute>
46 </xs:attributeGroup>
47 <xs:attributeGroup name="AG_StoryFragmentReference">
48     <xs:attribute name="FragmentName" type="xs:string" use="optional">
49         </xs:attribute>
50     <xs:attribute name="Page" type="ST_IntGEOne" use="required">
51         </xs:attribute>
52 </xs:attributeGroup>
53 <!-- =====StoryFragments Part===== -->
54 <!-- Complex Types -->
55 <xs:complexType name="CT_StoryFragments">
56     <xs:sequence>
57         <xs:element ref="StoryFragment" maxOccurs="unbounded" />
58     </xs:sequence>
59 </xs:complexType>
60 <xs:complexType name="CT_StoryFragment">
61     <xs:sequence>
62         <xs:element ref="StoryBreak" minOccurs="0" />
```

```

1         <xs:choice maxOccurs="unbounded">
2             <xs:element ref="SectionStructure" />
3             <xs:element ref="ParagraphStructure" />
4             <xs:element ref="ListStructure" />
5             <xs:element ref="TableStructure" />
6             <xs:element ref="FigureStructure" />
7         </xs:choice>
8         <xs:element ref="StoryBreak" minOccurs="0" />
9     </xs:sequence>
10    <xs:attributeGroup ref="AG_StoryFragment" />
11</xs:complexType>
12<xs:complexType name="CT_Break">
13</xs:complexType>
14<xs:complexType name="CT_Section">
15    <xs:choice maxOccurs="unbounded">
16        <xs:element ref="ParagraphStructure" />
17        <xs:element ref="ListStructure" />
18        <xs:element ref="TableStructure" />
19        <xs:element ref="FigureStructure" />
20    </xs:choice>
21</xs:complexType>
22<xs:complexType name="CT_Paragraph">
23    <xs:choice minOccurs="0" maxOccurs="unbounded">
24        <xs:element ref="NamedElement" />
25    </xs:choice>
26</xs:complexType>
27<xs:complexType name="CT_Table">
28    <xs:choice maxOccurs="unbounded">
29        <xs:element ref="TableRowGroupStructure" />
30    </xs:choice>
31</xs:complexType>
32<xs:complexType name="CT_TableRowGroup">
33    <xs:choice maxOccurs="unbounded">
34        <xs:element ref="TableRowStructure" />
35    </xs:choice>
36</xs:complexType>
37<xs:complexType name="CT_TableRow">
38    <xs:choice maxOccurs="unbounded">
39        <xs:element ref="TableCellStructure" />
40    </xs:choice>
41</xs:complexType>
42<xs:complexType name="CT_TableCell">
43    <xs:choice minOccurs="0" maxOccurs="unbounded">
44        <xs:element ref="ParagraphStructure" />
45        <xs:element ref="ListStructure" />
46        <xs:element ref="TableStructure" />
47        <xs:element ref="FigureStructure" />
48    </xs:choice>
49    <xs:attributeGroup ref="AG_TableCell" />
50</xs:complexType>
51<xs:complexType name="CT_List">
52    <xs:choice maxOccurs="unbounded">
53        <xs:element ref="ListItemStructure" />
54    </xs:choice>
55</xs:complexType>
56<xs:complexType name="CT_ListItem">
57    <xs:choice minOccurs="0" maxOccurs="unbounded">
58        <xs:element ref="ParagraphStructure" />
59        <xs:element ref="ListStructure" />
60        <xs:element ref="TableStructure" />
61        <xs:element ref="FigureStructure" />
62    </xs:choice>
63    <xs:attributeGroup ref="AG_ListItem" />

```

## D Document Structure Schema

```
1      </xs:complexType>
2      <xs:complexType name="CT_Figure">
3          <xs:choice minOccurs="0" maxOccurs="unbounded">
4              <xs:element ref="NamedElement" />
5          </xs:choice>
6      </xs:complexType>
7      <xs:complexType name="CT_NamedElement">
8          <xs:attributeGroup ref="AG_NamedElement" />
9      </xs:complexType>
10     <!-- Simple Types -->
11     <!-- FragmentType enumeration -->
12     <xs:simpleType name="ST_FragmentType">
13         <xs:restriction base="xs:string">
14             <xs:enumeration value="Content" />
15             <xs:enumeration value="Header" />
16             <xs:enumeration value="Footer" />
17         </xs:restriction>
18     </xs:simpleType>
19     <xs:simpleType name="ST_Location">
20         <xs:restriction base="xs:string">
21             <xs:pattern value="([0-9][0-9]*) (\, [0-9][0-9]*)*" />
22         </xs:restriction>
23     </xs:simpleType>
24     <xs:simpleType name="ST_TableSpan">
25         <xs:restriction base="xs:int">
26             <xs:minInclusive value="1" />
27         </xs:restriction>
28     </xs:simpleType>
29     <xs:simpleType name="ST_ElementIndex">
30         <xs:restriction base="xs:int">
31             <xs:minInclusive value="0" />
32         </xs:restriction>
33     </xs:simpleType>
34     <!-- Elements -->
35     <xs:element name="StoryFragments" type="CT_StoryFragments">
36         </xs:element>
37     <xs:element name="StoryFragment" type="CT_StoryFragment">
38         </xs:element>
39     <xs:element name="StoryBreak" type="CT_Break">
40         </xs:element>
41     <xs:element name="SectionStructure" type="CT_Section">
42         </xs:element>
43     <xs:element name="ParagraphStructure" type="CT_Paragraph">
44         </xs:element>
45     <xs:element name="TableStructure" type="CT_Table">
46         </xs:element>
47     <xs:element name="TableRowGroupStructure" type="CT_TableRowGroup">
48         </xs:element>
49     <xs:element name="TableRowStructure" type="CT_TableRow">
50         </xs:element>
51     <xs:element name="TableCellStructure" type="CT_TableCell">
52         </xs:element>
53     <xs:element name="ListStructure" type="CT_List">
54         </xs:element>
55     <xs:element name="ListItemStructure" type="CT_ListItem">
56         </xs:element>
57     <xs:element name="FigureStructure" type="CT_Figure">
58         </xs:element>
59     <xs:element name="NamedElement" type="CT_NamedElement">
60         </xs:element>
61     <!-- Attribute Groups -->
62     <xs:attributeGroup name="AG_StoryFragment">
```

```
1         <xs:attribute name="StoryName" type="xs:string" use="optional">
2             </xs:attribute>
3         <xs:attribute name="FragmentName" type="xs:string" use="optional">
4             </xs:attribute>
5         <xs:attribute name="FragmentType" type="ST_FragmentType"
6 use="required">
7             </xs:attribute>
8     </xs:attributeGroup>
9     <xs:attributeGroup name="AG_TableCell">
10        <xs:attribute name="RowSpan" type="ST_TableSpan" use="optional"
11 default="1">
12            </xs:attribute>
13        <xs:attribute name="ColumnSpan" type="ST_TableSpan" use="optional"
14 default="1">
15            </xs:attribute>
16    </xs:attributeGroup>
17    <xs:attributeGroup name="AG_ListItem">
18        <xs:attribute name="Marker" type="ST_NameUnique" use="optional">
19            </xs:attribute>
20    </xs:attributeGroup>
21    <xs:attributeGroup name="AG_NamedElement">
22        <xs:attribute name="NameReference" type="ST_Name" use="required">
23            </xs:attribute>
24    </xs:attributeGroup>
25 </xs:schema>
26
27
```





## 1 E. Discard Control Schema

```
2 <?xml version="1.0" encoding="UTF-8"?>
3 <xs:schema targetNamespace="http://schemas.microsoft.com/xps/2005/06/discard-
4 control" xmlns="http://schemas.microsoft.com/xps/2005/06/discard-control"
5 xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
6 blockDefault="#all">
7
8     <xs:complexType name="CT_DiscardControl">
9         <xs:sequence>
10             <xs:element ref="Discard" minOccurs="0"
11 maxOccurs="unbounded" />
12         </xs:sequence>
13     </xs:complexType>
14
15     <xs:complexType name="CT_Discard">
16         <xs:attribute name="SentinelPage" type="xs:anyURI" use="required">
17         </xs:attribute>
18         <xs:attribute name="Target" type="xs:anyURI" use="required">
19         </xs:attribute>
20     </xs:complexType>
21
22     <xs:element name="DiscardControl" type="CT_DiscardControl">
23     </xs:element>
24
25     <xs:element name="Discard" type="CT_Discard">
26     </xs:element>
27
28 </xs:schema>
29
30
```



## 1 F. Abbreviated Geometry Syntax Algorithm

2 A path geometry specified using the abbreviated geometry syntax (see §11.2.3,  
3 "Abbreviated Geometry Syntax," on page 90) is equivalent to a path specified using  
4 a path geometry. The following algorithm describes how the abbreviated path syntax  
5 can be transformed into a path geometry containing path figures that, in turn,  
6 contain various segments.

7 This algorithm assumes that the presented string is well-formed according to the  
8 markup schema. Whitespace skipping is assumed without being explicitly spelled out  
9 in the algorithm.

```
10
11     Let CURRENTPOINT = 0,0
12     Create a new PathGeometry PG
13     PG.FillRule = EvenOdd
14     Let CURRENTPATHFIGURE = undefined
15
16     Read input character CH
17
18     if ( CH == 'F' )
19     {
20         Read input character CH
21         if ( CH == '0' )
22         {
23             PG.FillRule = EvenOdd
24         }
25         else
26         {
27             PG.FillRule = NonZero
28         }
29     }
30     else
31     {
32         GOTO label_first
33     }
34
35     label_repeat:
36     Read input character CH
37
38     label_first:
39
40     if ( CH == 'm' )
41     {
42         Read relative coordinate pair DX,DY
43         Let CURRENTPOINT.X = CURRENTPOINT.X + DX
44         Let CURRENTPOINT.Y = CURRENTPOINT.Y + DY
45         Create a new PathFigure CURRENTPATHFIGURE and add to PG
46         Let attribute CURRENTPATHFIGURE.StartPoint = CURRENTPOINT
47     }
48     else if ( CH == 'M' )
49     {
50         Read coordinate pair X,Y
51         Let CURRENTPOINT.X = X
52         Let CURRENTPOINT.Y = Y
53         Create a new PathFigure CURRENTPATHFIGURE and add to PG
54         Let attribute CURRENTPATHFIGURE.StartPoint = CURRENTPOINT
55     }
56     else if ( CH == 'l' )
57     {
58         Create new PolyLineSegment S
59         Add S to CURRENTPATHFIGURE
60
61     label_1:
```

## F Abbreviated Geometry Syntax Algorithm

```
1      Read relative coordinate pair DX,DY
2      Let CURRENTPOINT.X = CURRENTPOINT.X + DX
3      Let CURRENTPOINT.Y = CURRENTPOINT.Y + DY
4      Add CURRENTPOINT.X,CURRENTPOINT.Y to end of S.Points attribute list
5      if ( next character is not a letter )
6      {
7          GOTO label_1
8      }
9      }
10     else if ( CH == 'L' )
11     {
12         Create new PolyLineSegment S
13         Add S to CURRENTPATHFIGURE
14     label_2:
15         Read coordinate pair X,Y
16         Let CURRENTPOINT.X = X
17         Let CURRENTPOINT.Y = Y
18         Add CURRENTPOINT.X,CURRENTPOINT.Y to end of S.Points attribute list
19         if ( next character is not a letter )
20         {
21             GOTO label_2
22         }
23     }
24     else if ( CH == 'h' )
25     {
26         Create new PolyLineSegment S
27         Add S to CURRENTPATHFIGURE
28     label_3:
29         Read relative coordinate value DX
30         Let CURRENTPOINT.X = CURRENTPOINT.X + DX
31         Add CURRENTPOINT.X,CURRENTPOINT.Y to end of S.Points attribute list
32         if ( next character is not a letter )
33         {
34             GOTO label_3
35         }
36     }
37     else if ( CH == 'H' )
38     {
39         Create new PolyLineSegment S
40         Add S to CURRENTPATHFIGURE
41     label_4:
42         Read coordinate value X
43         Let CURRENTPOINT.X = X
44         Add CURRENTPOINT.X,CURRENTPOINT.Y to end of S.Points attribute list
45         if ( next character is not a letter )
46         {
47             GOTO label_4
48         }
49     }
50     else if ( CH == 'v' )
51     {
52         Create new PolyLineSegment S
53         Add S to CURRENTPATHFIGURE
54     label_5:
55         Read relative coordinate value DY
56         Let CURRENTPOINT.Y = CURRENTPOINT.Y + DY
57         Add CURRENTPOINT.X,CURRENTPOINT.Y to end of S.Points attribute list
58         if ( next character is not a letter )
59         {
60             GOTO label_5
61         }
62     }
63     else if ( CH == 'V' )
64     {
65         Create new PolyLineSegment S
66         Add S to CURRENTPATHFIGURE
67     label_6:
68         Read coordinate value Y
69         Let CURRENTPOINT.Y = Y
70         Add CURRENTPOINT.X,CURRENTPOINT.Y to end of S.Points attribute list
```

```

1         if ( next character is not a letter )
2         {
3             GOTO label_6
4         }
5     }
6     else if ( CH == 'c' )
7     {
8         Create new PolyBezierSegment S
9         Add S to CURRENTPATHFIGURE
10    label_7:
11        Read relative coordinate pair DX,DY
12        Let POINT.X = CURRENTPOINT.X + DX
13        Let POINT.Y = CURRENTPOINT.Y + DY
14        Add POINT.X, POINT.Y to end of S.Points attribute list
15        Read coordinate pair DX,DY
16        Let POINT.X = CURRENTPOINT.X + DX
17        Let POINT.Y = CURRENTPOINT.Y + DY
18        Add POINT.X, POINT.Y to end of S.Points attribute list
19        Read coordinate pair DX,DY
20        Let CURRENTPOINT.X = CURRENTPOINT.X + DX
21        Let CURRENTPOINT.Y = CURRENTPOINT.Y + DY
22        Add CURRENTPOINT.X,CURRENTPOINT.Y to end of S.Points attribute list
23        if ( next character is not a letter )
24        {
25            GOTO label_7
26        }
27    }
28    else if ( CH == 'C' )
29    {
30        Create new PolyBezierSegment S
31        Add S to CURRENTPATHFIGURE
32    label_8:
33        Read coordinate pair X,Y
34        Let POINT.X = X
35        Let POINT.Y = Y
36        Add POINT.X, POINT.Y to end of S.Points attribute list
37        Read coordinate pair X,Y
38        Let POINT.X = X
39        Let POINT.Y = Y
40        Add POINT.X, POINT.Y to end of S.Points attribute list
41        Read coordinate pair X,Y
42        Let CURRENTPOINT.X = X
43        Let CURRENTPOINT.Y = Y
44        Add CURRENTPOINT.X,CURRENTPOINT.Y to end of S.Points attribute list
45        if ( next character is not a letter )
46        {
47            GOTO label_8
48        }
49    }
50    else if ( CH == 'q' )
51    {
52        Create new PolyQuadraticBezierSegment S
53        Add S to CURRENTPATHFIGURE
54    label_9:
55        Read relative coordinate pair DX,DY
56        Let POINT.X = CURRENTPOINT.X + DX
57        Let POINT.Y = CURRENTPOINT.Y + DY
58        Add POINT.X, POINT.Y to end of S.Points attribute list
59        Read relative coordinate pair DX,DY
60        Let CURRENTPOINT.X = CURRENTPOINT.X + DX
61        Let CURRENTPOINT.Y = CURRENTPOINT.Y + DY
62        Add CURRENTPOINT.X,CURRENTPOINT.Y to end of S.Points attribute list
63        if ( next character is not a letter )
64        {
65            GOTO label_9
66        }
67    }
68    else ( if CH == 'Q' )
69    {
70        Create new PolyQuadraticBezierSegment S
71        Add S to CURRENTPATHFIGURE

```

## F Abbreviated Geometry Syntax Algorithm

```

1  label_10:
2      Read coordinate pair X,Y
3      Let POINT.X = X
4      Let POINT.Y = Y
5      Add POINT.X, POINT.Y to end of S.Points attribute list
6      Read coordinate pair X,Y
7      Let CURRENTPOINT.X = X
8      Let CURRENTPOINT.Y = Y
9      Add CURRENTPOINT.X,CURRENTPOINT.Y to end of S.Points attribute list
10     if ( next character is not a letter )
11     {
12         GOTO label_10
13     }
14     }
15     else if ( CH == 's' )
16     {
17         Create new PolyBezierSegment S
18         Add S to CURRENTPATHFIGURE
19     label_11:
20         if ( S.Points is non-empty )
21         {
22             Let LASTCTRLPOINT = Point before last point in S.Points
23             Let POINT.X = 2 * CURRENTPOINT.X - LASTCTRLPOINT.X
24             Let POINT.Y = 2 * CURRENTPOINT.Y - LASTCTRLPOINT.Y
25         }
26         else if ( segment before CURRENTPATHSEGMENT is a PolyBezierSegment )
27         {
28             Let LASTCTRLPOINT = Point before last point in previous
29             PolyBezierSegment
30             Let POINT.X = 2 * CURRENTPOINT.X - LASTCTRLPOINT.X
31             Let POINT.Y = 2 * CURRENTPOINT.Y - LASTCTRLPOINT.Y
32         }
33         else
34         {
35             Let POINT = CURRENTPOINT
36         }
37
38         Add POINT.X, POINT.Y to end of S.Points attribute list
39         Read relative coordinate pair DX,DY
40         Let POINT.X = CURRENTPOINT.X + DX
41         Let POINT.Y = CURRENTPOINT.Y + DY
42         Add POINT.X, POINT.Y to end of S.Points attribute list
43         Read relative coordinate pair DX,DY
44         Let CURRENTPOINT.X = CURRENTPOINT.X + DX
45         Let CURRENTPOINT.Y = CURRENTPOINT.Y + DY
46         Add CURRENTPOINT.X,CURRENTPOINT.Y to end of S.Points attribute list
47         if ( next character is not a letter )
48         {
49             GOTO label_11
50         }
51     }
52     else if ( CH == 'S' )
53     {
54         Create new PolyBezierSegment S
55         Add S to CURRENTPATHFIGURE
56     label_12:
57         if ( S.Points is non-empty )
58         {
59             Let LASTCTRLPOINT = Point before last point in S.Points
60             Let POINT.X = 2 * CURRENTPOINT.X - LASTCTRLPOINT.X
61             Let POINT.Y = 2 * CURRENTPOINT.Y - LASTCTRLPOINT.Y
62         }
63         else if ( segment before CURRENTPATHSEGMENT is a PolyBezierSegment )
64         {
65             Let LASTCTRLPOINT = S.Point before last point in previous
66             PolyBezierSegment
67             Let POINT.X = 2 * CURRENTPOINT.X - LASTCTRLPOINT.X
68             Let POINT.Y = 2 * CURRENTPOINT.Y - LASTCTRLPOINT.Y
69         }
70         else

```

```

1      {
2          Let POINT = CURRENTPOINT
3      }
4
5      Add POINT.X, POINT.Y to end of S.Points attribute list
6      Read coordinate pair X,Y
7      Let POINT.X = X
8      Let POINT.Y = Y
9      Add POINT.X, POINT.Y to end of S.Points attribute list
10     Read coordinate pair X,Y
11     Let CURRENTPOINT.X = X
12     Let CURRENTPOINT.Y = Y
13     Add CURRENTPOINT.X,CURRENTPOINT.Y to end of S.Points attribute list
14     if ( next character is not a letter )
15     {
16         GOTO label_12
17     }
18 }
19 else if ( CH == 'a' )
20 {
21 label_13:
22     Create new ArcSegment S
23     Add S to CURRENTPATHFIGURE
24     Read Radius Pair RX,RY
25     Read Rotation ROT
26     Read integer FLAG1
27     Read integer FLAG2
28     Read relative coordinate pair DX,DY
29     Let CURRENTPOINT.X = CURRENTPOINT.X + DX
30     Let CURRENTPOINT.Y = CURRENTPOINT.Y + DY
31     Let S.Point = CURRENTPOINT.X,CURRENTPOINT.Y
32     Let S.IsLargeArc = (FLAG1 == 1 ? true : false)
33     Let S.SweepDirection = (FLAG2 == 1 ? Clockwise : Counterclockwise)
34     Let S.RotationAngle = ROT
35     Let S.Size = RX, RY
36     if ( next character is not a letter )
37     {
38         GOTO label_13
39     }
40 }
41 else if ( CH == 'A' )
42 {
43 label_14:
44     Create new ArcSegment S
45     Add S to CURRENTPATHFIGURE
46     Read Radius Pair RX,RY
47     Read Rotation ROT
48     Read integer FLAG1
49     Read integer FLAG2
50     Read coordinate pair X,Y
51     Let CURRENTPOINT.X = X
52     Let CURRENTPOINT.Y = Y
53     Let S.Point = CURRENTPOINT.X,CURRENTPOINT.Y
54     Let S.IsLargeArc = (FLAG1 == 1 ? true : false)
55     Let S.SweepDirection = (FLAG2 == 1 ? Clockwise : Counterclockwise)
56     Let S.RotationAngle = ROT
57     Let S.Size = RX, RY
58     if ( next character is not a letter )
59     {
60         GOTO label_14
61     }
62 }
63 else if ( CH == 'z' or CH == 'Z' )
64 {
65     Let attribute CURRENTPATHFIGURE.IsClosed = true
66     Left CURRENTPOINT = First point of first segment of CURRENTPATHFIGURE
67     Let CURRENTPATHFIGURE = undefined
68 }
69 /* This case can not occur, because the input is assumed to be well-formed
70 according to the markup schema
71 else

```

## F Abbreviated Geometry Syntax Algorithm

```
1      {  
2          ERROR: Invalid input character  
3      }  
4      */  
5  
6      if ( End of input reached )  
7      {  
8          Terminate algorithm, return PG  
9      }  
10     else  
11     {  
12         GOTO label_repeat  
13     }  
14
```



## 1 G. scRGB Gamut Boundary Definition

```

2 <?xml version="1.0" encoding="utf-8" ?>
3 <cdm:ColorDeviceModel
4   ID="http://schemas.microsoft.com/windows/2005/02/color/wcsRGB.cdmp"
5   xmlns:cdm="http://schemas.microsoft.com/windows/2005/02/color/
6     ColorDeviceModel"
7   xmlns:wcs="http://schemas.microsoft.com/windows/2005/02/color/
8     WcsCommonProfileTypes"
9   xmlns:xs="http://www.w3.org/2001/XMLSchema-instance">
10
11   <cdm:ProfileName>
12     <wcs:Text xml:lang="en-US">wcsRGB virtual device model profile</wcs:Text>
13   </cdm:ProfileName>
14   <cdm:Description>
15     <wcs:Text xml:lang="en-US">wcsRGB virtual s2.13 device model system
16 profile</wcs:Text>
17   </cdm:Description>
18   <cdm:Author>
19     <wcs:Text xml:lang="en-US">Microsoft Corporation</wcs:Text>
20   </cdm:Author>
21
22   <cdm:MeasurementConditions>
23     <cdm:ColorSpace>CIEXYZ</cdm:ColorSpace>
24     <cdm:WhitePointName>D65</cdm:WhitePointName>
25   </cdm:MeasurementConditions>
26   <cdm:SelfLuminous>true</cdm:SelfLuminous>
27   <cdm:MaxColorant>4.0</cdm:MaxColorant>
28   <cdm:MinColorant>-4.0</cdm:MinColorant>
29
30   <cdm:RGBVirtualDevice>
31     <cdm:MeasurementData TimeStamp="2005-02-09T22:00:00">
32       <cdm:WhitePrimary X="95.05" Y="100.00" Z="108.90" />
33       <cdm:RedPrimary X="41.24" Y="21.26" Z="1.93" />
34       <cdm:GreenPrimary X="35.76" Y="71.52" Z="11.92" />
35       <cdm:BluePrimary X="18.05" Y="7.22" Z="95.05" />
36       <cdm:BlackPrimary X="0" Y="0" Z="0" />
37       <cdm:Gamma value="1.0" />
38       <cdm:GamutBoundarySamples>
39         <cdm:RGB R="0.166433" G="-0.016114" B="0.027918" />
40         <cdm:RGB R="0.257134" G="-0.026542" B="0.041950" />
41         <cdm:RGB R="0.373396" G="-0.039395" B="0.060310" />
42         <cdm:RGB R="0.516590" G="-0.054207" B="0.083665" />
43         <cdm:RGB R="0.161885" G="-0.013701" B="0.015224" />
44         <cdm:RGB R="0.254873" G="-0.024242" B="0.022314" />
45         <cdm:RGB R="0.370591" G="-0.036207" B="0.031873" />
46         <cdm:RGB R="0.645317" G="-0.053422" B="0.061566" />
47         <cdm:RGB R="0.246052" G="-0.020570" B="0.009804" />
48         <cdm:RGB R="0.366951" G="-0.033595" B="0.013350" />
49         <cdm:RGB R="0.513086" G="-0.047777" B="0.018342" />
50         <cdm:RGB R="0.348159" G="-0.027092" B="0.002697" />
51         <cdm:RGB R="0.501415" G="-0.043030" B="0.003038" />
52         <cdm:RGB R="0.204158" G="-0.007054" B="-0.001921" />
53         <cdm:RGB R="0.102843" G="0.005460" B="-0.002574" />
54         <cdm:RGB R="0.161680" G="0.005828" B="-0.003751" />
55         <cdm:RGB R="0.241218" G="0.005609" B="-0.005263" />
56         <cdm:RGB R="0.345298" G="0.004589" B="-0.007162" />
57         <cdm:RGB R="1.381166" G="0.758736" B="0.088389" />
58         <cdm:RGB R="0.747224" G="0.490147" B="-0.065045" />

```

## G scRGB Gamut Boundary Definition

```
1 <cdm:RGB R="0.892023" G="0.585517" B="-0.076908" />
2 <cdm:RGB R="1.054700" G="0.693668" B="-0.087994" />
3 <cdm:RGB R="1.237403" G="0.815408" B="-0.099715" />
4 <cdm:RGB R="0.429786" G="0.463528" B="-0.056073" />
5 <cdm:RGB R="0.109329" G="0.215829" B="-0.023359" />
6 <cdm:RGB R="0.170974" G="0.342970" B="-0.037798" />
7 <cdm:RGB R="0.032781" G="0.182702" B="-0.017604" />
8 <cdm:RGB R="0.040703" G="0.236387" B="-0.023182" />
9 <cdm:RGB R="0.052010" G="0.299807" B="-0.029289" />
10 <cdm:RGB R="-0.015232" G="0.150695" B="-0.012907" />
11 <cdm:RGB R="-0.020689" G="0.198705" B="-0.017162" />
12 <cdm:RGB R="-0.086126" G="0.133080" B="0.007658" />
13 <cdm:RGB R="-0.178409" G="0.235796" B="0.126003" />
14 <cdm:RGB R="0.000000" G="0.000000" B="0.000000" />
15 <cdm:RGB R="-0.150496" G="0.178902" B="0.156523" />
16 <cdm:RGB R="-0.193313" G="0.234293" B="0.204426" />
17 <cdm:RGB R="-0.202798" G="0.230449" B="0.290886" />
18 <cdm:RGB R="-0.211558" G="0.225780" B="0.384316" />
19 <cdm:RGB R="-0.249008" G="0.286162" B="0.470316" />
20 <cdm:RGB R="1.024267" G="0.995126" B="0.956518" />
21 <cdm:RGB R="-0.211156" G="0.218574" B="0.473918" />
22 <cdm:RGB R="-0.209022" G="0.210461" B="0.567755" />
23 <cdm:RGB R="0.368209" G="0.814614" B="1.020972" />
24 <cdm:RGB R="-0.194449" G="0.199547" B="0.648518" />
25 <cdm:RGB R="0.099507" G="0.602350" B="1.001310" />
26 <cdm:RGB R="0.447496" G="0.790515" B="1.022822" />
27 <cdm:RGB R="-0.123046" G="0.108061" B="0.470343" />
28 <cdm:RGB R="-0.150585" G="0.144010" B="0.596962" />
29 <cdm:RGB R="0.185831" G="0.575754" B="1.007060" />
30 <cdm:RGB R="0.331216" G="0.668936" B="1.023494" />
31 <cdm:RGB R="0.410196" G="0.645807" B="1.014063" />
32 <cdm:RGB R="0.574938" G="0.752116" B="1.021593" />
33 <cdm:RGB R="0.483140" G="0.624458" B="1.005310" />
34 <cdm:RGB R="0.635447" G="0.734762" B="1.010061" />
35 <cdm:RGB R="-0.052205" G="0.035138" B="0.436365" />
36 <cdm:RGB R="1.024267" G="0.995126" B="0.956518" />
37 <cdm:RGB R="0.028844" G="-0.005277" B="0.123163" />
38 <cdm:RGB R="0.049636" G="-0.010904" B="0.228673" />
39 <cdm:RGB R="0.080157" G="-0.012435" B="0.135751" />
40 <cdm:RGB R="0.127487" G="-0.020154" B="0.216960" />
41 <cdm:RGB R="0.190206" G="-0.029844" B="0.323182" />
42 <cdm:RGB R="0.097580" G="-0.013313" B="0.083401" />
43 <cdm:RGB R="0.155931" G="-0.021750" B="0.133422" />
44 <cdm:RGB R="0.230038" G="-0.031267" B="0.196426" />
45 <cdm:RGB R="0.326426" G="-0.043430" B="0.278289" />
46 <cdm:RGB R="0.426090" G="-0.050043" B="0.360668" />
47 <cdm:RGB R="0.098869" G="-0.010753" B="0.048113" />
48 <cdm:RGB R="0.164545" G="-0.019812" B="0.079230" />
49 <cdm:RGB R="0.249557" G="-0.030741" B="0.119834" />
50 <cdm:RGB R="0.357564" G="-0.043900" B="0.171719" />
51 <cdm:RGB R="0.495437" G="-0.060672" B="0.237960" />
52 <cdm:RGB R="0.619457" G="-0.065550" B="0.301872" />
53 <cdm:RGB R="0.000000" G="0.000000" B="0.000000" />
54 <cdm:RGB R="0.169101" G="-0.018572" B="0.047904" />
55 <cdm:RGB R="0.255247" G="-0.028467" B="0.071995" />
56 <cdm:RGB R="0.368932" G="-0.041653" B="0.103698" />
57 <cdm:RGB R="0.508606" G="-0.056762" B="0.143412" />
58 <cdm:RGB R="0.637110" G="-0.061275" B="0.186481" />
59 <cdm:RGB R="1.249149" G="-0.024436" B="-0.002194" />
60 <cdm:RGB R="0.000475" G="0.001355" B="0.043728" />
61 <cdm:RGB R="1.257972" G="-0.026125" B="0.049916" />
62 <cdm:RGB R="1.264195" G="-0.027432" B="0.104713" />
```

```

1      <cdm:RGB R="0.003291" G="-0.009926" B="0.365422" />
2      <cdm:RGB R="1.287764" G="-0.033804" B="0.414704" />
3      <cdm:RGB R="0.004536" G="-0.013745" B="0.512430" />
4      <cdm:RGB R="1.297183" G="-0.037024" B="0.573653" />
5      <cdm:RGB R="1.307374" G="-0.040929" B="0.763754" />
6      <cdm:RGB R="0.007818" G="-0.023805" B="0.902837" />
7      <cdm:RGB R="-0.001325" G="0.002612" B="-0.000199" />
8      <cdm:RGB R="-0.007337" G="0.015263" B="-0.001058" />
9      <cdm:RGB R="-0.005430" G="-0.000728" B="0.906022" />
10     <cdm:RGB R="-0.020327" G="0.043021" B="-0.002887" />
11     <cdm:RGB R="-0.042091" G="0.089818" B="-0.005932" />
12     <cdm:RGB R="-0.117963" G="0.253649" B="-0.016500" />
13     <cdm:RGB R="-0.141327" G="0.262090" B="0.943868" />
14     <cdm:RGB R="-0.174730" G="0.376455" B="-0.024386" />
15     <cdm:RGB R="-0.204641" G="0.390245" B="0.954021" />
16     <cdm:RGB R="-0.245665" G="0.530007" B="-0.034229" />
17     <cdm:RGB R="-0.282073" G="0.548898" B="0.962351" />
18     <cdm:RGB R="1.150086" G="0.568927" B="1.001779" />
19     <cdm:RGB R="-0.374737" G="0.740509" B="0.968358" />
20     <cdm:RGB R="1.083883" G="0.766726" B="1.001175" />
21     <cdm:RGB R="1.002593" G="0.985026" B="-0.073045" />
22     <cdm:RGB R="-0.434478" G="0.938894" B="-0.060390" />
23     <cdm:RGB R="0.000055" G="-0.000136" B="0.003135" />
24     <cdm:RGB R="0.000203" G="-0.000555" B="0.016188" />
25     <cdm:RGB R="0.000896" G="-0.002619" B="0.089419" />
26     <cdm:RGB R="0.001492" G="-0.004425" B="0.156418" />
27     <cdm:RGB R="0.002283" G="-0.006839" B="0.247555" />
28     <cdm:RGB R="0.006038" G="-0.018353" B="0.690852" />
29     <cdm:RGB R="0.007818" G="-0.023805" B="0.902837" />
30     <cdm:RGB R="-0.483692" G="0.967403" B="0.971726" />
31     <cdm:RGB R="0.003728" G="-0.000194" B="0.002807" />
32     <cdm:RGB R="0.004140" G="-0.000555" B="0.015782" />
33     <cdm:RGB R="0.004738" G="-0.001328" B="0.043372" />
34     <cdm:RGB R="0.005489" G="-0.002582" B="0.089155" />
35     <cdm:RGB R="0.006397" G="-0.004383" B="0.156254" />
36     <cdm:RGB R="0.007479" G="-0.006795" B="0.247483" />
37     <cdm:RGB R="0.008758" G="-0.009881" B="0.365433" />
38     <cdm:RGB R="0.010255" G="-0.013700" B="0.512516" />
39     <cdm:RGB R="0.011993" G="-0.018308" B="0.691003" />
40     <cdm:RGB R="0.013995" G="-0.023760" B="0.903048" />
41     <cdm:RGB R="0.034050" G="-0.018254" B="0.692249" />
42     <cdm:RGB R="0.036746" G="-0.023689" B="0.904397" />
43     <cdm:RGB R="0.080545" G="-0.023830" B="0.908226" />
44     <cdm:RGB R="0.149752" G="-0.024452" B="0.915123" />
45     <cdm:RGB R="1.002593" G="0.985026" B="-0.073045" />
46     <cdm:RGB R="1.249149" G="-0.024436" B="-0.002194" />
47     <cdm:RGB R="1.252780" G="-0.025117" B="0.015342" />
48     <cdm:RGB R="1.257972" G="-0.026125" B="0.049916" />
49     <cdm:RGB R="1.271263" G="-0.029093" B="0.182298" />
50     <cdm:RGB R="1.279122" G="-0.031189" B="0.284927" />
51     <cdm:RGB R="1.002258" G="0.999362" B="0.998757" />
52     <cdm:RGB R="0.007818" G="-0.023805" B="0.902837" />
53     <cdm:RGB R="-0.001325" G="0.002612" B="-0.000199" />
54     <cdm:RGB R="0.004903" G="-0.018821" B="0.903364" />
55     <cdm:RGB R="-0.005430" G="-0.000728" B="0.906022" />
56     <cdm:RGB R="-0.024203" G="0.033431" B="0.912191" />
57     <cdm:RGB R="-0.042091" G="0.089818" B="-0.005932" />
58     <cdm:RGB R="-0.052335" G="0.086586" B="0.921578" />
59     <cdm:RGB R="-0.074175" G="0.159024" B="-0.010406" />
60     <cdm:RGB R="-0.090966" G="0.161809" B="0.932665" />
61     <cdm:RGB R="-0.117963" G="0.253649" B="-0.016500" />
62     <cdm:RGB R="-0.174730" G="0.376455" B="-0.024386" />
63     <cdm:RGB R="-0.204641" G="0.390245" B="0.954021" />

```

## G scRGB Gamut Boundary Definition

```
1 <cdm:RGB R="-0.331890" G="0.716723" B="-0.046182" />
2 <cdm:RGB R="-0.374737" G="0.740509" B="0.968358" />
3 <cdm:RGB R="-0.434478" G="0.938894" B="-0.060390" />
4 <cdm:RGB R="0.003567" G="-0.000129" B="-0.000031" />
5 <cdm:RGB R="0.013995" G="-0.023760" B="0.903048" />
6 <cdm:RGB R="0.011145" G="-0.018782" B="0.903675" />
7 <cdm:RGB R="0.001010" G="-0.000696" B="0.906548" />
8 <cdm:RGB R="-0.017488" G="0.033476" B="0.912853" />
9 <cdm:RGB R="-0.045318" G="0.086662" B="0.922243" />
10 <cdm:RGB R="-0.083641" G="0.161924" B="0.933260" />
11 <cdm:RGB R="-0.133701" G="0.262246" B="0.944373" />
12 <cdm:RGB R="-0.196722" G="0.390440" B="0.954439" />
13 <cdm:RGB R="0.020512" G="-0.000542" B="-0.000142" />
14 <cdm:RGB R="0.034096" G="-0.018719" B="0.905266" />
15 <cdm:RGB R="0.024587" G="-0.000632" B="0.908671" />
16 <cdm:RGB R="0.006983" G="0.033602" B="0.915280" />
17 <cdm:RGB R="-0.019836" G="0.086907" B="0.924618" />
18 <cdm:RGB R="-0.057114" G="0.162314" B="0.935374" />
19 <cdm:RGB R="-0.106133" G="0.262787" B="0.946166" />
20 <cdm:RGB R="-0.168134" G="0.391122" B="0.955918" />
21 <cdm:RGB R="0.080545" G="-0.023830" B="0.908226" />
22 <cdm:RGB R="0.078174" G="-0.018848" B="0.909282" />
23 <cdm:RGB R="0.069557" G="-0.000692" B="0.913080" />
24 <cdm:RGB R="0.053304" G="0.033705" B="0.919825" />
25 <cdm:RGB R="0.028089" G="0.087255" B="0.928902" />
26 <cdm:RGB R="-0.007469" G="0.162946" B="0.939149" />
27 <cdm:RGB R="-0.054727" G="0.263702" B="0.949360" />
28 <cdm:RGB R="0.149752" G="-0.024452" B="0.915123" />
29 <cdm:RGB R="0.147668" G="-0.019423" B="0.916222" />
30 <cdm:RGB R="0.140009" G="-0.001093" B="0.920043" />
31 <cdm:RGB R="0.125292" G="0.033611" B="0.926547" />
32 <cdm:RGB R="0.102016" G="0.087566" B="0.935050" />
33 <cdm:RGB R="0.068642" G="0.163702" B="0.944526" />
34 <cdm:RGB R="0.023701" G="0.264902" B="0.953919" />
35 <cdm:RGB R="0.248626" G="-0.025756" B="0.924685" />
36 <cdm:RGB R="0.246801" G="-0.020644" B="0.925712" />
37 <cdm:RGB R="0.240032" G="-0.002034" B="0.929243" />
38 <cdm:RGB R="0.226828" G="0.033128" B="0.935165" />
39 <cdm:RGB R="0.205577" G="0.087658" B="0.942824" />
40 <cdm:RGB R="0.174609" G="0.164417" B="0.951306" />
41 <cdm:RGB R="0.132331" G="0.266234" B="0.959693" />
42 <cdm:RGB R="0.381162" G="-0.027864" B="0.936087" />
43 <cdm:RGB R="0.379554" G="-0.022644" B="0.936982" />
44 <cdm:RGB R="0.313886" G="0.164903" B="0.959216" />
45 <cdm:RGB R="0.274412" G="0.267520" B="0.966469" />
46 <cdm:RGB R="0.551062" G="-0.030854" B="0.948537" />
47 <cdm:RGB R="0.404402" G="0.399877" B="0.979322" />
48 <cdm:RGB R="0.761768" G="-0.034778" B="0.961424" />
49 <cdm:RGB R="1.016507" G="-0.039680" B="0.974324" />
50 <cdm:RGB R="1.247925" G="-0.024207" B="-0.006804" />
51 <cdm:RGB R="1.318331" G="-0.045598" B="0.986957" />
52 <cdm:RGB R="1.002593" G="0.985026" B="-0.073045" />
53 <cdm:RGB R="-0.017084" G="0.004071" B="0.105192" />
54 <cdm:RGB R="0.000276" G="-0.000778" B="0.024704" />
55 <cdm:RGB R="0.001189" G="-0.003497" B="0.124447" />
56 <cdm:RGB R="0.003835" G="-0.011408" B="0.425034" />
57 <cdm:RGB R="0.006525" G="-0.019382" B="0.730759" />
58 <cdm:RGB R="0.991838" G="0.854433" B="0.996305" />
59 <cdm:RGB R="-0.085296" G="0.980282" B="0.983420" />
60 <cdm:RGB R="0.997109" G="1.001108" B="1.001729" />
61 <cdm:RGB R="0.000038" G="-0.000093" B="0.002143" />
62 <cdm:RGB R="0.000087" G="-0.000230" B="0.006229" />
```

```

1      <cdm:RGB R="0.000627" G="-0.001818" B="0.062146" />
2      <cdm:RGB R="0.005522" G="-0.016416" B="0.616931" />
3      <cdm:RGB R="0.006525" G="-0.019382" B="0.730759" />
4      <cdm:RGB R="0.001838" G="-0.000115" B="0.001922" />
5      <cdm:RGB R="0.001968" G="-0.000231" B="0.005969" />
6      <cdm:RGB R="0.002147" G="-0.000441" B="0.013224" />
7      <cdm:RGB R="0.002370" G="-0.000761" B="0.024478" />
8      <cdm:RGB R="0.002634" G="-0.001207" B="0.040491" />
9      <cdm:RGB R="0.002944" G="-0.001796" B="0.062002" />
10     <cdm:RGB R="0.003302" G="-0.002545" B="0.089732" />
11     <cdm:RGB R="0.003715" G="-0.003473" B="0.124383" />
12     <cdm:RGB R="0.004188" G="-0.004596" B="0.166643" />
13     <cdm:RGB R="0.004728" G="-0.005932" B="0.217190" />
14     <cdm:RGB R="0.005341" G="-0.007497" B="0.276684" />
15     <cdm:RGB R="0.006813" G="-0.011384" B="0.425118" />
16     <cdm:RGB R="0.008657" G="-0.016392" B="0.617056" />
17     <cdm:RGB R="0.009736" G="-0.019358" B="0.730905" />
18     <cdm:RGB R="-0.080359" G="0.980412" B="0.983525" />
19     <cdm:RGB R="0.005604" G="-0.000209" B="0.001935" />
20     <cdm:RGB R="0.005786" G="-0.000307" B="0.006004" />
21     <cdm:RGB R="0.006041" G="-0.000495" B="0.013270" />
22     <cdm:RGB R="0.006363" G="-0.000795" B="0.024529" />
23     <cdm:RGB R="0.006745" G="-0.001225" B="0.040547" />
24     <cdm:RGB R="0.007184" G="-0.001801" B="0.062069" />
25     <cdm:RGB R="0.007679" G="-0.002542" B="0.089815" />
26     <cdm:RGB R="0.008231" G="-0.003463" B="0.124487" />
27     <cdm:RGB R="0.009523" G="-0.005912" B="0.217344" />
28     <cdm:RGB R="0.010273" G="-0.007474" B="0.276866" />
29     <cdm:RGB R="0.012011" G="-0.011356" B="0.425354" />
30     <cdm:RGB R="0.013011" G="-0.013710" B="0.515596" />
31     <cdm:RGB R="0.014108" G="-0.016361" B="0.617344" />
32     <cdm:RGB R="0.015308" G="-0.019327" B="0.731220" />
33     <cdm:RGB R="0.021793" G="-0.013690" B="0.516185" />
34     <cdm:RGB R="0.023073" G="-0.016337" B="0.617959" />
35     <cdm:RGB R="0.024454" G="-0.019298" B="0.731861" />
36     <cdm:RGB R="0.025943" G="-0.022591" B="0.858499" />
37     <cdm:RGB R="0.037804" G="-0.019298" B="0.732989" />
38     <cdm:RGB R="0.039522" G="-0.022583" B="0.859653" />
39     <cdm:RGB R="0.057988" G="-0.022628" B="0.861439" />
40     <cdm:RGB R="-0.012314" G="0.982157" B="0.984935" />
41     <cdm:RGB R="0.020932" G="0.982985" B="0.985609" />
42     <cdm:RGB R="0.110705" G="0.985131" B="0.987376" />
43     <cdm:RGB R="0.234831" G="0.987909" B="0.989706" />
44     <cdm:RGB R="0.311005" G="0.989512" B="0.991076" />
45     <cdm:RGB R="0.494043" G="0.993078" B="0.994197" />
46     <cdm:RGB R="0.721366" G="0.997001" B="0.997773" />
47     <cdm:RGB R="0.852925" G="0.999044" B="0.999710" />
48     <cdm:RGB R="0.997109" G="1.001108" B="1.001729" />
49     <cdm:RGB R="0.007645" G="-0.022679" B="0.857328" />
50     <cdm:RGB R="0.001742" G="-0.000069" B="-0.000016" />
51     <cdm:RGB R="0.010928" G="-0.022656" B="0.857495" />
52     <cdm:RGB R="0.005453" G="-0.000169" B="-0.000042" />
53     <cdm:RGB R="0.016619" G="-0.022623" B="0.857831" />
54     <cdm:RGB R="-0.071870" G="0.980634" B="0.983705" />
55     <cdm:RGB R="0.012251" G="-0.000330" B="-0.000086" />
56     <cdm:RGB R="-0.058141" G="0.980991" B="0.983991" />
57     <cdm:RGB R="-0.038488" G="0.981494" B="0.984397" />
58     <cdm:RGB R="0.038332" G="-0.000891" B="-0.000240" />
59     <cdm:RGB R="0.081991" G="-0.022759" B="0.863942" />
60     <cdm:RGB R="0.112193" G="-0.023008" B="0.867198" />
61     <cdm:RGB R="0.061770" G="0.983975" B="0.986421" />
62     <cdm:RGB R="0.168229" G="0.986444" B="0.988470" />
63     <cdm:RGB R="0.397243" G="0.991242" B="0.992576" />

```

## G scRGB Gamut Boundary Definition

```
1         <cdm:RGB R="0.601915" G="0.995006" B="0.995934" />
2         <cdm:RGB R="0.721366" G="0.997001" B="0.997773" />
3         <cdm:RGB R="0.849328" G="0.852650" B="0.994033" />
4         <cdm:RGB R="0.000000" G="0.000000" B="1.000000" />
5     </cdm:GamutBoundarySamples>
6 </cdm:MeasurementData>
7
8 </cdm:RGBVirtualDevice>
9
10 </cdm:ColorDeviceModel>
11
```

1

## 2 H. Standard Namespaces and Content Types

3 The following tables list the namespaces and content types used in XPS packages  
4 and XPS Documents.

### 5 H.1 XML Namespace URIs

6 *Table H-1. Package-wide namespaces*

<b>Description</b>	<b>Namespace URI</b>
Content Types	<a href="http://schemas.openxmlformats.org/package/2006/content-types">http://schemas.openxmlformats.org/package/2006/content-types</a>
Core Properties	<a href="http://schemas.openxmlformats.org/package/2006/metadata/core-properties">http://schemas.openxmlformats.org/package/2006/metadata/core-properties</a>
Digital Signatures	<a href="http://schemas.openxmlformats.org/package/2006/digital-signature">http://schemas.openxmlformats.org/package/2006/digital-signature</a>
Relationships	<a href="http://schemas.openxmlformats.org/package/2006/relationships">http://schemas.openxmlformats.org/package/2006/relationships</a>
Markup Compatibility	<a href="http://schemas.openxmlformats.org/markup-compatibility/2006">http://schemas.openxmlformats.org/markup-compatibility/2006</a>

7 *Table H-2. XPS Document namespaces*

<b>Description</b>	<b>Namespace URI</b>
DiscardControl	<a href="http://schemas.microsoft.com/xps/2005/06/discard-control">http://schemas.microsoft.com/xps/2005/06/discard-control</a>
Document Structure	<a href="http://schemas.microsoft.com/xps/2005/06/documentstructure">http://schemas.microsoft.com/xps/2005/06/documentstructure</a>
FixedDocument	<a href="http://schemas.microsoft.com/xps/2005/06">http://schemas.microsoft.com/xps/2005/06</a>
FixedDocumentSequence	<a href="http://schemas.microsoft.com/xps/2005/06">http://schemas.microsoft.com/xps/2005/06</a>
FixedPage	<a href="http://schemas.microsoft.com/xps/2005/06">http://schemas.microsoft.com/xps/2005/06</a>
Print Schema Framework	<a href="http://schemas.microsoft.com/windows/2003/08/printing/printschemaframework">http://schemas.microsoft.com/windows/2003/08/printing/printschemaframework</a>
Print Schema Keywords	<a href="http://schemas.microsoft.com/windows/2003/08/printing/printschemakeywords">http://schemas.microsoft.com/windows/2003/08/printing/printschemakeywords</a>
Resource Dictionary (Key attribute)	<a href="http://schemas.microsoft.com/xps/2005/06/resource-dictionary-key">http://schemas.microsoft.com/xps/2005/06/resource-dictionary-key</a>
Signature Definitions	<a href="http://schemas.microsoft.com/xps/2005/06/signature-definitions">http://schemas.microsoft.com/xps/2005/06/signature-definitions</a>
Story Fragments	<a href="http://schemas.microsoft.com/xps/2005/06/documentstructure">http://schemas.microsoft.com/xps/2005/06/documentstructure</a>

---

## 1 H.2 Content Types

2 The content types in the tables below MUST NOT include parameters. A consumer  
 3 MUST treat the presence of parameters on these content types as an error when the  
 4 affected part is accessed [M12.7].

5 *Table H-3. Package-wide content types*

<b>Description</b>	<b>Content type</b>
Core Properties part	application/vnd.openxmlformats-package.core-properties+xml
Digital Signature Certificate part	application/vnd.openxmlformats-package.digital-signature-certificate
Digital Signature Origin part	application/vnd.openxmlformats-package.digital-signature-origin
Digital Signature XML Signature part	application/vnd.openxmlformats-package.digital-signature-xmlsignature+xml
Relationships part	application/vnd.openxmlformats-package.relationships+xml

6 *Table H-4. XPS Document content types*

<b>Description</b>	<b>Content type</b>
FixedDocument	application/vnd.ms-package.xps-fixeddocument+xml
FixedDocumentSequence	application/vnd.ms-package.xps-fixeddocumentsequence+xml
FixedPage	application/vnd.ms-package.xps-fixedpage+xml
DiscardControl	application/vnd.ms-package.xps-discard-control+xml
DocumentStructure	application/vnd.ms-package.xps-documentstructure+xml
Font	application/vnd.ms-opentype
ICC profile	application/vnd.ms-color.iccprofile
JPEG image	image/jpeg
Obfuscated font	application/vnd.ms-package.obfuscated-opentype
PNG image	image/png
PrintTicket	application/vnd.ms-printing.printticket+xml
Remote resource dictionary	application/vnd.ms-package.xps-resourcedictionary+xml
StoryFragments	application/vnd.ms-package.xps-storyfragments+xml
TIFF image	image/tiff
Thumbnail part	image/jpeg or image/png
Windows Media Photo image	image/vnd.ms-photo



---

### 1 H.3 Relationship Types

2 *Table H-5. Package-wide relationship types*

<b>Description</b>	<b>Relationship type</b>
Core Properties	<a href="http://schemas.openxmlformats.org/package/2006/relationships/metadata/core-properties">http://schemas.openxmlformats.org/package/2006/relationships/metadata/core-properties</a>
Digital Signature	<a href="http://schemas.openxmlformats.org/package/2006/relationships/digital-signature/signature">http://schemas.openxmlformats.org/package/2006/relationships/digital-signature/signature</a>
Digital Signature Certificate	<a href="http://schemas.openxmlformats.org/package/2006/relationships/digital-signature/certificate">http://schemas.openxmlformats.org/package/2006/relationships/digital-signature/certificate</a>
Digital Signature Origin	<a href="http://schemas.openxmlformats.org/package/2006/relationships/digital-signature/origin">http://schemas.openxmlformats.org/package/2006/relationships/digital-signature/origin</a>
Thumbnail	<a href="http://schemas.openxmlformats.org/package/2006/relationships/metadata/thumbnail">http://schemas.openxmlformats.org/package/2006/relationships/metadata/thumbnail</a>

3 *Table H-6. XPS Document relationship types*

<b>Description</b>	<b>Relationship type</b>
Digital Signature Definitions	<a href="http://schemas.microsoft.com/xps/2005/06/signature-definitions">http://schemas.microsoft.com/xps/2005/06/signature-definitions</a>
DiscardControl	<a href="http://schemas.microsoft.com/xps/2005/06/discard-control">http://schemas.microsoft.com/xps/2005/06/discard-control</a>
DocumentStructure	<a href="http://schemas.microsoft.com/xps/2005/06/documentstructure">http://schemas.microsoft.com/xps/2005/06/documentstructure</a>
PrintTicket	<a href="http://schemas.microsoft.com/xps/2005/06/printticket">http://schemas.microsoft.com/xps/2005/06/printticket</a>
Required Resource	<a href="http://schemas.microsoft.com/xps/2005/06/required-resource">http://schemas.microsoft.com/xps/2005/06/required-resource</a>
Restricted Font	<a href="http://schemas.microsoft.com/xps/2005/06/restricted-font">http://schemas.microsoft.com/xps/2005/06/restricted-font</a>
StartPart	<a href="http://schemas.microsoft.com/xps/2005/06/fixerepresentation">http://schemas.microsoft.com/xps/2005/06/fixerepresentation</a>
StoryFragments	<a href="http://schemas.microsoft.com/xps/2005/06/storyfragments">http://schemas.microsoft.com/xps/2005/06/storyfragments</a>



1

## 2 I. Conformance Requirements

### 3 This annex is informative

4 This annex summarizes all conformance requirements for producers and consumers  
5 implementing the XML Paper Specification. It is intended as a convenience; the text  
6 in the referenced clause or subclause is considered normative in all cases.

7 Conformance requirements are divided into three tables per clause below, containing  
8 the requirements that producers and consumers must follow, those that they should  
9 follow, and those that are optional. Each conformance requirement is given a unique  
10 ID comprised of a letter (M – MUST; S – SHOULD; O – OPTIONAL), the clause in  
11 which the requirement first appears, and a unique ID within that clause. Producers  
12 and consumers may use these IDs to report error conditions.

13 Additionally, the table identifies who is burdened with enforcing or supporting the  
14 requirement—the producer of content for that format or the consumer of content in  
15 that format—marked with an “x” in the appropriate column. The consumer could be  
16 required simply to validate that the producer correctly enforced a requirement in the  
17 generation of an XPS Document; these cases are marked with a “v” instead of an  
18 “x”. In certain cases, a requirement only applies to certain producers or consumers;  
19 these are marked with a superscripted letter referenced at the end of the table.

20 Consumers MUST support the usage of conformance rules marked as “OPTIONAL”  
21 and “SHOULD” only for producers if the consumer accesses the referenced feature. If  
22 they do not access the referenced feature they must ignore the manifestation of the  
23 rule without error.

24 In addition to the conformance requirements identified below, producers and  
25 consumers must meet the conformance requirements described in the OPC.

---

### 26 I.1 XPS Document Format

27 *Table I-1. XPS Document format MUST conformance requirements*

ID	Rule	Reference	Producer	Consumer
M1.1	XPS Documents MUST observe all conformance requirements of the OPC specification, except where specifically noted otherwise in this specification.	Clause 1	x	x
M1.2	The XPS Document format MUST use a ZIP archive for its physical model.	1.2	x	x

1 **I.2 Parts and Relationships**2 *Table I-2. Parts and Relationships MUST conformance requirements*

<b>ID</b>	<b>Rule</b>	<b>Reference</b>	<b>Producer</b>	<b>Consumer</b>
M2.1	All content to be rendered MUST be contained in the XPS Document.	2.1	x	v
M2.2	Each part contained in an XPS Document MUST use only the appropriate content type.	2.1	x	v <sup>U</sup>
M2.3	An XPS Document MUST contain exactly one FixedDocumentSequence part per fixed payload.	2.1	x	v
M2.4	An XPS Document MUST contain at least one FixedDocument part per fixed payload.	2.1	x	v
M2.5	An XPS Document MUST contain at least one FixedPage part per fixed payload.	2.1	x	v
M2.6	A <Glyphs> element in FixedPage markup MUST reference a Font part that exists in the XPS Document.	2.1	x	v
M2.7	An <ImageBrush> element in FixedPage markup MUST reference an Image part that exists in the XPS Document.	2.1	x	v
M2.8	If FixedPage markup references a Remote Resource Dictionary part, it MUST be included in the XPS Document	2.1	x	v
M2.10	Resources, which include fonts, images, color profiles, and remote resource dictionaries, that are referenced by URIs in FixedPage markup MUST use the Required Resource relationship from the FixedPage to the resource. If any resource references other resources, the indirectly required resource is also targeted by a Required Resource relationship from the FixedPage to the indirectly required resource.	2.1.1	x	v
M2.12	A Restricted Font relationship is REQUIRED for each print and preview font used, from the FixedDocument part to the preview and print Font part. When invoking editing functionality, an editing consumer MUST treat as an error any font with the print and preview licensing intent bit set for which no Restricted Font relationship has been added to the FixedDocument part. Printing and display-only consumers MUST consider an XPS Document valid, even if the producer failed to properly set the Restricted Font relationship.	2.1.1, 2.1.7.2, 2.1.7.4	x	v <sup>E</sup>
M2.13	Exactly one StartPart relationship is REQUIRED.	2.1.1	x	v
M2.14	The StartPart relationship MUST point from the package to the FixedDocumentSequence part that is the primary fixed payload root.	2.1.2	x	v

M2.15	The order of <DocumentReference> elements in a FixedDocumentSequence part MUST be preserved by editing consumers.	2.1.2	x <sup>E</sup>
M2.16	The order of <PageContent> elements in a FixedDocument MUST be preserved by editing consumers.	2.1.3	x <sup>E</sup>
M2.17	JPEG image parts MUST contain images that conform to the JPEG specification.	2.1.5.1	x v <sup>U</sup>
M2.18	PNG image parts MUST contain images that conform to the PNG specification.	2.1.5.2	x v <sup>U</sup>
M2.19	The PNG ancillary chunk tRNS MUST be supported.	2.1.5.2	x
M2.20	The PNG ancillary chunk iCCP MUST be supported.	2.1.5.2	x
M2.21	The PNG ancillary chunk sRGB MUST be ignored.	2.1.5.2	x
M2.22	The PNG ancillary chunk cHRM MUST be ignored.	2.1.5.2	x
M2.23	The PNG ancillary chunk gAMA MUST be ignored.	2.1.5.2	x
M2.24	The PNG ancillary chunk sBIT MUST be ignored.	2.1.5.2	x
M2.25	TIFF image parts MUST contain images that conform to the TIFF specification	2.1.5.3	x v <sup>U</sup>
M2.26	XPS Document consumers MUST support baseline TIFF 6.0 with the tag values described in Table 9-5 for the specified TIFF image types, excepting the tags described in §9.1.5.3, "TIFF Images."	2.1.5.3	x
M2.27	If a TIFF file contains multiple image file directories (IFDs), consumers MUST use only the first IFD and ignore all others.	2.1.5.3	x
M2.28	XPS Document consumers MUST support TIFF images using CCITT bilevel encoding.	2.1.5.3	x
M2.29	XPS Document consumers MUST support CMYK TIFF images.	2.1.5.3	x
M2.30	XPS Document consumers MUST support TIFF images with associated alpha data. If the ExtraSamples tag is 1, the alpha is treated as pre-multiplied alpha. With an ExtraSamples tag of 2, the alpha is treated as non-pre-multiplied alpha.	2.1.5.3	x
M2.31	XPS Document consumers MUST support TIFF images using LZW compression.	2.1.5.3	x
M2.32	XPS Document consumers MUST support TIFF images using differencing predictors.	2.1.5.3	x
M2.33	XPS Document consumers MUST support TIFF images using JPEG compression (compression mode 6 only).	2.1.5.3	x
M2.34	XPS Document consumers MUST support TIFF images with an embedded ICC profile.	2.1.5.3	x
M2.35	Windows Media Photo image files MUST conform to the Windows Media Photo specification.	2.1.5.4	x v <sup>U</sup>
M2.36	Each FixedPage part MUST NOT have more than one	2.1.6	x v <sup>T</sup>

I Conformance Requirements

	thumbnail part attached.			†
M2.37	Thumbnails MUST be either JPEG or PNG images	2.1.6	x	v <sup>†</sup>
M2.38	If using a fragment in the <b>FontURI</b> attribute of the <Glyphs> element to indicate the font face to use from a TrueType Collection, the attribute value MUST be an integer between 0 and <i>n</i> -1 inclusive, where <i>n</i> is the number of font faces in the TrueType Collection.	2.1.7	x	v
M2.39	All fonts used in XPS Documents MUST adhere to the OpenType font format, which includes TrueType and CFF fonts. A subsetted font MUST still be a valid OpenType font file.	2.1.7.1	x	v
M2.40	Producers MUST honor the licensing rights specified in OpenType fonts by following the embedding and obfuscation mechanisms described in this specification.	2.1.7.2	x	
M2.41	Consumers MUST be able to process XPS Documents using any combination of the embedding and obfuscation mechanisms described in this specification (even if produced in violation of the production requirements).	2.1.7.2		x
M2.42	For fonts with "Restricted license embedding" licensing intent, producers MUST NOT embed the font.	2.1.7.2	x	
M2.43	For fonts with "Print and preview embedding" licensing intent, consumers MUST NOT edit or modify any part of the XPS Document markup or hierarchical structure from the FixedDocument containing such a font downwards.	2.1.7.2		x <sup>E</sup>
M2.44	For fonts with "Print and preview embedding" licensing intent, producers MUST perform embedded font obfuscation.	2.1.7.2	x	
M2.45	For fonts with "Print and preview embedding" licensing intent, consumers MUST NOT extract or permanently install the font.	2.1.7.2		x
M2.46	For fonts with "Editable embedding" licensing intent, producers MUST perform embedded font obfuscation.	2.1.7.2	x	
M2.47	For fonts with "Editable embedding" licensing intent, consumers MUST NOT extract or permanently install the font.	2.1.7.2		x
M2.48	For fonts with "No subsetting" licensing intent, producers MUST perform embedded font obfuscation.	2.1.7.2	x	
M2.49	For fonts with "No subsetting" licensing intent, producers MUST NOT subset the font.	2.1.7.2	x	
M2.50	For fonts with "No subsetting" licensing intent, consumers MUST NOT extract or permanently install the font.	2.1.7.2		x
M2.51	For fonts with "Bitmap embedding only" licensing intent, producers MUST perform embedded font obfuscation for bitmap characters only. If no bitmap characters are present in the font, the producer MUST NOT embed the font.	2.1.7.2	x	

M2.52	For fonts with "Bitmap embedding only" licensing intent, consumers MUST NOT extract or permanently install the font.	2.1.7.2	x	
M2.53	Producers and consumers MUST perform font obfuscation and de-obfuscation according to the steps described in §9.1.7.3, "Embedded Font Obfuscation."	2.1.7.3	x	x
M2.54	The last segment of the part name for an obfuscated font MUST be the GUID generated during the font obfuscation process, with or without an extension.	2.1.7.3	x	v
M2.55	When processing <Glyphs> elements, the consumer MUST select a cmap table from the OpenType font according to Table 1-8 in §9.1.7.5, "Non-Standard Font Compatibility Encoding." All further processing for that font MUST use the selected cmap table.	2.1.7.5	x	x
M2.56	When processing <Glyphs> elements, if a WanSung, Big5, Prc, ShiftJis, or MacRoman cmap has been selected, the consumer MUST correctly map from Unicode codepoints in the <b>UnicodeString</b> attribute to the corresponding codepoints used by the cmap before looking up glyphs.	2.1.7.5		x
M2.57	When processing <Glyphs> elements that reference a cmap (3,0) encoding font, consumers MUST handle the case where the <b>UnicodeString</b> attribute contains character codes instead of PUA codepoints by computing the correct glyph index according to the general recommendations of the OpenType specification.	2.1.7.5		x
M2.58	Consumers MUST process all PrintTicket parts when an XPS Document is printed.	2.1.9		x <sup>P</sup>
M2.59	A level-specific PrintTicket MUST contain only settings scoped to the current level and child levels. Job-level PrintTicket parts MUST contain only job-, document-, and page-scoped settings; document-level PrintTicket parts MUST contain only document-scoped and page-scoped settings; and page-level PrintTicket parts MUST contain only page-scoped settings. Print schema elements that interact between levels MUST be specified at the root of each level ticket. Each FixedDocumentSequence, FixedDocument, or FixedPage part MUST have no more than one attached PrintTicket.	2.1.9.2	x <sup>U</sup>	
M2.60	Consumers MUST process job-level, document-level and page-level settings of PrintTicket parts associated with FixedDocumentSequence parts.	2.1.9.2		x <sup>P</sup>
M2.61	Consumers MUST process document-level and page-level settings of PrintTicket parts associated with FixedDocument parts and MUST ignore job-level settings of PrintTicket parts associated with FixedDocument parts.	2.1.9.2		x <sup>P</sup>
M2.62	Consumers MUST process page-level settings of PrintTicket parts associated with FixedPage parts and MUST ignore job-level and document-level settings of PrintTicket parts associated with FixedPage parts.	2.1.9.2		x <sup>P</sup>

I Conformance Requirements

M2.63	When processing a PrintTicket, consumers MUST first remove all levels of PrintTicket content not applicable to the current element.	2.1.9.3	x <sup>P</sup>
M2.64	When processing a PrintTicket, consumers MUST second validate the PrintTicket according to the methods defined in the PrintTicket Validation Checklist of the Print Schema documentation.	2.1.9.3	x <sup>P</sup>
M2.65	Following validation of a PrintTicket, the printing consumer MUST properly interpret the print settings according to the rules for merging two PrintTicket parts.	2.1.9.3	x <sup>P</sup>
M2.66	If there is no print setting merge conflict between different PrintTicket levels, a prefix-scoped element MUST be pushed down, or inherited, from a more general ticket to a more specific ticket. This case is isomorphic to the case where both tickets contain an identical element.	2.1.9.3	x <sup>P</sup>
M2.67	If there is a print setting merge conflict between different PrintTicket levels, the setting from the most specific ticket MUST take precedence.	2.1.9.3	x <sup>P</sup>
M2.68	Consumers MUST use semantic document structure provided in included DocumentStructure and StoryFragments parts in preference to any other analysis method of generating such structure.	2.1.11	x
M2.69	Consumers MUST support Markup Compatibility elements and attributes in DocumentStructure, FixedDocument, FixedDocumentSequence, FixedPage, Relationships, Remote Resource Dictionary, SignatureDefinitions, and StoryFragments parts.  Before attempting to validate one of these parts against a schema, consumers MUST remove all Markup Compatibility elements and attributes, ignorable namespace declarations, and all ignored elements and attributes not defined in the expected version of XPS Document markup.	2.3.1	x
M2.70	XML content MUST be encoded using either UTF-8 or UTF-16. If any such part includes an encoding declaration (as defined in §4.3.3 of the XML specification), that declaration MUST NOT name any encoding other than UTF-8 or UTF-16.	2.3.2	x v
M2.71	DTD content MUST NOT be used in the XML markup defined in this specification, and consumers MUST treat the presence of DTD content as an error.	2.3.2	x x
M2.72	XML content MUST be valid against the corresponding XSD schema defined in this specification. In particular, the XML content MUST NOT contain elements or attributes drawn from namespaces that are not explicitly defined in the corresponding XSD unless the XSD allows elements or attributes drawn from any namespace to be present in particular locations in the XML markup.	2.3.2	x v



M2.73	XML content MUST NOT contain elements or attributes drawn from "xml" or "xsi" namespaces unless they are explicitly defined in the XSD schema or by other means in the specification.	2.3.2	x	v
M2.74	Properties MUST NOT be set more than once, regardless of the syntax used to specify the value. In certain cases, they can be specified using either property attributes or property elements. Consumers MUST treat properties that are specified in both ways as an error.	2.3.3.2	x	v
M2.75	XPS Document markup MUST NOT use the <b>xml:space</b> attribute.	2.3.4	x	v
M2.76	The language of the contents of an XPS Document MUST be identified using the <b>xml:lang</b> attribute, the value of which is inherited by child and descendant elements. When the language of the contents is unknown and is required, the value "und" (undetermined) MUST be used.	2.3.5.1	x	
M2.77	Producers that generate a relationship MUST include the target part in the XPS Document for any of the following relationship types: DiscardControl, DocumentStructure, PrintTicket, Required Resource, Restricted Font, StartPart, StoryFragments, and Thumbnail. Consumers that access the target part of any relationship with one of these relationship types MUST generate an error if the part is not included in the XPS Document.	2.1.1	x <sup>U</sup>	v <sup>U</sup>
M2.78	Consumers MUST support JPEG images that contain the APP1 marker and interpret the EXIF color space correctly.	2.1.5.1	x	
M2.79	XPS Document consumers MUST support TIFF images that include the EXIF IFD (tag 34665) as described in the EXIF specification. The EXIF color space MUST be interpreted correctly.	2.1.5.3	x	

1 **Notes:**

- 2 E — Only applies to editing consumers.
- 3 P — Only applies to printing consumers.
- 4 T — Only applies if thumbnails are used by the consumer.
- 5 U — Only required for a producer if the part is generated or for a consumer when
- 6 accessing the part.

7 *Table I-3. Parts and Relationships SHOULD conformance requirements*

ID	Rule	Reference	Producer	Consumer
S2.1	It is RECOMMENDED that there be exactly one Required Resource relationship from the FixedPage part for each resource referenced from the markup.	2.1.1	x	

I Conformance Requirements

S2.2	Each <DocumentReference> element in a FixedDocumentSequence part SHOULD reference a FixedDocument part by relative URI.	2.1.2	×	
S2.3	Each <PageContent> element in a FixedDocument part SHOULD reference a FixedPage part by relative URI.	2.1.3	×	
S2.4	<ImageBrush> and <Glyphs> elements SHOULD reference Image and Font parts by relative URI.	2.1.4	×	
S2.5	Color profiles embedded in image files SHOULD be used if present and compatible with this specification.	2.1.5		×
S2.6	It is RECOMMENDED that JPEG image part names end with the extension “.jpg”.	2.1.5.1	×	
S2.7	The use of CMYK JPEG images is NOT RECOMMENDED. TIFF or Windows Media Photo images SHOULD be used instead to represent CMYK images.	2.1.5.1	×	
S2.8	It is RECOMMENDED that PNG image part names end with the extension “.png”.	2.1.5.2	×	
S2.9	It is RECOMMENDED that TIFF image part names end with the extension “.tif”.	2.1.5.3	×	
S2.10	Consumers SHOULD ignore unsupported TIFF tags (those not described in Table 9–5 and §9.1.5.3, “TIFF Images”). Producers SHOULD NOT include unsupported tags.	2.1.5.3	×	×
S2.11	Given the wide variety of incompliant TIFF images in circulation, consumers SHOULD test as many different TIFF images as possible, correct common mistakes in TIFF images, and implement a reasonable recovery strategy when a problematic TIFF image is encountered.	2.1.5.3		×
S2.12	It is RECOMMENDED that Windows Media Photo images end with the extension “.wdp”.	2.1.5.4	×	
S2.13	It is RECOMMENDED that if thumbnails are used for pages, a thumbnail SHOULD be included for every page in the document.	2.1.6	×	
S2.14	Consumers SHOULD only process thumbnails associated via a package relationship from the package as a whole or via a relationship from a FixedPage part. Thumbnails attached to any other part SHOULD be ignored.	2.1.6		×
S2.15	Producers SHOULD use Unicode-encoded fonts.	2.1.7, 2.1.7.5	×	
S2.16	For fonts with “Installable embedding” licensing intent, producers SHOULD perform embedded font obfuscation.	2.1.7.2	×	
S2.17	For fonts with “Installable embedding” licensing intent, consumers SHOULD NOT extract or permanently install the font.	2.1.7.2		×
S2.18	For fonts with “Restricted license embedding” licensing intent, producers SHOULD generate a path filled with an image brush referencing an image of rendered characters and SHOULD include the actual text in the	2.1.7.2	×	

<b>AutomationProperties.Name</b> attribute of the <Path> element.		
S2.19	Although the licensing intent allows embedding of non-obfuscated fonts and installation of the font on a remote client system under certain conditions, this is NOT RECOMMENDED in XPS Documents. Instead, producers SHOULD always perform font obfuscation, and consumers SHOULD never extract or permanently install fonts.	2.1.7.3 x x
S2.20	It is RECOMMENDED that the extension of an obfuscated Font part name be ".odttf" for TrueType fonts and ".odttc" for TrueType collections.	2.1.7.3 x
S2.21	Producers SHOULD include only PrintTicket settings that support portability of the XPS Document.	2.1.9.1 x
S2.22	Producers SHOULD only attach PrintTicket parts containing only document-level and page-level settings with FixedDocument parts.	2.1.9.2 x
S2.23	Producers SHOULD only attach PrintTicket parts containing only page-level settings with FixedPage parts.	2.1.9.3 x
S2.24	The FixedDocumentSequence part SHOULD follow the part name recommendation "/<FixedDocSeq>.fdseq" where <FixedDocSeq> is the name of the FixedDocumentSequence. The FixedDocumentSequence SHOULD use the extension ".fdseq".	2.2 x
S2.25	A FixedDocument part SHOULD follow the part name recommendation "/Documents/<n>/<FixedDocument>.fdoc" where <n> is a numeral that indicates the ordinal position of the fixed document in the fixed document sequence and <FixedDocument> is the name of the fixed document. FixedDocument parts SHOULD use the extension ".fdoc".	2.2 x
S2.26	A FixedPage part SHOULD follow the part name recommendation "/Documents/<n>/Pages/<m>.fpage" where <n> represents the document that includes this page and <m> is the page number. FixedPage parts SHOULD use the extension ".fpage".	2.2 x
S2.27	A resource that is specific to a particular document SHOULD follow the part name recommendation "/Documents/<n>/Resources/<Resource>" where <n> is the document that uses the resource and <Resource> is the segments identifying the particular resource. A resource that is shared across multiple documents SHOULD follow the part name recommendation "/Resources/<Resource>". A Font resource SHOULD use "Fonts/<FontName>.<FontExt>" for its <Resource> value, where <FontExt> SHOULD be either ".ttf" or ".odttf" for non-obfuscated and obfuscated fonts respectively. An Image resource SHOULD use "Images/<ImageName>.<ImageExt>" for its <Resource>	2.2 x

I Conformance Requirements

---

<p>value, where <i>&lt;ImageExt&gt;</i> is the correct extension for the image type.</p> <p>A Remote Resource Dictionary resource SHOULD use "Dictionaries/<i>&lt;DictName&gt;.dict</i>" for its <i>&lt;Resource&gt;</i> value. A Remote Resource Dictionary SHOULD use ".dict" as its extension.</p> <p><i>&lt;FontName&gt;</i>, <i>&lt;ImageName&gt;</i>, and <i>&lt;DictName&gt;</i> SHOULD be a string representation of a GUID value if the resource is a shared resource.</p>			
<p>S2.28 A DocumentStructure part SHOULD follow the part name recommendation "/Documents/<i>&lt;n&gt;/Structure/&lt;DocStruct&gt;.struct</i>" where <i>&lt;n&gt;</i> is the fixed document that the document structure applies to. DocumentStructure parts SHOULD use the extension ".struct".</p>	2.2	x	
<p>S2.29 A StoryFragments part SHOULD follow the part name recommendation "/Documents/<i>&lt;n&gt;/Structure/Fragments/&lt;m&gt;.frag</i>" where <i>&lt;n&gt;</i> is the fixed document that contains the story fragments and <i>&lt;m&gt;</i> is the page number the StoryFragments part applies to. StoryFragments parts SHOULD use the extension ".frag".</p>	2.2	x	
<p>S2.30 ICC profile parts SHOULD follow the part name recommendation "/Documents/<i>&lt;n&gt;/Metadata/&lt;ProfileName&gt;.&lt;ProfileExt&gt;</i>" where <i>&lt;n&gt;</i> is the fixed document that contains the profile. ICC profile parts shared across multiple documents SHOULD follow the part name recommendation "/Metadata/<i>&lt;ProfileName&gt;.&lt;ProfileExt&gt;</i>". In this case, <i>&lt;ProfileName&gt;</i> SHOULD be a string representation of a GUID value. The <i>&lt;ProfileExt&gt;</i> SHOULD be appropriate to the color profile type, such as ".icm".</p>	2.2	x	
<p>S2.31 Thumbnail parts SHOULD follow the part name recommendation "/Documents/<i>&lt;n&gt;/Metadata/&lt;ThumbName&gt;.&lt;ThumbExt&gt;</i>" where <i>&lt;n&gt;</i> is the fixed document that contains the profile. If the Thumbnail part represents the package as a whole, it SHOULD follow the part name recommendation "/Metadata/<i>&lt;ThumbName&gt;.&lt;ThumbExt&gt;</i>". In this case, <i>&lt;ThumbName&gt;</i> SHOULD be a string representation of a GUID value. The <i>&lt;ThumbExt&gt;</i> SHOULD be appropriate to the image type, either ".png" or ".jpg".</p>	2.2	x	
<p>S2.32 PrintTicket part names associated with the entire job SHOULD be associated via relationship with the FixedDocumentSequence part and follow the part name recommendation "/Metadata/<i>&lt;PrintTicketName&gt;.xml</i>". PrintTicket parts associated with a particular fixed document</p>	2.2	x	

---

	or fixed page SHOULD follow the part name recommendation "/Documents/<n>/Metadata/<PrintTicketName>.xml", where <n> is the fixed document that uses these parts. PrintTicket parts SHOULD use the extension ".xml".		
S2.33	The names of any non-standard parts that are associated with a particular fixed document SHOULD follow the part name recommendation "/Documents/<n>/Other/<PartName>", where <n> is the fixed document to which the part belongs.	2.2	x
S2.34	Consumers SHOULD support JPEG images that contain APP0, APP2, APP13, and APP14 markers.	2.1.5.1	x

1 Table I-4. Parts and Relationships OPTIONAL conformance requirements

ID	Rule	Reference		
			Producer	Consumer
O2.1	Thumbnail parts MAY be included in an XPS Document	2.1	x	
O2.2	PrintTicket parts MAY be included in an XPS Document.	2.1	x	
O2.3	ICC Profile parts MAY be included in an XPS Document.	2.1	x	
O2.4	DocumentStructure parts MAY be included in an XPS Document.	2.1	x	
O2.5	StoryFragments parts MAY be included in an XPS Document.	2.1	x	
O2.6	SignatureDefinitions parts MAY be included in an XPS Document.	2.1	x	
O2.7	DiscardControl parts MAY be included in an XPS Document.	2.1	x	
O2.8	A Core Properties relationship MAY be included in an XPS Document, from the package to the Core Properties part.	2.1.1	x	
O2.9	A Digital Signatures Origin relationship MAY be included in an XPS Document, from the package to the Digital Signature Origin part.	2.1.1	x	
O2.10	Digital Signature relationships MAY be included in an XPS Document, from the Digital Signature Origin part to a Digital Signature XML Signature part.	2.1.1	x	
O2.11	Digital Signature Certificate relationships MAY be included in an XPS Document, from a Digital Signature XML Signature part to the Digital Signature Certificate part.	2.1.1	x	
O2.12	Digital Signature Definitions parts MAY be included in an XPS Document, from a FixedDocument part to the Digital Signature Definitions part.	2.1.1	x	
O2.13	DiscardControl relationships MAY be included in an XPS Document, from the package to a DiscardControl part.	2.1.1	x	
O2.14	DocumentStructure relationships MAY be included in an XPS Document, from a FixedDocument part to the DocumentStructure part.	2.1.1	x	

I Conformance Requirements

O2.15	PrintTicket relationships MAY be included in an XPS Document, from a FixedDocumentSequence, FixedDocument, or FixedPage part to a PrintTicket part.	2.1.1	×
O2.16	StoryFragments relationships MAY be included in an XPS Document, from a FixedPage part to a StoryFragments part.	2.1.1	×
O2.17	Thumbnail relationships MAY be included in an XPS Document, from the package to an Image part or from a FixedPage part to an Image part.	2.1.1	×
O2.18	Color Profiles MAY be embedded in image files.	2.1.5	×
O2.19	Thumbnail images MAY be attached to a FixedPage part using a Thumbnail relationship.	2.1.6	×
O2.20	Fonts MAY be subsetted based on glyph usage.	2.1.7.1	×
O2.21	Producers MAY use a 128-bit random number instead of a true GUID for an obfuscated font name.	2.1.7.3	×
O2.22	An obfuscated Font part MAY have an arbitrary extension.	2.1.7.3	×
O2.23	Producers MAY add digital signature requests and instructions to an XPS Document in the form of signature definitions.	2.1.10	×
O2.24	A producer MAY sign against an existing signature definition to provide additional signature information.	2.1.10	×
O2.25	A recipient of an XPS Document MAY also sign it against a signature definition.	2.1.10	×
O2.27	Consumers MAY provide an algorithmic construction of the structure of an XPS Document based on a page-layout analysis, provided such structure is not explicitly provided in DocumentStructure and StoryFragments parts.	2.1.11	×
O2.28	A resource that is intended to be used across multiple fixed documents MAY be named according to the guidelines for shared resources.	2.2	×
O2.29	Producers MAY include Markup Compatibility elements and attributes in DocumentStructure, FixedDocument, FixedDocumentSequence, FixedPage, Relationships, Remote Resource Dictionary, SignatureDefinitions, and StoryFragments parts.	2.3.1	×
O2.30	Wherever a single whitespace character is allowed in XPS Document markup, multiple whitespace characters MAY be used (unless explicitly restricted by a pattern restriction in the corresponding schema).	2.3.4	×
O2.31	Attributes in XPS Document markup that specify comma-delimited attribute values MAY, unless specified otherwise, OPTIONALLY include whitespace characters preceding or following the comma.	2.3.4	×
O2.32	Where the XPS Document schema specifies attributes of types that allow whitespace collapsing, leading and trailing whitespace in the attribute value MAY be used along with other whitespace that relies on the whitespace collapsing	2.3.4	×

---

behavior specified in the XML Schema Specification.

---

### 1 I.3 Documents

2 Table I-5. Document MUST conformance requirements

ID	Rule	Reference	Producer	Consumer
M3.1	The order of <DocumentReference> elements MUST match the order of the documents in the fixed document sequence.	3.1	x	
M3.2	The Source attribute of the <DocumentReference> element MUST specify a FixedDocument part within the XPS Document.	3.1.1	x	v
M3.3	Producers MUST NOT produce a document with multiple <DocumentReference> elements that reference the same fixed document.	3.1.1	x	v
M3.4	The order of <PageContent> elements MUST match the order of the pages in the document.	3.2	x	
M3.5	The Source attribute of the <PageContent> element MUST specify a FixedPage part within the XPS Document.	3.2.1	x	v
M3.6	Producers MUST NOT produce markup where a <PageContent> element references the same fixed page referenced by any other <PageContent> element in the entire XPS Document, even in other fixed documents within the fixed payload.	3.2.1	x	v
M3.7	If the attribute is specified, the <b>BleedBox</b> BleedOriginX value MUST be less than or equal to 0.	3.3.1	x	v <sup>F</sup>
M3.8	If the attribute is specified, the <b>BleedBox</b> BleedOriginY value MUST be less than or equal to 0.	3.3.1	x	v <sup>F</sup>
M3.9	If the attribute is specified, the <b>BleedBox</b> BleedWidth value MUST be greater than or equal to the Width attribute value plus the absolute value of the <b>BleedBox</b> BleedOriginX value.	3.3.1	x	v <sup>F</sup>
M3.10	If the attribute is specified, the <b>BleedBox</b> BleedHeight value MUST be greater than or equal to the fixed page <b>Height</b> value plus the absolute value of the <b>BleedBox</b> BleedOriginY value.	3.3.1	x	v <sup>F</sup>
M3.11	If the attribute is specified, the <b>ContentBox</b> ContentOriginX value MUST be greater than or equal to 0 and less than the fixed page <b>Width</b> attribute value	3.3.2	x	v <sup>F</sup>
M3.12	If the attribute is specified, the <b>ContentBox</b> ContentOriginY value MUST be greater than or equal to 0 and less than the fixed page <b>Height</b> attribute value.	3.3.2	x	v <sup>F</sup>
M3.13	If the attribute is specified, the <b>ContentBox</b> ContentWidth value MUST be less than or equal to the difference between	3.3.2	x	v <sup>F</sup>

I Conformance Requirements

	the fixed page <b>Width</b> attribute value and the <b>ContentBox</b> ContentOriginX value.			
M3.14	If the attribute is specified, the <b>ContentBox</b> ContentHeight value MUST be less than or equal to the difference between the fixed page <b>Height</b> attribute value and the <b>ContentBox</b> ContentOriginY value.	3.3.2	x	v <sup>F</sup>
M3.15	When rendering a fixed page for printing, consumers MUST be aware of the interaction between the fixed page markup and the PrintTicket settings.	3.3.4	x	v <sup>P</sup>
M3.16	When the PrintTicket specifies the page scaling option FitApplicationBleedSizeToPageImageableSize, printing consumers MUST scale the bleed box (producer bleed size) to the PageImageableSize, preserving the aspect ratio.	3.3.4.1		x <sup>P</sup>
M3.17	When the PrintTicket specifies the page scaling option FitApplicationContentSizeToPageImageableSize, printing consumers MUST scale the content box (producer content size) to the PageImageableSize, preserving the aspect ratio.	3.3.4.2		x <sup>P</sup>
M3.18	When the PrintTicket specifies the page scaling option FitApplicationMediaSizeToPageImageableSize, printing consumers MUST scale the height and width (producer media size) to the PageImageableSize, preserving the aspect ratio.	3.3.4.3		x <sup>P</sup>
M3.19	When the PrintTicket specifies the page scaling option FitApplicationMediaSizeToPageMediaSize, printing consumers MUST scale the height and width (producer media size) to the PageMediaSize, preserving the aspect ratio.	3.3.4.4		x <sup>P</sup>
M3.20	The <b>x:Key</b> attribute of the <Canvas> element MUST be present when the element is defined in a resource dictionary. It MUST NOT be specified outside a resource dictionary.	3.4	x	v

1 **Notes:**

- 2 F — Only necessary if the referenced feature is used.
- 3 P Only applies to printing consumers.
- 4 *Table I-6. Document SHOULD conformance requirements*

ID	Rule			
		Reference	Producer	Consumer
S3.1	Specifying a ContentBox attribute for the <FixedPage> element is RECOMMENDED.	3.3	x	
S3.2	Invalid bleed box specifications SHOULD be ignored in favor of the default bleed box.	3.3.1		x
S3.3	Invalid content box specifications SHOULD NOT be rendered and SHOULD generate an error.	3.3.1		x



S3.4	In the absence of media scaling, the fixed page content is imaged directly to the physical media with the origin of the fixed page aligned with the origin of the physical media size. Any fixed page content that extends beyond the dimension of the physical media size SHOULD be clipped.	3.3.4	x <sup>P</sup>
------	---	-------	----------------

1 **Notes:**

2 P — Only applies to printing consumers.

3 **I.4 Graphics**4 *Table I-7. Graphics MUST conformance requirements*

ID	Rule	Reference	Producer	Consumer
M4.1	The <b>x:Key</b> attribute of the <Path> element MUST be present when the element is defined in a resource dictionary. It MUST NOT be specified outside a resource dictionary.	4.1	x	v
M4.2	The <b>x:Key</b> attribute of the <PathGeometry> element MUST be present when the element is defined in a resource dictionary. It MUST NOT be specified outside a resource dictionary.	4.2.1.1	x	v
M4.3	A <PathGeometry> element contains a set of path figures specified either with the <b>Figures</b> attribute or with a child <PathFigure> element. Producers MUST NOT specify the path figures of a geometry with both the <b>Figures</b> attribute and a child <PathFigure> element.	4.2.1.1	x	v

5 *Table I-8. Graphics SHOULD conformance requirements*

ID	Rule	Reference	Producer	Consumer
S4.1	Line segments and curve segments SHOULD NOT be specified as zero-length.	4.2.2.1	x	

6 *Table I-9. Graphics OPTIONAL conformance requirements*

ID	Rule	Reference	Producer	Consumer
O4.1	Consumers or viewers that perform anti-aliasing MAY "snap" those control points of the path that are situated on the path bounding box to whole device pixels if the ignorable	4.1		x

---

**SnapsToDevicePixels** attribute is specified as true.

---

## 1 I.5 Text

2 Table I-10. Text *MUST* conformance requirements

ID	Rule	Reference	Producer	Consumer
M5.1	If the <b>CaretStops</b> attribute is missing from the <Glyphs> element, a consumer <i>MUST</i> interpret the text as having a caret stop between each Unicode UTF-16 code unit and at the beginning and end of the text.	5.1		x <sup>s</sup>
M5.2	If the <b>UnicodeString</b> attribute of the <Glyphs> element specifies an empty string (" or "{}") and the <b>Indices</b> attribute is not specified or is empty, the consumer <i>MUST</i> generate an error.	5.1	x	v
M5.3	The <b>x:Key</b> attribute of the <Glyphs> element <i>MUST</i> be present when the element is defined in a resource dictionary. It <i>MUST NOT</i> be specified outside a resource dictionary.	5.1	x	v
M5.4	The sum of the code unit counts for all the GlyphMapping entries in the <b>Indices</b> attribute <i>MUST NOT</i> exceed the number of UTF-16 code units in the <b>UnicodeString</b> attribute if the <b>UnicodeString</b> attribute is specified and does not contain an empty value (" or "{}"). If a ClusterMapping is not specified within a GlyphMapping entry, the code unit count is 1. If the <b>Indices</b> attribute specifies a GlyphIndex that does not exist in the font, the consumer <i>MUST</i> generate an error.	5.1.3	x	v
M5.5	If there is not a one-to-one mapping between code units in the <b>UnicodeString</b> attribute and the glyph indices, the GlyphIndex value in the <b>Indices</b> attribute <i>MUST</i> be specified.	5.1.3	x	
M5.6	The AdvanceWidth of the <b>Indices</b> attribute <i>MUST</i> be calculated as the exact unrounded origin of the subsequent glyph minus the sum of the calculated (that is, rounded) advance widths of the preceding glyphs.	5.1.3	x	
M5.7	A <b>UnicodeString</b> attribute value that begins with an open brace ("{" ) <i>MUST</i> be escaped with a prefix of "{}". If a <b>UnicodeString</b> attribute value starts with "{}", consumers <i>MUST</i> ignore those first two characters in processing the <b>UnicodeString</b> and in calculating index positions for the characters of the <b>UnicodeString</b> .	5.1.4	x	x
M5.9	If the <b>UnicodeString</b> attribute contains a Unicode code unit that cannot be mapped to a glyph index via a cmap table in the font and there is no corresponding GlyphIndex entry in	5.1.4	x	x

	the <b>Indices</b> attribute, the consumer MUST display the .notdef glyph			
M5.10	In the absence of entries in the <b>Indices</b> attribute to override the Unicode code units in the <b>UnicodeString</b> attribute value, consumers MUST treat Unicode control marks in the <b>UnicodeString</b> attribute like ordinary characters and render the glyphs to which the Unicode control marks are mapped in the CMAP table.	5.1.4	x	
M5.11	Because advance-widths, glyph indices, and caret-stops are associated with the generated Unicode string, consumers MUST NOT normalize the <b>UnicodeString</b> attribute value to produce an internal representation.	5.1.4	x	
M5.12	Producers MUST lay out algorithmically emboldened glyphs using advance widths that are 2% of the em size larger than when not algorithmically emboldened.	5.1.5	x	
M5.13	Consumers MUST implement the effect of algorithmic emboldening such that the black box of the glyph grows by 2% of the em size. When advance widths are omitted from the markup and the glyphs are algorithmically emboldened, the advance widths obtained from the horizontal metrics font table (if <b>IsSideways</b> is false) or the vertical metrics font table (if <b>IsSideways</b> is true) of the font MUST be increased by 2% of the em size.	5.1.5	x	
M5.14	Producers MUST lay out algorithmically italicized glyphs using exactly the same advance widths as when not algorithmically italicized.	5.1.5	x	
M5.15	Producers MUST NOT specify text that is both right-to-left ( <b>BidiLevel</b> attribute value of 1) and vertical ( <b>IsSideways</b> attribute set to true).	5.1.6.2	x	v
M5.16	If a consumer does not understand the specified device font name, it MUST render the embedded version of the font.	5.1.7	x <sup>P</sup>	
M5.17	When rendering a printer device font, consumers MUST use the <b>UnicodeString</b> attribute and ignore the glyph index components of the <b>Indices</b> attribute.	5.1.7	x <sup>P</sup>	
M5.18	When rendering a printer device font, consumers MUST still honor the advance width and x,y offset values present in the <b>Indices</b> attribute.	5.1.7	x <sup>P</sup>	
M5.19	For producers, a <Glyphs> element with a specified device font name MUST have exactly one <b>Indices</b> glyph per character in the <b>UnicodeString</b> attribute. Its <b>Indices</b> attribute MUST NOT include any cluster specifications. If the <b>Indices</b> attribute includes a cluster mapping, the consumer MUST NOT use the device font name and MUST render the embedded version of the font.	5.1.7	x	v <sup>FP</sup>
M5.20	For producers of a <Glyphs> element with a specified device font name, each of the <b>Indices</b> glyphs MUST include a specified advance width and MUST include specified x and y offset values if they are non-zero.	5.1.7	x	

I Conformance Requirements

M5.22	If there are insufficient flags in the <b>CaretStops</b> attribute value to correspond to all the UTF-16 code units in the <b>UnicodeString</b> attribute value, all remaining UTF-16 code units in the Unicode string MUST be considered valid caret stops.	5.1.9	x <sup>5</sup>
M5.23	If the <b>Indices</b> attribute is specified, the values provided MUST be used in preference to values determined from the <b>UnicodeString</b> attribute alone.	12.1.3	x

1 **Notes:**

- 2 F — Only necessary if the referenced feature is used.
- 3 P — Only applies to printing consumers.
- 4 S — Only applies to consumers that implement selection.

5 *Table I-11. Text SHOULD conformance requirements*

ID	Rule	Reference	Producer	Consumer
S5.1	The value of the <b>CaretStops</b> attribute SHOULD indicate that the caret cannot stop in front of most combining marks and the second UTF-16 code unit of UTF-16 surrogate pairs.	5.1	x	
S5.2	If producers include control marks in the Unicode string, they SHOULD include an <b>Indices</b> attribute to specify glyph indices and/or character-to-glyph mapping information for the control marks.	5.1.4	x	
S5.3	If alternate vertical character representations are available in the font, the producer SHOULD use those in preference to the <b>IsSideways</b> attribute and provide their glyph indices in the <b>Indices</b> attribute.	5.1.6	x	
S5.4	Producers SHOULD NOT produce markup that will result in different rendering between consumers using the embedded font to render and consumers using the device font to render.	5.1.7	x	
S5.5	Specifying a <b>UnicodeString</b> for <Glyphs> elements is RECOMMENDED, as it supports searching, selection, and accessibility.	5.1.4	x	

6 *Table I-12. Text OPTIONAL conformance requirements*

ID	Rule	Reference	Producer	Consumer
O5.1	Producers MAY include Unicode control marks in the Unicode string. Such marks include control codes, layout controls, invisible operators, deprecated format characters, variation selectors, non-characters, and specials, according to their	5.1.4	x	

	definition within the Unicode specification.		
O5.2	Producers MAY choose to generate <b>UnicodeString</b> attribute values that are not normalized by any Unicode-defined algorithm.	5.1.4	x
O5.3	Consumers that understand the device font name MAY ignore the embedded font and use the device-resident version.	5.1.7	x <sup>P</sup>
O5.4	Glyph indices MAY be omitted from markup where there is a one-to-one mapping between the positions of Unicode scalar values in the <b>UnicodeString</b> attribute and the positions of glyphs in the glyph string and the glyph index is the value in selected character mapping table of the font.	5.1.10.1	x
O5.5	Glyph advance widths MAY be omitted from markup where the advance width desired is specified in the font tables, once adjusted for algorithmic emboldening.	5.1.10.2	x
O5.6	Glyph horizontal and vertical offsets MAY be omitted from markup where the offset is 0.0.	5.1.10.2	x

1 **Notes:**

2 P — Only applies to printing consumers.

3 **I.6 Brushes**4 *Table I-13. Brushes MUST conformance requirements*

ID	Rule	Reference	Producer	Consumer
M6.1	The <b>x:Key</b> attribute of the <SolidColorBrush> element MUST be present when the element is defined in a resource dictionary. It MUST NOT be specified outside a resource dictionary.	6.1	x	v
M6.2	The <b>x:Key</b> attribute of the <ImageBrush> element MUST be present when the element is defined in a resource dictionary. It MUST NOT be specified outside a resource dictionary.	6.2	x	v
M6.3	An <ImageBrush> element MUST reference a JPEG, PNG, TIFF 6.0, or Windows Media Photo Image part within the XPS Document package.	6.2	x	v
M6.4	The <b>x:Key</b> attribute of the <VisualBrush> element MUST be present when the element is defined in a resource dictionary. It MUST NOT be specified outside a resource dictionary.	6.3	x	v
M6.5	The <b>x:Key</b> attribute of the <LinearGradientBrush> element MUST be present when the element is defined in a resource dictionary. It MUST NOT be specified outside a resource dictionary.	6.5	x	v
M6.6	The <b>x:Key</b> attribute of the <RadialGradientBrush> element MUST be present when the element is defined in a resource dictionary. It MUST NOT be specified outside a resource dictionary.	6.6	x	v

1 **I.7 Common Properties**2 *Table I-14. Common properties MUST conformance requirements*

ID	Rule	Reference	Producer	Consumer
M7.1	Individual resource values MUST be specified within a resource dictionary.	7.2	x	
M7.2	Namespace prefixes in resource definitions MUST apply in the context of the definition, rather than in the context of the resource reference.	7.2.3	x	x
M7.3	An <b>xml:lang</b> attribute within a resource definition MUST be interpreted in the context of the resource reference, not the resource definition.	7.2.3	x	x
M7.4	A remote resource dictionary MUST follow the requirements that apply to inline resource dictionaries.	7.2.3.1	x	v
M7.5	A remote resource dictionary MUST NOT contain any resource definition children that reference another remote resource dictionary.	7.2.3.1	x	v
M7.6	A <ResourceDictionary> element that specifies a remote resource dictionary in its <b>Source</b> attribute MUST NOT contain any resource definition children.	7.2.3.1	x	v
M7.7	Inline references to fonts or images in remote resource dictionary entries MUST be interpreted with the same base URI as the Remote Resource Dictionary part, not from the base URI of the part referring to the particular remote resource dictionary entry.	7.2.3.1	x	x
M7.8	When a resource definition references a previously defined resource with the same name in an ancestor resource dictionary, the reference MUST be resolved before the redefined resource is added to the dictionary	7.2.5	x	x
M7.9	If a resource definition references another resource, the reference MUST be resolved in the context of the resource definition, not in the context of the resource use.	7.2.5	x	x
M7.10	If a resource dictionary contains Markup Compatibility elements and attributes, the processing of the Markup Compatibility markup MUST occur in the context of the definition of the resource dictionary, not in the context of resource references.	7.2.6	x	x
M7.11	The <b>x:Key</b> attribute of the <MatrixTransform> element MUST be present when the element is defined in a resource dictionary. It MUST NOT be specified outside a resource dictionary.	7.4.1	x	v

1 Table I-15. Common properties OPTIONAL conformance requirements

ID	Rule	Reference	Producer	Consumer
O7.1	Resource dictionaries MAY be specified in separate parts (called remote resource dictionaries) and referenced from within the <FixedPage.Resources> or <Canvas.Resources> property element.	7.2	x	
O7.2	A resource definition MAY reference another resource defined prior to the point of reference, including a resource previously within the same resource dictionary.	7.2.3	x	
O7.3	If the resource dictionary does not appear in a separate part, a resource definition MAY reference a previously defined resource in a resource dictionary of a parent or ancestor <Canvas> or <FixedPage> element.	7.2.3	x	
O7.5	The resource dictionary of a <Canvas> element MAY re-use (and thus override within the scope of the re-use) an <b>x:Key</b> value defined in the resource dictionary of a parent or ancestor <Canvas> or <FixedPage> element.	7.2.5	x	
O7.6	A resource definition MAY reference a previously defined resource with the same name that is defined in an ancestor resource dictionary.	7.2.5	x	

2 **I.8 Color**

3 Table I-16. Color MUST conformance requirements

ID	Rule	Reference	Producer	Consumer
M8.1	Consumers MUST support sRGB colors (8 bit-per-channel) in vector data, with and without alpha.	8.1		x
M8.2	Consumers MUST support sRGB colors in image data, using the JPEG, PNG, TIFF, or Windows Media Photo image formats.	8.1		x
M8.3	Consumers MUST support scRGB color specification in vector data, with and without alpha.	8.1		x
M8.4	Consumers MUST support scRGB colors in image data, using the Windows Media Photo image format.	8.1		x
M8.5	Consumers MUST support CMYK colors in vector data.	8.1		x
M8.6	Consumers MUST support CMYK colors in image data, using the TIFF or Windows Media Photo image formats.	8.1		x
M8.7	Consumers MUST support N-Channel colors in vector data.	8.1		x

I Conformance Requirements

M8.8	Consumers MUST support N-Channel colors in image data, using the Windows Media Photo image format.	8.1		x
M8.9	Consumers MUST support ICC Version 2 profiles for 3-, 4-, 5-, 6-, 7-, and 8-channel color.	8.1		x
M8.10	Consumers MUST support ICC Version 2 profiles with a Windows Color System (WCS) profile embedded as a private tag, either by handling or ignoring the WCS profile.	8.1		x
M8.11	Consumers MUST inspect the PageDeviceColorSpaceProfileURI PrintTicket setting to determine that this particular color specification is a native device color and MUST NOT be color-managed according to the included profile unless forced to do so for transparency effects.	8.1.7		x <sup>FP</sup>
M8.12	ICC profiles MUST conform to the requirements in the ICC Color Profile specification, Version 3.4.	8.1.8	x <sup>U</sup>	v
M8.13	All ICC profiles used in XPS Documents MUST be either an Input profile, an Output profile, a Monitor (RGB) profile, or a ColorSpace Conversion profile.	8.1.8	x <sup>U</sup>	v
M8.14	Real numbers specified for color channel values of scRGB and ContextColor colors MUST NOT use exponent forms of numbers.	8.1.11	x	v
M8.15	Although alpha values smaller than 0.0 and larger than 1.0 can be specified in scRGB images, the alpha values MUST be clamped to the valid range from 0.0 to 1.0 before any further processing.	8.1.13		x
M8.16	Although alpha values smaller than 0.0 and larger than 1.0 can be specified in CMYK images, the alpha values MUST be clamped to the valid range from 0.0 to 1.0 before any further processing.	8.1.14		x
M8.17	For N-Channel colors, the context color MUST specify a number of channel float values equal to the number of channels in the profile, setting unused ones to 0.0.	8.1.15	x	v
M8.18	Although alpha values smaller than 0.0 and larger than 1.0 can be specified in N-Channel images, the alpha values MUST be clamped to the valid range from 0.0 to 1.0 before any further processing.	8.1.15		x
M8.19	The profile for a named color MUST have 3, 4, 5, 6, 7 or 8 channels (and an <i>n</i> CLR signature, where <i>n</i> is the number of channels), mapping to a valid PCS.	8.1.16	x <sup>U</sup>	v
M8.20	Although alpha values smaller than 0.0 and larger than 1.0 can be specified in named color images, the alpha values MUST be clamped to the valid range from 0.0 to 1.0 before any further processing.	8.1.16		x
M8.21	Consumers that do not understand a named color MUST compute a color approximation through ICM-compliant color management functions using the specified profile.	8.1.16		x
M8.22	The color name specified by the DocumentImpositionColor	8.3		x <sup>FP</sup>



	PrintTicket setting MUST be matched only to profiles containing exactly one non-zero-length colorant name in the profile's colorantTable.		
M8.23	The color name specified by the DocumentImpositionColor setting serves as a label for that color only and MUST NOT be matched against any Named Colors known by the consumer.	8.3	x <sup>FP</sup>
M8.24	The comparison of the color name specified by the DocumentImpositionColor PrintTicket setting with the colorant name in the profile's colorantTable MUST be performed as a case-sensitive ASCII comparison after trimming leading and trailing whitespace from each string.	8.3	x <sup>FP</sup>
M8.25	For gradients, the specified blending color space in the PageBlendColorSpace PrintTicket setting is used only if no gradient stop color values are specified using sRGB or scRGB colors. If any of the gradient stop color values are specified using sRGB or scRGB colors or the consumer does not understand the PageBlendColorSpace PrintTicket setting, the color interpolation mode of the gradient brush MUST be used instead.	8.4	x <sup>FP</sup>
M8.27	If the PageDeviceColorSpaceUsage is set to MatchToDeviceDefault and the profile specified by the PageDeviceColorSpaceURI PrintTicket setting cannot be used as a device color space profile, elements using the profile MUST be color managed like any other element using a color profile.	8.5	x <sup>P</sup>
M8.28	If the PageBlendColorSpace PrintTicket setting is set to ICCProfile, the profile MUST be an output profile, otherwise it MUST be ignored.	8.5	x <sup>FP</sup>
M8.29	Elements using the named color identified by the DocumentImpositionColor PrintTicket setting MUST appear on all color separations.	8.5	x <sup>FP</sup>
M8.30	If an ICC profile is not embedded or associated with a raster image or if the embedded or associated profile is not compatible with the pixel format of the image, the default pixel formats for each color space MUST be treated as defined in §15.2.9, "Color Space Pixel Formats for Raster Images."	8.2.9	x
M8.31	Channel and tint float values in CMYK, N-Channel, and Named Color syntax MUST be clamped to the valid range from 0.0 to 1.0 before further processing. If the value is used as input for an ICC profile color transformation, it MUST subsequently be linearly scaled to the range from 0 to 255 or from 0 to 65535, depending on whether the profile uses 8-bit or 16-bit input tables.	8.1.14, 8.1.15, 8.1.16	x

1 **Notes:**

- 2 F — Only necessary if supporting the referenced feature.

I Conformance Requirements

- 1 P — Only applies to printing consumers.
- 2 U — Only required for a producer if the part is generated or for a consumer when
- 3 accessing the part.
- 4 *Table I-17. Color SHOULD conformance requirements*

ID	Rule	Reference	Producer	Consumer
S8.1	ICC profiles SHOULD be used when embedded in any image format with any color space. Images with integer pixel formats are assumed to have sRGB as the default color space and images with floating point pixel formats are assumed to have scRGB as the default color space; in these cases, an ICC profile is unnecessary.	8.1.8		x
S8.2	If consistency of appearance is important, instead of attaching or embedding a gray ICC profile, the producer SHOULD adjust the gray tone response curve of a grayscale image before adding it to the XPS Document.	8.1.8	x	
S8.3	A producer of XPS documents containing named colors SHOULD create the color profile in such a way that a linear ramp of the channel values corresponding to a named colorant maps to PCS values resulting in the same color appearance for consumers unaware of named colors (or the specific colorant).	8.1.16	x	
S8.4	The ContextColor syntax requires a minimum of 1 Alpha value and 3 Channel values for named colors. It is RECOMMENDED that a 1 or 2 tone profile uses the first 1 or 2 channels, respectively, and specifies 0 for the remaining channels.	8.1.16	x	
S8.5	If the consumer does not know ALL of the colorants named in the clr tag, it SHOULD treat the profile as if it were a regular N-channel source profile and SHOULD NOT attempt to use any of the known colorants, as that would result in undefined results.	8.1.16		x
S8.6	Support for JPEG CMYK images varies by implementation and SHOULD NOT be used in XPS Documents.	8.2.4.3	x	
S8.7	For consumers that do perform separation, the imposition named color is an indicator that the tint level supplied SHOULD be used for all device colorants.	8.3		x <sup>P</sup>
S8.8	Producers SHOULD create the profile used by the imposition color in such a way that it does not lay down excessive ink when printed on a device that does not perform separation.	8.3	x <sup>P</sup>	
S8.9	If a consumer understands the PageBlendColorSpace PrintTicket setting, it SHOULD convert all color to the specified blending color space before performing a blend operation.	8.4		x <sup>P</sup>

S8.10	If the PageDeviceColorSpaceUsage PrintTicket setting is set to MatchToDeviceDefault, the device's internal color profile SHOULD be used for color management of all elements not using the profile specified by the PageDeviceColorSpaceProfileURI PrintTicket setting.	8.5	x <sup>P</sup>
S8.11	If the PageDeviceColorSpaceUsage PrintTicket setting is set to OverrideDeviceDefault and the profile specified by the PageDeviceColorSpaceProfileURI PrintTicket setting has a number of channels matching the number of primaries of the device, it SHOULD be used instead of the device's internal color management for all elements.	8.5	x <sup>P</sup>
S8.12	If the PageBlendColorSpace PrintTicket setting is set to ICCProfile, the Uri property of the option specifies an ICC profile defining the color space that SHOULD be used for blending.	8.5	x <sup>P</sup>
S8.13	The PageICMRenderingIntent PrintTicket setting SHOULD be ignored for elements using a profile that specifies the rendering intent in the profile.	8.5	x <sup>P</sup>
S8.14	A consumer incapable of supporting named colors SHOULD treat the colorant table for named colors tag in an ICC profile as a user-defined custom tag, and therefore ignore it. The consumer SHOULD instead use the color tables as provided in the profile to convert the specified colors to the Profile Connection Space (PCS).	8.1.8	x

1 **Notes:**

2 P — Only applies to printing consumers.

3 *Table I-18. Color OPTIONAL conformance requirements*

ID	Rule	Reference	Producer	Consumer
O8.1	Consumers are not required to handle all color spaces natively, but rather MAY convert data specified in a rich color space to sRGB at an early stage (possibly resulting in reduced fidelity).	8.1		x
O8.2	An ICC profile MAY contain the private tag, "MS00", which specifies an embedded Windows Color System (WCS) profile.	8.1.8	x	x
O8.3	When a named color is used in a gradient brush or with transparency, the result produced by consumers that are not named-color aware MAY differ significantly from the result produced by consumers that are named-color aware.	8.1.16		x
O8.4	A named color profile MAY be used with images for spot coloring.	8.1.16	x	x
O8.5	Consumers MAY perform color separation, if desired.	8.3		x
O8.6	Consumers MAY support rich color spaces such as scRGB or CMYK. Consumers that encounter any document using non-sRGB colors MAY process those colors using the simpler sRGB	8.4		x

I Conformance Requirements

	color space, resulting in deviations, especially for alpha blending		
O8.7	If the PageColorManagement PrintTicket setting specifies a value of Driver, the driver MAY color manage elements or convert them to different color spaces.	8.5	x <sup>P</sup>
O8.8	Elements using the profile specified by PageBlendColorSpace PrintTicket setting with a value of ICCProfile MAY be blended naively (channel-by-channel) without converting through PCS.	8.5	x <sup>P</sup>

1 **Notes:**

2 P — Only applies to printing consumers.

3 **I.9 Document Structure and Interactivity**

4 *Table I-19. Document structure MUST conformance requirements*

ID	Rule	Reference	Producer	Consumer
M9.1	In order to merge the table cells and rows correctly, producers MUST specify empty <TableCellStructure> elements for cells that do not break across story fragments.	9.1.2	x <sup>D</sup>	
M9.2	If hyperlinked <Path> or <Glyphs> elements are rendered as overlapping on the page, consumers MUST treat the topmost element as the only hyperlink that may be activated in the overlapping region.	9.2.1		x <sup>H</sup>
M9.3	If a producer specifies a <b>FixedPage.NavigateUri</b> attribute on a <Canvas> element, consumers MUST treat all child elements of that canvas that do not override this value with their own <b>FixedPage.NavigateUri</b> attribute setting as having an associated hyperlink.	9.2.1		x <sup>H</sup>
M9.4	Relative internal hyperlinks between FixedPage parts MUST specify, at a minimum, the named address relative to the FixedDocument part.	9.2.1	x <sup>H</sup>	v <sup>H</sup>
M9.5	In order to be addressable by either a hyperlink or the document outline, the named address MUST appear in the <PageContent.LinkTargets> element in the fixed document.	9.2.1	x <sup>H</sup>	
M9.6	If a named address appears in the <PageContent.LinkTargets> element in the fixed document but is not found in the <b>Name</b> attribute of an element within the associated fixed page, consumers MUST treat the top of the associated fixed page as the named address.	9.2.1		x <sup>H</sup>
M9.7	If a named address in a URI fragment is not found, consumers MUST ignore the fragment portion of the URI.	9.2.1		x <sup>H</sup>
M9.8	Internal references MUST specify a page address relative to the fixed document sequence.	9.2.2	x <sup>H</sup>	v <sup>H</sup>

M9.9	Consumers MUST expose every element of the fixed page markup to an accessibility interface in the determined reading order, even if the elements are not referenced in the content structure markup.	9.4.1	x	
M9.10	The <b>Name</b> attribute MUST NOT be specified on any children of a <ResourceDictionary> element.	9.2.3	x	v

1 **Notes:**

2 D — Only applies to producers that generate document structure or consumers that  
3 use it.

4 H — Only applies to producers that generate hyperlinks or consumers that  
5 implement hyperlinks.

6 *Table I-20. Document structure SHOULD conformance requirements*

ID	Rule	Reference	Producer	Consumer
S9.1	Every meaningful element in the fixed page markup SHOULD specify a <b>Name</b> attribute in order for the document structure markup to refer to it	9.1.1	x	
S9.3	Document structure markup SHOULD NOT refer to a single named element more than once in the document content or to a named element that embeds another named element that it also refers to. When referring to a <Canvas> element, producers SHOULD consider all descendant elements to be referenced in markup order.	9.1.1	x <sup>D</sup>	
S9.4	If a <StoryBreak> element is not present at the beginning of the content structure markup, consumers SHOULD consider the markup a continuation of the previous story fragment that must be merged. Likewise, if a <StoryBreak> element is not present at the end of the content structure markup, consumers SHOULD consider the markup a continuation to the next story fragment that must be merged to determine the cross-fragment content structure.	9.1.2		x <sup>D</sup>
S9.5	Producers authoring document structure information SHOULD reference every element of the fixed page markup that has semantic meaning (such as text or images) in the StoryFragments parts.	9.1.2.2	x <sup>D</sup>	
S9.6	If consumers enable user interactivity, they SHOULD support hyperlink activation and addressing.	9.2		x <sup>H</sup>
S9.7	When activating a hyperlink, consumers SHOULD load the specified resource if they understand the URI type. If the URI is an internal reference to the XPS Document, consumers SHOULD navigate to the URI.	9.2.1		x <sup>H</sup>
S9.8	The value of the <b>Name</b> attribute on a <FixedPage>,	9.2.1	x	v

I Conformance Requirements

	<Canvas>, <Path>, or <Glyphs> element SHOULD be unique within the scope of the fixed document.			
S9.9	It is RECOMMENDED that <b>Name</b> attribute values on <FixedPage>, <Canvas>, <Path>, and <Glyphs> elements be unique within an entire fixed document sequence.	9.2.1	x	v
S9.10	If the <b>Name</b> attribute is specified, producers SHOULD also create a corresponding <LinkTarget> element in the FixedDocument part within the <PageContent> element that links to the parent fixed page	9.2.3	x	
S9.11	A hyperlink destination in the same fixed document SHOULD be expressed as a relative URI.	9.2.4	x <sup>H</sup>	
S9.12	If selection is supported, consumers SHOULD provide a visual cue over or around selected elements.	9.3	x <sup>S</sup>	
S9.13	Selection order within an XPS Document SHOULD follow reading order.	9.3	x <sup>S</sup>	
S9.14	In the absence of document structure provided in the XPS Document, consumers SHOULD, at minimum, rely on the markup order to determine reading order.	9.4.1	x	
S9.15	Producers SHOULD order the markup in FixedPage parts to reflect the order in which it is intended to be read.	9.4.1	x	
S9.16	When document structure information is present, consumers SHOULD rely on the order of appearance of named elements in the content structure markup to determine reading order.	9.4.1	x	
S9.17	The RECOMMENDED reading order of a page-centric application is 1) order the content by page, 2) order by story fragment within the page based on the order the <StoryFragment> elements are specified in the StoryFragments part for that page, 3) order by <NamedElement> reference within the <StoryFragment> element, 4) append all un-referenced elements that appear in the fixed page markup, ordered by markup order.	9.4.1	x	
S9.18	Producers SHOULD order <StoryFragment> elements in each StoryFragments part in their intended reading order.	9.4.1	x <sup>D</sup>	
S9.19	The RECOMMENDED reading order of a story-centric application is as follows: 1) Order content by story in the sequence the <Story> elements appear in the DocumentStructure part. 2) Within a story, order <StoryFragmentReference> elements in the sequence they appear in the DocumentStructure part. 3) Within a story fragment, order by <NamedElement> references in the StoryFragments part markup. 4) Append all un-referenced elements that appear in the fixed page markup, ordered by page number, then markup order	9.4.1	x <sup>D</sup>	
S9.20	Producers SHOULD order <Story> elements in the DocumentStructure part in their intended reading order.	9.4.1	x <sup>D</sup>	
S9.21	Producers SHOULD order <StoryFragmentReference>	9.4.1	x <sup>D</sup>	

	elements within a <Story> element in their intended reading order.		
S9.22	A screen reader consumer SHOULD read the document according to its reading order.	9.4.2	×
S9.23	A screen reader SHOULD use the <b>UnicodeString</b> attribute of each <Glyphs> element to determine the text to read.	9.4.2	×
S9.24	If a screen reader provides features to navigate the document by structural elements, such as paragraphs or table rows, it SHOULD use any document structure information included in the XPS Document.	9.4.2	× <sup>D</sup>
S9.25	If the screen reader provides features to describe images, it SHOULD read the text provided in the <b>AutomationProperties.Name</b> and <b>AutomationProperties.HelpText</b> attributes.	9.4.2	×
S9.26	If the screen reader provides features to describe hyperlink addresses, it SHOULD read the text provided in the <b>FixedPage.NavigateUri</b> attribute.	9.4.2	×
S9.27	Images and graphics SHOULD specify text alternatives for images and graphics to make this content accessible to vision-impaired individuals. The <b>AutomationProperties.Name</b> attribute SHOULD contain a short description of the basic contents of the image or vector graphic. Individual <Path> elements that do not provide any semantic meaning (such as a line between sections or outlining a table) SHOULD NOT specify these text alternative attributes.	9.4.3	×
S9.28	An image SHOULD specify the <b>AutomationProperties.Name</b> and <b>AutomationProperties.HelpText</b> attributes on the <Path> element that is filled with an <ImageBrush> that describes the content specified by the ImageSource attribute of the <ImageBrush> element.	9.4.3	×
S9.29	A vector graphic (a collection of one or more <Path> elements representing a single drawing) SHOULD specify the <b>AutomationProperties.Name</b> and <b>AutomationProperties.HelpText</b> attributes only once, directly on a <Canvas> element wrapping the <Path> elements comprising the graphic.	9.4.3	×
S9.30	Children of <VisualBrush> elements SHOULD NOT be referenced by document structure markup.	9.1.2.13	× <sup>D</sup>

1 **Notes:**

- 2 D — Only applies to producers that generate document structure or consumers that  
3 use it.
- 4 H — Only applies to producers that generate hyperlinks or consumers that  
5 implement hyperlinks.
- 6 S — Only applies to consumers that implement selection.

## I Conformance Requirements

1 Table I-21. Document structure OPTIONAL conformance requirements

ID	Rule	Reference	Producer	Consumer
O9.1	Producers MAY choose to add document structure information to XPS Documents. Consumers MAY ignore any authored document structure or hyperlinks.	Clause 9	x <sup>D</sup>	x <sup>D</sup>
O9.2	Producers MAY provide either the document outline or the document content, or both; consumers MAY ignore either or both.	9.1	x <sup>D</sup>	
O9.3	Consumers MAY choose to interpret document structure markup that refers to a single named element more than once, or refers to a named element that embeds another named element that is also referenced, as duplicate content.	9.1.1		x <sup>D</sup>
O9.4	Consumers MAY first attempt to locate named elements for document structure directly from the FixedDocument part markup, where they might appear as <LinkTarget> elements if that named element is also intended as an addressable location.	9.1.1		x <sup>D</sup>
O9.5	A <TableStructure> element is the complete definition of a table. An implementation MAY use it to build special functionality, such as row or column selection.	9.1.2.6		x <sup>D</sup>
O9.6	Internal hyperlinks can specify a named element fragment relative to a particular fixed document, but consumers MAY interpret such a URI relative to the entire fixed document sequence instead	9.2.1		x <sup>H</sup>
O9.7	Consumers MAY ignore the <b>Name</b> attribute.	9.2.3		x
O9.8	Consumers MAY ignore the <b>FixedPage.NavigateUri</b> attribute.	9.2.4		x
O9.9	Viewing consumers that support interactivity MAY support selection and copying.	9.3		x <sup>S</sup>
O9.10	Consumers MAY use the <b>FragmentType</b> attribute of the <StoryFragment> element to determine selection behavior, such as disallowing selection of both the page header and the page contents while allowing independent selection within those stories.	9.3		x <sup>DS</sup>
O9.11	In the absence of document structure information provided in the XPS Document, consumers MAY infer the reading order from the position of elements on the page.	9.4.1		x
O9.12	Consumers MAY use the <b>FragmentType</b> attribute of the <StoryFragment> element to determine reading order by interpreting elements that have <b>FragmentType</b> values of Header and Footer as belonging first or last in the reading order, respectively.	9.4.1		x <sup>D</sup>



09.13	Screen readers MAY inspect the <b>Indices</b> attribute to resolve potential ambiguities in the <b>UnicodeString</b> attribute.	9.4.2	x
-------	---	-------	---

1 **Notes:**

2 D — Only applies to producers that generate document structure or consumers that  
3 use it.

4 H — Only applies to producers that generate hyperlinks or consumers that  
5 implement hyperlinks.

6 S — Only applies to consumers that implement selection.

7 **I.10 XPS Document Package Features**

8 *Table I-22. XPS Document package feature MUST conformance requirements*

ID	Rule	Reference	Producer	Consumer
M10.1	Consumers MUST be prepared to correctly process interleaved packages in which the PrintTicket or the portion of the relationship data attaching the PrintTicket appears in the package after the affected part.	10.1	x <sup>P</sup>	
M10.2	Consumers MUST be able to consume packages regardless of their interleaving structure.	10.1.3	x	
M10.3	Consumers that lack the resources to process a part MUST indicate an error condition.	10.1.3	x	
M10.4	When consuming interleaved packages, consumers MUST NOT discard any parts without instruction from a DiscardControl part unless they have the ability to access the parts again.	10.1.3	x	
M10.5	If a consumer encounters a reference to an unknown part, it MUST continue to receive further bytes of the package until the unknown part has been transmitted <i>or</i> until the end of the package is reached (indicating an error condition).	10.1.3	x	
M10.6	The DiscardControl part MUST NOT reference itself.	10.1.4.1	x	v
M10.7	If either the <b>Target</b> attribute or the <b>SentinelPage</b> attribute of the <Discard> element contain an invalid reference (refer outside the package), the <Discard> element MUST be ignored.	10.1.4.1.2	x	
M10.8	All producers and consumers signing and verifying signatures for end users or applications MUST adhere to the XPS Document signature policy, and producers and consumers MUST interpret digital signatures consistently.	10.2.1	x <sup>A</sup>	x <sup>A</sup>
M10.9	Consumers MUST NOT disable functionality based on	10.2.1		x <sup>A</sup>

I Conformance Requirements

	whether or not an action will invalidate a signature.		
M10.10	An XPS Document MUST be considered signed according to the XPS Document signing policy, regardless of the validity of that signature, if the signing rules described in §10.2.1.1, "Signing Rules" are observed.	10.2.1.1	x <sup>A</sup>
M10.11	An XPS Document MUST NOT be considered signed according to the XPS Document signing policy if any part not covered by the signing rules is included in the signature or if any relationship not covered by the signing rules is included in the signature.	10.2.1.1	x <sup>A</sup>
M10.12	An XPS Document digital signer MUST NOT sign an XPS Document that contains content (parts or relationships parts) to be signed that defines the Markup Compatibility namespace but the signer does not fully understand all elements, attributes, and alternate content representations introduced through the markup compatibility mechanisms.	10.2.1.1	x <sup>A</sup>
M10.13	An XPS Document digital signature MUST be shown as an in-compliant digital signature if it violates any of the signing rules regarding parts or relationships that MUST or MUST NOT be signed.	10.2.1.2	x <sup>A</sup>
M10.14	An XPS Document digital signature MUST be shown as a broken digital signature if it is not an in-compliant digital signature, but the signature fails the signature validation routines described in the Open Packaging Conventions.	10.2.1.2	x <sup>A</sup>
M10.15	An XPS Document digital signature MUST be shown as a questionable digital signature if it is not an in-compliant or broken digital signature, but the certificate cannot be authenticated against the certificate authority or the signed content (parts and relationships) contain elements or attributes from an unknown namespace introduced through Markup Compatibility mechanisms.	10.2.1.2	x <sup>A</sup>
M10.16	An XPS Document digital signature MUST be shown as a valid digital signature if it is not an in-compliant, broken, or questionable digital signature.	10.2.1.2	x <sup>A</sup>
M10.17	To prohibit additional signatures in an XPS Document, the signing application MUST sign all the Digital Signature Origin part's relationships of relationship type Digital Signature with the same signature as the rest of the content.	10.2.1.3	x <sup>A</sup>
M10.18	XPS Document signatures MUST NOT refer to a remote certificate store. All certificates MUST be stored in the XPS Document either as a Certificate part or in the Digital Signature XML Signature part.	10.2.1.4	x <sup>A</sup> v <sup>A</sup>
M10.19	To link a <SignatureDefinition> to a signature, the value of the <b>SpotID</b> MUST be specified in the <b>Id</b> attribute of the corresponding <Signature> element in	10.2.2.2.1	x <sup>A</sup>

	the Digital Signature XML Signature part.			
M10.20	Due to space and rendering limitations, producers MUST NOT assume that consumers will use the values specified in the <SpotLocation> element.	10.2.2.3	x <sup>A</sup>	
M10.21	Consumers MUST display the full value of the <Intent> element to the signing party, either in the signature spot or through some other mechanism.	10.2.2.4		x <sup>A</sup>
M10.22	If specified, the <SignBy> date and time MUST be specified as a complete date plus hours, minutes, and seconds in UTC time, as described in the W3C Note "Date and Time Formats."	10.2.2.5	x <sup>A</sup>	
M10.23	There MUST NOT be more than one DiscardControl package relationship.	10.1.4.1	x <sup>R</sup>	v <sup>R</sup>
M10.24	In some cases, producers might rewrite the contents of a package so that parts are provided more than once, allowing consumers to discard a part in order to free resources for additional processing. Each instance of a part MUST be stored as a new, uniquely named part in the package.	10.1.4.1	x <sup>R</sup>	v <sup>R</sup>

1 **Notes:**

2 A — Only applies to producers or consumers implementing digital signature features.

3 P — Only applies to printing consumers.

4 R — Only applies to producers or consumers that use DiscardControl parts.

5 *Table I-23. XPS Document package feature SHOULD conformance requirements*

ID	Rule	Reference	Producer	Consumer
S10.1	When interleaving, the Content Types stream SHOULD be interleaved according to the recommendations in the OPC specification.	10.1	x	
S10.2	When interleaving, PrintTicket parts SHOULD be written to the package before the part to which they are attached.	10.1	x	
S10.3	When interleaving, the portion of the relationship data attaching the PrintTicket to a part SHOULD be written to the package before the part to which it is attached or in close proximity to the part to which it is attached.	10.1	x	
S10.4	When interleaving, if no PrintTicket settings are specified for a FixedDocumentSequence, FixedDocument, or FixedPage part, an empty PrintTicket part SHOULD be attached to the part, and the portion of the relationship data attaching the empty PrintTicket SHOULD be written to the package before the part to which it is attached or in close proximity to the part to	10.1	x	

I Conformance Requirements

	which it is attached.		
S10.5	When interleaving, the last piece of the Relationships part for a FixedPage part SHOULD be written to the package in close proximity to the first piece of the FixedPage part.	10.1	x
S10.6	It is RECOMMENDED that one empty PrintTicket be shared for all parts that attach an empty PrintTicket.	10.1.1	x
S10.7	Producers, such as drivers, that target resource-constrained consumers SHOULD: 1) Conservatively model the memory usage of the device. 2) Interleave pieces of parts in the correct order. 3) Decide when certain parts can be discarded by the consumer and inform the consumer within the package stream. 4) Add to the package a uniquely named copy of a resource that could have been discarded, if the resource is referenced by a part sent later in the stream.	10.1.4	x
S10.8	DiscardControl parts that are not well-formed SHOULD NOT be processed and an error SHOULD NOT be reported.	10.1.4.1	x <sup>U</sup>
S10.9	If a <Discard> element is encountered where either or both of the <b>Target</b> attribute and <b>SentinelPage</b> attribute identify a part which has not been processed yet (is still unknown), the <Discard> element SHOULD be retained until both parts identified by the <b>Target</b> attribute and <b>SentinelPage</b> attribute have been processed or until the end of the package is reached.	10.1.4.1.2	x <sup>U</sup>
S10.10	When adding a digital signature to an interleaved package, producers of digitally signed documents that are intended for streaming consumption SHOULD add all digital signature parts and the package relationship to the digital signature parts at the beginning of the package, before adding any other part.	10.1.5	x <sup>A</sup>
S10.11	Consumers SHOULD inform the end user if an action they are going to take will invalidate an existing signature.	10.2.1	x <sup>A</sup>
S10.12	When printing signed documents, the PrintTicket setting JobDigitalSignatureProcessing SHOULD be used to control the digital signature processing behavior. Consumers SHOULD process this PrintTicket setting, if present	10.2.1.5	x <sup>AP</sup>
S10.13	If the location specified by the <SpotLocation> element is not used when the signature spot is displayed, it is RECOMMENDED that consumers choose a location that does not contain any page content.	10.2.2.3	x <sup>A</sup>
S10.14	It is RECOMMENDED that consumers render signature spots as consistently sized rectangles that include the signer name, the intent, the signing location, and the scope of the XPS Document to be signed.	10.2.2.3	x <sup>A</sup>

S10.15	It is RECOMMENDED that a signature spot be a clickable area used to launch the digital signing process.	10.2.2.3	x <sup>A</sup>
S10.16	If the <SignBy> element is specified, the consumer SHOULD NOT allow the signing party to sign the document using this particular signature spot after the date and time specified.	10.2.2.5	x <sup>A</sup>
S10.17	The values specified in the Core Properties part SHOULD refer to the entire fixed payload, including the root FixedDocumentSequence part and the compilation of all FixedDocument parts it references.	10.3	x <sup>C</sup>
S10.18	Head-first XPS Document consumers SHOULD attempt to detect inconsistent packages as soon as possible and SHOULD generate an error message, even if they have already processed the pages that resulted in the error.	10.1	x <sup>K</sup>
S10.19	The viewing consumer SHOULD use the values specified in the <SpotLocation> element to place a signature spot.	10.2.2.3	x <sup>F</sup>

1 **Notes:**

- 2 A — Only applies to producers or consumers implementing digital signature features.
- 3 C — Only applies if Core Properties are used.
- 4 F — Only necessary if supporting the referenced feature.
- 5 K — Only applies to consumers that consume the XPS Document package head-first.
- 6 P — Only applies to printing consumers.
- 7 U — Only required for a producer if the part is generated or for a consumer when
- 8 accessing the part.

9 *Table I-24. XPS Document package feature OPTIONAL conformance requirements*

ID	Rule	Reference		
			Producer	Consumer
O10.1	Interleaving is OPTIONAL.	10.1	x	
O10.2	Producers MAY optimize the interleaving order of parts to help consumers avoid stalls during read-time streaming, and to allow consumers to manage their memory resources more efficiently.	10.1.2	x	
O10.3	Consumers MAY discard FixedPage parts once they have been processed.	10.1.3		x
O10.4	Consumers MAY discard FixedDocument and FixedDocumentSequence parts after all their child elements and their closing tags have been processed.	10.1.3		x
O10.5	In the absence of explicit directives to the contrary, consumers MAY discard parts as directed by the	10.1.3		x

## I Conformance Requirements

DiscardControl part.			
O10.6	Some producers (typically drivers) MAY choose a suitable interleaving order by modeling the resource management behavior of the consumer.	10.1.4	x
O10.7	A consumer MAY decide to ignore a malformed DiscardControl part in its entirety or from the first malformed node onward.	10.1.4.1	x
O10.8	An XPS Document digital signer MAY choose not to sign any content (parts or relationships parts) that defines the Markup Compatibility namespace, even if the content is fully understood.	10.2.1.1	x <sup>A</sup>
O10.9	An XPS Document digital signature MAY be shown as a questionable digital signature if it is not an incompliant or broken digital signature, but contains some other detectable problem at the discretion of the consumer.	10.2.1.2	x <sup>A</sup>
O10.10	XPS Documents MAY be signed more than once.	10.2.1.3	x <sup>A</sup>
O10.11	Producers MAY include the JobDigitalSignatureProcessing setting in the job-level PrintTicket within the XPS Document content.	10.2.1.5	x <sup>A</sup>
O10.12	The SpotID attribute of the <SignatureDefinition> element MAY be used to link to an existing signature.	10.2.2.2.1	x <sup>A</sup>
O10.13	Consumers MAY choose a size and shape to display a signature spot based on the desired display information and page content.	10.2.2.3	x <sup>A</sup>
O10.14	The <SigningLocation> element MAY be set by the original producer of the XPS Document or by the signing party at the time of signing.	10.2.2.6	x <sup>A</sup>

### 1 Notes:

- 2 A — Only applies to producers or consumers implementing digital signature features.

## 3 I.11 Rendering Rules

4 Table I-25. Rendering rules MUST conformance requirements

ID	Rule	Reference	Producer	Consumer
M11.1	Producers MUST generate XPS Documents that can be accurately rendered by following the rules described in the "Rendering Rules" clause. Consumers MUST adhere to the rules described in the "Rendering Rules" clause when rendering XPS Documents	Clause 11	x	x
M11.2	If a non-invertible transform is encountered during rendering, consumers MUST omit rendering the affected	11.1.3		x

	element and all of its child and descendant elements.		
M11.3	If a non-invertible transform is encountered on a geometry (as specified directly on the geometry or through concatenation), the geometry MUST be considered to contain no area.	11.1.3	×
M11.4	Producers MUST NOT assume a specific placement error for curve decomposition or rely on side-effects of a specific consumer implementation.	11.1.5	×
M11.5	Encountering markup with characteristics outside of the consumer-specific implementation limits MUST cause an error condition.	11.2	×
M11.6	The alpha information in TIFF images using an ExtraSamples tag value of 1 and in Windows Media Photo images using pixel formats WICPixelFormat32bppPBGRA, WICPixelFormat64bppPRGBA or WICPixelFormat128bppPRGBAFloat MUST be interpreted as pre-multiplied alpha information.	11.4.1	×
M11.7	Composition MUST have the same effect as the application of the rules in §18.5, "Composition Rules," in sequence.	11.5	×
M11.8	The precise source coordinates as specified by the viewbox MUST be used to place an up-sampled image tile, which is equivalent to using fractional pixels of the original source image.	11.7.2	×
M11.9	Consumers MUST precisely position the tiles specified by the image brush and visual brush. If the specified values result in fractional device pixels, the consumer MUST calculate a running placement-error delta and adjust the placement of the next tile where the delta reaches a full device pixel in order to keep the tiles from being increasingly out of phase as the expanse of the path is filled.	11.7.3	×

1 Table I-26. Rendering rules SHOULD conformance requirements

ID	Rule	Reference	Producer	Consumer
S11.1	All computations on coordinate values SHOULD be performed with at least single floating-point precision. Final conversion (after all transforms have been computed) to device coordinates SHOULD retain at least as much fractional precision as a 28.4 fixed-point representation before performing pixel coverage calculations.	11.1.2	×	×
S11.2	An <i>ideal</i> consumer implementation SHOULD render pixels in an 8x8 sub-pixel space, perform an 8x8 box filter sampling, and set the pixel to the resulting color value.	11.1.4		×
S11.3	When rendering a shape, a <i>practical</i> implementation (such as a bi-tonal printing device) SHOULD turn on each pixel	11.1.4		×

I Conformance Requirements

	whose center (at $x+0.5$ ) is covered by the shape, or is touched by the shape with the shape extending beyond the pixel center in the positive $x$ or $y$ direction of the device.		
S11.4	When rendering geometries, consumers SHOULD render curves so they appear smooth from a normal viewing distance.	11.1.5	×
S11.5	When no anti-aliasing is used, abutting shapes that share the same device coordinates for the end-points and control-points of an edge SHOULD be rendered without overlap and without gaps. Ideally, an implementation SHOULD also follow this rule for shapes that are mathematically abutting without sharing device coordinates for end-points and control-points of edges.	11.1.7	×
S11.6	Clipping occurs as if a mask were created from the clip geometry according to the pixel inclusion rules. An ideal consumer SHOULD create such a mask in an 8x8 sub-pixel space and subsequently draw only those sub-pixels of a shape that correspond to "ON" sub-pixels in the mask.	11.1.8	×
S11.7	A practical implementation (such as a bi-tonal printing device) SHOULD create a pixel mask according to the pixel inclusion rules and subsequently draw only those pixels of a shape that correspond to "ON" pixels in the mask. In creating the mask and drawing the shape, the abutment of shapes rule SHOULD be observed so that no pixel of the shape is drawn that would not have been drawn if the clip geometry were another abutting shape.	11.1.8	×
S11.8	A typical consumer SHOULD be able to process markup with the implementation limit characteristics indicated in Table 18-1. Producers SHOULD produce only XPS Documents that stay within these implementation limits.	11.2	×
S11.9	Coordinates are real numbers and SHOULD be computed with at least single floating point precision.	11.2	×
S11.10	If the nesting level of <VisualBrush> elements is higher than 16, a consumer SHOULD attempt to flatten the nested content to a bitmap representation rather than failing to draw.	11.2	×
S11.11	Gradients SHOULD be rendered according to the guidelines described in §18.3, "Gradient Computations."	11.3	×
S11.12	Consumers SHOULD pre-process gradient stops for all gradients using the steps described in §11.3.1.1, "Gradient Stop Pre-Processing."	11.3.1.1	×
S11.13	If any gradient stops use an sRGB or scRGB color specification or the consumer does not understand the PageBlendColorSpace PrintTicket setting, consumers SHOULD blend colors between gradient stops in the color space indicated by the <b>ColorInterpolationMode</b> attribute of the gradient brush. If none of the gradient stop	11.3.1.2	×



	elements uses an sRGB or scRGB color specification and the consumer understands the PageBlendColorSpace PrintTicket setting, the PageBlendColorSpace PrintTicket setting SHOULD be used. This SHOULD be a linear, channel-by-channel blend operation.			
S11.14	If a <b>ColorInterpolationMode</b> value of SRgbLinearInterpolation is used, the BLEND() function SHOULD convert the color values to sRGB first, and then perform a linear interpolation between them.	11.3.1.2		×
S11.15	If a <b>ColorInterpolationMode</b> value of ScRgbLinearInterpolation is used, the BLEND() function SHOULD convert the color values to scRGB first, and then perform a linear interpolation between them.	11.3.1.2		×
S11.16	In the presence of transformations or when individual gradient stops are very close, the local color gradient at the offset used in the BLEND() function may be large, resulting in a large change over the extent of a single device pixel. In this case, it is RECOMMENDED that the BLEND() function interpolate the gradient over the extent of each device pixel. Producers SHOULD NOT, however, rely on a specific effect for such dense gradient specifications.	11.3.1.2	×	×
S11.17	Producers SHOULD either avoid very close gradient stops to the gradient end point when specifying radial gradients where the outside area is visible or avoid specifying radial gradients with a gradient origin on or outside the ellipse (in which case there is no outside area) to ensure consistent rendering results.	11.3.1.2		×
S11.18	All opacity calculations SHOULD be performed with at least 8-bit precision to provide sufficient quality for nested content.	11.4		×
S11.19	When composing superluminous colors, management of out-of-gamut colors SHOULD be deferred until the result is rendered to the final target, at which point out-of-gamut colors are clipped or color managed.	11.4.1		×
S11.20	The color and appearance of the surface created to hold drawing content as it is composed SHOULD match the destination color and appearance, typically a solid white background for a fixed page or transparent for a canvas.	11.5		×
S11.21	Contours and dashes SHOULD be rendered so that they have the same appearance as if rendered by sweeping the complete length of the contour or dash with a line segment that is perpendicular to the contour and extends with half its length to each side of the contour. All points covered by the sweep of this perpendicular line are part of the dash or contour.	11.6		×
S11.22	Consumers SHOULD ensure that parallel edges of strokes appear parallel.	11.6.1		×
S11.23	Consumers SHOULD produce a visually consistent	11.6.2		×

I Conformance Requirements

	appearance of stroke thickness for thin lines, regardless of their orientation or how they fit on the device pixel grid.			
S11.24	Consumers SHOULD select line and curve drawing algorithms that behave symmetrically and result in the same set of device pixels being drawn regardless of the direction of the line or curve (start point and end point exchanged).	11.6.3		×
S11.25	If the current render transform is an invertible matrix, consumers SHOULD perform computations on poly line segments and poly Bézier segments with sufficient accuracy to avoid producing zero-length segments.	11.6.8		×
S11.26	If both width and height of a tile are nearly zero, implementations SHOULD average the color values of the brush contents, resulting in a constant-color brush.	11.7.1		×
S11.27	Producers SHOULD avoid producing extreme cases where either the height, width, or both height and width are nearly zero and SHOULD NOT rely on any specific behavior when they do	11.7.1	×	
S11.28	Source sampling SHOULD be done from the center of the pixel and should be mapped to the center of the pixel in the device-space. With one extent of the viewbox zero, sampling SHOULD be done along a line parallel to the non-zero side. With both extents of the viewbox zero, a point sample SHOULD be taken.	11.7.2		×
S11.29	When up-sampling an image presented at a lower resolution than the device resolution, bilinear filtering SHOULD be used.	11.7.2		×
S11.30	When down-sampling an image presented at a higher resolution than the device resolution, at least a bilinear filter SHOULD be used.	11.7.2		×
S11.31	A stroke using the consistent nominal stroke width convention SHOULD be rendered with a width consistent with other strokes using the convention that have the same <b>StrokeThickness</b> attribute value, and consumers aware of this convention SHOULD render such a stroke no thinner than the thinnest visible line that consumer supports without dropouts.			× <sup>F</sup>

1 F — Only necessary if supporting the referenced feature.

2 *Table I-27. Rendering rules OPTIONAL conformance requirements*

ID	Rule	Reference	Producer	Consumer
O11.1	Very high resolution devices MAY use lower fractional precision than a 28.4 fixed-point representation to represent device coordinates.	11.1.2		×

O11.2	Consumers MAY use different rendering logic as long as it closely approximates the logic of rendering pixels in an 8x8 sub-pixel space, performing an 8x8 box filter sampling, and setting the pixel to the resulting color value.	11.1.4	×
O11.3	Devices MAY use sub-pixel masking.	11.1.4	×
O11.4	An implementation capable of anti-aliasing MAY draw a thin line in a way that blends with the background to varying degrees.	11.1.4	×
O11.5	A bi-tonal implementation on a printer MAY draw thin lines with or without drop-outs, or by applying half-toning, depending on the desired output quality.	11.1.4	×
O11.6	Consumers MAY apply pixel placement rules optimized for character rendering to individual glyphs in a <Glyphs> element.	11.1.6	×
O11.7	Behavior of blending with very close gradient stops MAY vary in an implementation-specific manner (see S11.16).	11.3.1.2	×
O11.8	When a radial gradient origin is on or outside the ellipse, the "outside" area (outside the cone defined by the origin and the ellipse) MAY be filled with an interpolated color value, depending on the resolution.	11.3.1.2	×
O11.9	In certain scenarios (such as when rendering 3D scenes to a bitmap), producers MAY choose to create pre-multiplied bitmap data specifying "superluminous" colors.	11.4.1	×
O11.10	Consumers MAY handle superluminous colors natively or MAY instead choose to convert pre-multiplied source data containing superluminous colors to non-pre-multiplied data before composition by ignoring the superluminous portion of each color channel value.	11.4.1	×
O11.11	A consumer MAY choose always to initialize the alpha channel of the surface created to hold the drawing content as it is composed to 0.0 (transparent) and the color value to black.	11.5	×
O11.12	When doing page composition, if all elements on a canvas and the canvas itself are opaque (an opacity of 1.0) and parent or ancestor <Canvas> elements are also opaque, the elements MAY be drawn directly to the containing fixed page (or canvas), provided all render transform and clip values are observed	11.5.1	×
O11.13	When doing page composition, if an element is fully transparent (an opacity of 0.0), it MAY be skipped.	11.5.1	×
O11.14	When doing page composition, if a canvas has an opacity of 0.0, it and all of its child and descendant elements MAY be skipped.	11.5.1	×
O11.15	When doing page composition, if a canvas has a Clip property with no contained area, the canvas and all of its child and descendant elements MAY be skipped.	11.5.1	×

I Conformance Requirements

O11.16	When doing page composition, a consumer MAY further restrict the size of the temporary surface it creates by the effective extent of the geometry specified by the Clip property of the canvas.	11.5.1	x
O11.17	When doing page composition, a consumer MAY use methods to achieve transparency other than creating a temporary surface. Such methods MAY include planar mapping.	11.5.1	x
O11.21	If only one of the width and height values of a tile is nearly zero, the brush should be constant-colored along lines parallel to the narrow side of the viewport, but implementations MAY differ.	11.7.1	x
O11.22	Consumers MAY choose to implement a more sophisticated algorithm for down-sampling an image presented at a higher resolution than the device resolution, such as a Fant scaler, to prevent aliasing artifacts.	11.7.2	x
O11.23	Consumers MAY choose any technique desired to achieve the requirement to precisely place a tile possibly resulting in fractional device pixel placement, such as linear filtering for seams, stretching of the tile (up-sampling or down-sampling), or pre-computing multiple tiles and adjusting behavior according to how the tiles fit on a grid.	11.7.3	x
O11.24	Temporary work canvases MAY be re-used when tiling transparent brushes.	11.7.4	x
O11.25	Producers MAY generate a <Path> element intended to be treated as having a consistent nominal stroke width by specifying the <b>StrokeDashArray</b> attribute and by specifying the <b>StrokeDashOffset</b> attribute value less than -1.0 times the sum of all the numbers in the <b>StrokeDashArray</b> attribute value.	11.6.12	x

1 **I.12 Additional Conformance Requirements**

2 *Table I-28. Additional MUST conformance requirements*

ID	Rule	Reference	Producer	Consumer
M12.1	FixedDocument parts MUST be referenced by <DocumentReference> elements within the FixedDocumentSequence part in ascending order. If additional FixedDocument parts are inserted into a fixed document sequence, producers MUST NOT unintentionally change the order of the existing FixedDocument part references.	-	x	
M12.2	A FixedDocument part MUST NOT be referenced more than once	-	x	v

	by a FixedDocumentSequence part.			
M12.3	A FixedPage part MUST NOT be referenced more than once <i>in total</i> , throughout all FixedDocument parts.	-	x	v
M12.4	FixedPage parts MUST be referenced by <PageContent> elements within a fixed document in ascending order. If additional FixedPage parts are inserted into a FixedDocument part, producers MUST NOT unintentionally change the order of the existing FixedPage part references. Documents in languages for which the reading order of pages is back-to-front can be accommodated by adding <PageContent> elements to the FixedDocument in reverse order or by binding the right side of the page.	-	x	
M12.5	Any FixedDocumentSequence, FixedDocument, or FixedPage part that is reachable from the primary fixed payload root or its related parts by relationship or by the <b>Source</b> attribute on a <DocumentReference> or <PageContent> element MUST have no more than one attached PrintTicket part.	-	x	v
M12.6	Every Font part reachable from the primary fixed payload root or its related parts by relationship or by the <b>Source</b> attribute on a <DocumentReference> or <PageContent> element MUST be a valid OpenType font.	-	x	v
M12.7	The content types defined in this specification MUST NOT include parameters. A consumer MUST treat the presence of parameters on these content types as an error when the affected part is accessed.	<b>I.2</b>	x	v

1 **End of informative text.**



## 1 J. Bibliography

### 2 BNF of Generic URI Syntax

3 "BNF of Generic URI Syntax." World Wide Web Consortium.  
4 [http://www.w3.org/Addressing/URL/5\\_URI\\_BNF.html](http://www.w3.org/Addressing/URL/5_URI_BNF.html)

### 5 EXIF Specification

6 Technical Standardization Committee on AV & IT Storage Systems and Equipment.  
7 "Exchangeable Image File Format for Digital Still Cameras: Exif Version 2.2." Japan  
8 Electronic Industry Development Association. 2002. <http://www.jeita.or.jp>

### 9 Fant Scaler

10 Fant, Karl M. "A nonaliasing, real-time spatial transform technique." *IEEE Computer*  
11 *Graphics and Applications* 6 (Jan. 1986): 71-80.

### 12 HTML

13 Jacobs, Ian, Arnaud Le Hors, and Dave Raggett (editors). "HTML 4.01 Specification."  
14 World Wide Web Consortium. 1999. [http://www.w3.org/TR/1999/REC-html401-](http://www.w3.org/TR/1999/REC-html401-19991224/)  
15 [19991224/](http://www.w3.org/TR/1999/REC-html401-19991224/)

### 16 ICC Color Profile

17 International Color Consortium. "ICC Profile Format Specification, Version 3.4."  
18 1997. <http://www.color.org/icc34.pdf>

### 19 JPEG

20 Hamilton, Eric. "JPEG File Interchange Format, Version 1.02." World Wide Web  
21 Consortium. 1992. <http://www.w3.org/Graphics/JPEG/jfif3.pdf>

22 Independent JPEG Group. <http://www.ijg.org/files/>

23 International Telecommunication Union (ITU). "Digital Compression and Coding of  
24 Continuous-tone Still Images." 1993. <http://www.w3.org/Graphics/JPEG/itu-t81.pdf>

### 25 Markup Compatibility

26 Ecma International. "Office Open XML Part 5: Open XML Markup Compatibility."  
27 2006. <http://www.ecma-international.org>

### 28 Open Packaging Conventions

29 Ecma International. "Office Open XML Part 2: Open Packaging Conventions." 2006.  
30 <http://www.ecma-international.org>

### 31 OpenType

32 Microsoft Corporation. "OS/2 and Windows Metrics." 2001.  
33 <http://www.microsoft.com/typography/otspec/os2.htm>

34 Microsoft Corporation. "OpenType Font File." 2001.  
35 <http://www.microsoft.com/typography/otspec/otff.htm>

## J Bibliography

- 1 Microsoft Corporation. "OpenType Specification, Version 1.4." 2004.  
2 <http://www.microsoft.com/typography/otspec/default.htm>
- 3 **PNG**
- 4 Duce, David (editor). "Portable Network Graphics (PNG) Specification," Second  
5 Edition. World Wide Web Consortium. 2003. [http://www.w3.org/TR/2003/REC-PNG-](http://www.w3.org/TR/2003/REC-PNG-20031110)  
6 [20031110](http://www.w3.org/TR/2003/REC-PNG-20031110)
- 7 **Print Schema**
- 8 Microsoft Corporation. "Print Schema." 2006.  
9 <http://windowssdk.msdn.microsoft.com/en-us/library/default.aspx>
- 10 **RFC 2119**
- 11 Bradner, S. "Key words for use in RFCs to Indicate Requirement Levels." The  
12 Internet Society. 1997. <http://www.rfc-editor.org>
- 13 **RFC 3066**
- 14 Alvestrand, H. "Tags for the Identification of Languages." The Internet Society. 2001.  
15 <http://www.rfc-editor.org>
- 16 **RFC 4234**
- 17 Crocker, D. (editor). "Augmented BNF for Syntax Specifications: ABNF." The Internet  
18 Society. 2005. <http://www.rfc-editor.org>
- 19 **scRGB**
- 20 International Electrotechnical Commission (IEC). "Extended RGB colour space—  
21 scRGB." 2003. <http://domino.iec.ch/webstore/webstore.nsf/artnum/029678>
- 22 **sRGB**
- 23 Anderson, Matthew, Srinivasan Chandrasekar, Ricardo Motta, and Michael Stokes. "A  
24 Standard Default Color Space for the Internet—sRGB, Version 1.10." World Wide  
25 Web Consortium. 1996. <http://www.w3.org/Graphics/Color/sRGB>
- 26 **TIFF**
- 27 Adobe Systems Incorporated. "TIFF, Revision 6.0." 1992.  
28 <http://partners.adobe.com/public/developer/en/tiff/TIFF6.pdf>
- 29 **Unicode**
- 30 The Unicode Consortium. The Unicode Standard, Version 4.0.0, defined by: *The*  
31 *Unicode Standard, Version 4.0*. Boston, MA: Addison-Wesley, 2003.
- 32 **Unicode Character Database**
- 33 Davis, Mark and Ken Whistler. "Unicode Character Database, Revision 4.0.0." The  
34 Unicode Consortium. 2003. [http://www.unicode.org/Public/4.0-Update/UCD-](http://www.unicode.org/Public/4.0-Update/UCD-4.0.0.html)  
35 [4.0.0.html](http://www.unicode.org/Public/4.0-Update/UCD-4.0.0.html)



- 1 **WCS Color Profile**
- 2 Microsoft Corporation. "Windows Color System in Windows Longhorn, WinHEC 2005  
3 Version." 2005. [http://download.microsoft.com/download/5/D/6/5D6EAF2B-7DDF-  
4 476B-93DC-7CF0072878E6/WCS.doc](http://download.microsoft.com/download/5/D/6/5D6EAF2B-7DDF-476B-93DC-7CF0072878E6/WCS.doc)
- 5 **Windows Media Photo**
- 6 Microsoft Corporation. <http://www.microsoft.com/xps>
- 7 **XML**
- 8 Bray, Tim, Eve Maler, Jean Paoli, C. M. Sperberg-McQueen, and François Yergeau  
9 (editors). "Extensible Markup Language (XML) 1.0 (Fourth Edition)." World Wide Web  
10 Consortium. 2006. <http://www.w3.org/TR/2006/REC-xml-20060816/>
- 11 **XML Base**
- 12 Marsh, Jonathan. "XML Base." World Wide Web Consortium. 2001.  
13 <http://www.w3.org/TR/2001/REC-xmlbase-20010627/>
- 14 **XML Namespaces**
- 15 Bray, Tim, Dave Hollander, Andrew Layman, and Richard Tobin (editors).  
16 "Namespaces in XML 1.0 (Second Edition)." World Wide Web Consortium. 2006.  
17 <http://www.w3.org/TR/2006/REC-xml-names-20060816/>
- 18 **XML Schema**
- 19 Beech, David, Murray Maloney, Noah Mendelsohn, and Henry S. Thompson (editors).  
20 "XML Schema Part 1: Structures," Second Edition. World Wide Web Consortium.  
21 2004. <http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/>
- 22 Biron, Paul V. and Ashok Malhotra (editors). "XML Schema Part 2: Datatypes,"  
23 Second Edition. World Wide Web Consortium. 2004.  
24 <http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/>
- 25



# 1 K. Index

2 In the index that follows, italic page numbers are used to indicate illustrations and  
 3 examples with illustrations. Bold page numbers are used to indicate a primary  
 4 references when several pages are listed. Page ranges are elided. "See" references  
 5 indicate the primary index location for that topic, while "See also" references indicate  
 6 related index topics.

7	<hr/>	50	offset .....	166
8	<b>A</b>	51	specifying.....	166
9	abbreviated geometry syntax .....	52	image brush	
10	abbreviated syntax .....	53	described .....	<b>127</b> , 129
11	accessibility .....	54	example.....	130
12	document structure, enabled by .....	55	image.....	See image
13	image text alternative, long	56	mentioned .....	24, 67
14	described .....	57	source, specifying .....	128
15	on canvas .....	58	tile size and placement.....	See brush, view port
16	on path.....	59	tile source.....	See brush, view box
17	image text alternative, short	60	linear gradient brush	
18	described .....	61	color interpolation mode .....	151
19	on canvas .....	62	described .....	153
20	on path.....	63	end point .....	151
21	importance of.....	64	example.....	153
22	mentioned.....	65	gradient stops, specifying.....	157
23	of text .....	66	mappng mode .....	151
24	page elements, requirement to expose all .....	67	rendering.....	274
25	reading order	68	specifying.....	151
26	document structure, dependent on.....	69	spread method	
27	fragment type, dependent on.....	70	described .....	154
28	markup order, dependent on .....	71	Pad.....	154
29	of page-centric application .....	72	Reflect.....	155
30	of story-centric application.....	73	Repeat.....	156
31	screen reader	74	specifying.....	151
32	considerations.....	75	start point.....	151
33	mentioned.....	76	opacity .....	See opacity
34	alpha .....	77	radial gradient brush	
35	anti-aliasing	78	color interpolation mode .....	158
36	disabling of .....	79	described .....	160
37	rendering of.....	80	example.....	161
38	application media size.....	81	gradient center.....	158
39	arc .....	82	gradient origin .....	158
	See geometry, segment, arc	83	gradient stops, specifying.....	165
40	<hr/>	84	gradient x-radius .....	158
	<b>B</b>	85	gradient y-radius .....	158
41	bleed area .....	86	mappng mode .....	158
42	bookmark.....	87	rendering.....	276
43	brush .....	88	specifying.....	158
44	alpha.....	89	spread method	
45	described .....	90	described .....	162
46	gradient computations .....	91	Pad.....	162
47	gradient stop	92	Reflect.....	163
48	color, specifying.....	93	Repeat.....	164
49	described .....	94	specifying.....	158
		95	solid color brush .....	126, 127
		96	tile	

## K Index

1	behavior.....	See brush, tile, mode	55	elements applied to.....	171, 172
2	image scaling.....	299	56	geometry, reference to.....	77
3	mode		57	of canvas.....	61, 182
4	described.....	141–50	58	of glyphs.....	98, 184
5	FlipX.....	145–46	59	of path.....	69, 183
6	FlipXY.....	149–50	60	rules.....	270
7	FlipY.....	147–48	61	CMYK.....	See color, color space, CMYK
8	mentioned.....	136	62	color	
9	None.....	141–42	63	alpha.....	See opacity
10	specifying.....	128, 131, 136	64	black generation.....	221
11	Tile.....	143–44	65	blending.....	vii
12	placement.....	See brush, view port	66	behavior described.....	217
13	size.....	See brush, view port	67	blend color space for linear gradient brush....	151
14	small tile rendering.....	299	68	blend color space for radial gradient brush....	158
15	source.....	See brush, view box	69	color space.....	220
16	transparent brush tiling.....	300	70	for gradients.....	273
17	transformation.....	See transformation	71	implementation dependent.....	217
18	view box		72	brush, specifying for.....	See brush
19	calculating source coordinates for images.....	136	73	color management on device.....	219
20	described.....	136	74	color profile	
21	example.....	137–40	75	embedded in image.....	24, 25, 26, 27, 29, 207
22	larger than image.....	136, 140	76	ICC profile	
23	mapping to view port.....	128, 131	77	color channels supported.....	205, 206
24	specifying for image brush.....	128	78	colorant table usage for named colors.....	207
25	specifying for visual brush.....	131	79	described.....	206
26	syntax.....	128, 131	80	grayscale image, usage with.....	207
27	unit type.....	128, 131, 136	81	parts.....	See parts
28	units, for images.....	128, 136	82	profile types allowed.....	206
29	view port		83	version supported.....	205
30	described.....	136	84	Windows Color System (WCS) profile	
31	example.....	137–40	85	embedded in ICC profile.....	207
32	placement precision.....	299	86	mentioned.....	205, 208
33	specifying for image brush.....	128	87	WcsProfilesTag.....	207
34	specifying for visual brush.....	131	88	color separation.....	217
35	syntax.....	128, 131	89	color space	
36	unit type.....	128, 131, 136	90	CMYK.....	206
37	visual brush		91	in images.....	205
38	described.....	131, 133	92	in vector graphics.....	205
39	example.....	134	93	device color.....	206, 219
40	visual.....	131, 133	94	gray colors.....	206
			95	ICC profile	
			96	version supported.....	206
			97	named color.....	viii, 206
			98	named colors.....	207
			99	n-channel.....	206
			100	in images.....	205
			101	in vector graphics.....	205
			102	scRGB.....	206
			103	gamut boundary definition.....	399
			104	in images.....	205
			105	in vector graphics.....	205
			106	spot colors... See color, color space, named colors	
			107	sRGB.....	205
			108	in images.....	205
			109	in vector graphics.....	205
			110	support required.....	205
			111	support summarized.....	205
41	<b>C</b>				
42	canvas				
43	anti-aliasing control.....	61			
44	clipping.....	See clipping			
45	composing properties.....	64			
46	described.....	61			
47	opacity.....	See opacity			
48	opacity mask.....	See opacity mask			
49	transformation.....	See transformation			
50	caret stop.....	See selection, caret stop			
51	CFF font.....	See font:CFF			
52	circle.....	See geometry, segment, arc			
53	clipping				
54	described.....	182			

1	document imposition color .....	221	55	certificate	
2	fidelity, improved .....	205	56	relationship to .....	22
3	gamma.....	<i>See color, blending</i>	57	store .....	261
4	pixel formats.....	<i>See color, raster image support</i>	58	validity .....	260
5	printing .....	<i>See PrintTicket keywords</i>	59	co-signature .....	<i>See digital signature, signature</i>
6	raster image support		60	definitions	
7	associating a color profile part.....	216	61	multiple signatures .....	260
8	CMYK.....	215	62	namespace .....	<i>See namespace</i>
9	device color.....	216	63	Open Packaging Conventions, extended from....	258
10	gray colors.....	214	64	origin	
11	named colors .....	216	65	part.....	<i>See parts</i>
12	n-channel .....	215	66	relationship to .....	22
13	pixel format defaults.....	216	67	parts.....	<i>See parts</i>
14	scRGB .....	214	68	printing .....	261
15	sRGB.....	213	69	relationship to.....	22
16	rendering intent, specifying.....	221	70	relationships .....	<i>See relationships</i>
17	syntax		71	request... ..	<i>See digital signature, signature definitions</i>
18	CMYK.....	210	72	signature definitions .....	<b>ix</b>
19	named color .....	212	73	described .....	38, 261
20	n-channel .....	210	74	markup example.....	262
21	scRGB .....	209	75	mentioned .....	20
22	sRGB .....	209	76	namespace .....	<i>See namespace</i>
23	summarized .....	209	77	relationship to .....	22
24	where used .....	208	78	sign by date and time .....	265
25	values, exponent prohibited.....	209	79	signer name.....	263
26	color profile.....	<i>See color, color profile</i>	80	signing intent.....	264
27	composability of properties .....	<i>See XPS Document</i>	81	signing location .....	265
28	format, properties, composability		82	specifying.....	262
29	compression		83	spot ID .....	263
30	image.....	<i>See image</i>	84	spot location.....	263, 264
31	package.....	<i>See Open Packaging Conventions,</i>	85	signature policy	
32	specification		86	conditions where policy does not apply .....	259
33	conformance		87	described .....	258
34	inherited from Open Packaging Conventions .....	15	88	markup compatibility impact .....	260
35	language notes .....	<b>iii</b>	89	parts to sign	
36	of software .....	<b>iii</b>	90	optional.....	259
37	requirements tables .....	409–51	91	required .....	258
38	consumer .....	<b>vii</b>	92	relationships to sign	
39	described .....	15	93	as a group, required.....	259
40	implementation burden .....	409–51	94	conditionally required.....	259
41	content area.....	<i>See page, content area</i>	95	required .....	259
42	content type.....	<b>vii</b>	96	signing rules.....	<b>x</b> , 258–60
43	namespace .....	<i>See namespace</i>	97	signing validity.....	260
44	summarized .....	406	98	single signature .....	259
45	usage of .....	19	99	signature spot .....	<i>See digital signature, signature</i>
46	contour intersection point .....	<b>vii</b>	100	definitions	
47	copy and paste .....	<i>See also selection</i>	101	signature status	
48	document structure, improved by.....	223	102	broken digital signature .....	<b>vii</b> , 260
49	core properties .....	<i>See Open Packaging Conventions</i>	103	compliant digital signature.....	<b>vii</b>
50	curve .....	<i>See geometry, segment</i>	104	incompliant digital signature.....	<b>viii</b> , 260
			105	questionable digital signature .....	<b>ix</b> , 260
			106	valid digital signature .....	<b>x</b> , 260
51	<b>D</b>		107	discard control	
52	device.....	<b>vii</b>	108	consumer considerations.....	255
53	device color.....	<i>See color, color space, device color</i>	109	elements .....	256
54	digital signature		110	markup example .....	256
			111	namespace.....	<i>See namespace</i>





## K Index

1	<b>I</b>		
2	ICC.....	See color, color profile	
3	image		
4	brush .....	See brush, image brush	
5	JPEG.....	20, 24, 25	
6	APP markers.....	25	
7	CMYK.....	25	
8	compression.....	See image, JPEG, specification	
9	EXIF .....	25	
10	naming .....	25	
11	specification.....	25, 453	
12	parts .....	See parts	
13	PNG .....	20, 24, 25	
14	chunks .....	25	
15	naming .....	25	
16	specification .....	25	
17	relationships....	See relationships, required resource	
18	resolution .....	136	
19	resource .....	See resource	
20	sharing.....	24	
21	thumbnail .....	20	
22	TIFF.....	20, 24, 25	
23	alpha, associated .....	29	
24	CCITT bilevel encoding .....	28	
25	CMYK.....	28	
26	compression.....	29, See image, TIFF, specification	
27	EXIF .....	29	
28	features, supported .....	28	
29	image file directory (IFD) .....	27	
30	naming .....	25	
31	specification.....	25, 29, 454	
32	tags, supported .....	26	
33	tags, unsupported .....	28	
34	tags, unsupported .....	29	
35	variations, handling.....	29	
36	types supported .....	24, 129	
37	usage described.....	24, 67	
38	Windows Media Photo .....	20, 24, 29	
39	CMYK.....	30	
40	compression....	See image, Windows Media Photo, specification	
41	specification .....		
42	features supported .....	30	
43	grayscale .....	30	
44	named color .....	30	
45	naming .....	29	
46	N-channel.....	30	
47	profiled RGB.....	30	
48	scRGB .....	30	
49	specification.....	29, 455	
50	sRGB.....	30	
51	imageable size.....	See PrintTicket keywords, PageImageableSize	
52	implementation limits.....	270–72	
53	ink area .....	See page, content area	
54			
55	interleaving .....	See Open Packaging Conventions, interleaving	
56			
57	<b>J</b>		
58	JPEG.....	See image, JPEG	
59	<b>L</b>		
60	landscape orientation .....	See page, orientation of	
61	language		
62	markup.....	48, 172	
63	of canvas .....	61	
64	of glyphs .....	98, 119	
65	of outline .....	48, 226	
66	of outline entry .....	48, 227	
67	of page .....	53	
68	of path .....	69	
69	of resource.....	177	
70	of signature definition .....	263	
71	usage.....	47	
72	layout .....	See also transformation, matrix	
73	composition		
74	behavior .....	283	
75	examples .....	284–86	
76	optimization .....	283	
77	rules.....	282	
78	coordinate rounding .....	267	
79	coordinate space .....	185	
80	composability .....	54	
81	effective coordinate space .....	vii	
82	described .....	45	
83	mentioned .	53, 69, 98, 110, 125, 136, 151, 158, 172, 180	
84	mentioned.....	49	
85	origin .....	267	
86	transformation .....	See transformation	
87	units .....	267	
88	x-axis .....	267	
89	y-axis .....	267	
90	degenerate segments .....	297	
91	implementation limits ....	See implementation limits	
92	page dimensions .....	See page	
93	pixel		
94	center location .....	See layout, pixel	
95	inclusion .....	269	
96	placement .....	269	
97	placement behavior for glyphs.....	269	
98	placement error maximum .....	269	
99	rendering.....	269	
100	sub-pixel masking.....	269	
101	PrintTicket interactions.....	56	
102	shape abutment.....	269	
103	line		
104	characteristics of.....	See stroke	
105			



1	curved.....	See geometry, segment
2	drawing of .....	See path
3	geometry of.....	See geometry, segment, line
4	linear gradient.....	See brush, linear gradient brush
5	link .....	See hyperlink
6	link target.....	See hyperlink, target
7	list .....	See document structure, list
<hr/>		
8	<b>M</b>	
9	markup compatibility	
10	digital signature, impacted by .....	260
11	document structure, usage in.....	243
12	mentioned.....	1
13	namespace .....	See namespace
14	mentioned.....	44
15	preprocessing requirements .....	44
16	processing required.....	43
17	property elements, usage with.....	45
18	resource dictionary, usage in.....	181
19	specification .....	453
20	usage of .....	43
21	media size ..	See PrintTicket keywords, PageMediaSize
22	media size, application.....	56
23	memory management, device .....	See interleaving;
24	discard control	
25	miter .....	See stroke, line, join
<hr/>		
26	<b>N</b>	
27	name	
28	elements applied to.....	171
29	link target correspondence .....	245
30	link target, specifying as .....	51
31	purpose of .....	245
32	resource entries, prohibited for .....	245
33	syntax .....	245
34	uniqueness .....	245
35	named color.....	See color, color space, named color
36	namespace .....	405
37	content type.....	405
38	core properties .....	405
39	digital signatures .....	405
40	discard control.....	405
41	document .....	405
42	document sequence .....	405
43	document structure.....	405
44	markup compatibility .....	405
45	page.....	405
46	Print Schema framework.....	405
47	Print Schema keywords .....	405
48	relationships .....	405
49	resource dictionary key .....	405
50	signature definitions.....	405
51	story fragments .....	405
52	naming of parts..	See parts, naming recommendations
53	natural language.....	See language
54	n-channel color.....	See color, color space, n-channel
<hr/>		
55	<b>O</b>	
56	opacity	
57	blending.....	See color, blending
58	brush initial opacity .....	136
59	composition effects .....	283
60	computations.....	279–81
61	described .....	172
62	elements applied to.....	171
63	mask.....	See opacity mask
64	of canvas .....	61
65	of color... 30, 126, 151, 158, 205, 209, 210, 212, 217	
66	of glyphs.....	98
67	of image brush .....	128
68	of linear gradient brush .....	151
69	of path .....	69
70	of pixel formats.....	213, 214, 215
71	of radial gradient brush .....	158
72	of solid color brush .....	126
73	of stroke.....	283
74	of visual brush.....	131
75	pre-multiplied alpha .....	281
76	superluminous colors.....	281
77	transparent brush tiling .....	300
78	value range .....	279
79	opacity mask	
80	brush, filling with .....	See brush
81	described .....	167
82	elements applied to.....	171, 172
83	example .....	167, 169
84	of canvas .....	61, 200
85	of glyphs.....	98, 202
86	of path .....	69, 201
87	Open Packaging Conventions	
88	core properties .....	265
89	namespace .....	See namespace
90	digital signature .....	See digital signature
91	interleaving .....	viii
92	consumer considerations .....	255
93	optimization .....	249
94	parsing head-first vs. tail first.....	249
95	piece .....	viii
96	package.....	viii
97	packaging model .....	viii, 18
98	relationship .....	viii
99	physical model .....	viii, 18
100	specification .....	453
101	OpenType font .....	See font:OpenType
102	optimization	
103	for streaming consumption .....	See Open Packaging
104	Conventions, interleaving	
105	of composition rules .....	283
106	of digital signatures.....	257

## K Index

- 1 of glyphs .....120  
 2 of interleaving .....251–55  
 3 of named element location .....224  
 4 of pixel placement rules .....269  
 5 outline.....*See* document structure, outline
- 
- 6 **P**
- 7 packaging model .....*See* Open Packaging Conventions  
 8 page  
 9 bleed area.....53, 54  
 10 content area .....53, 55, 267  
 11 height of .....53, 267  
 12 height, advisory .....51  
 13 imageable area .....*See* PrintTicket keywords,  
 14 PageImageableSize  
 15 layout .....*See also* layout  
 16 markup grouping ..... *See* canvas  
 17 namespace .....*See* namespace  
 18 order of pages .....50  
 19 orientation of .....56, 58  
 20 reference to .....50  
 21 root of .....53  
 22 scaling for print .....56  
 23 size terminology .....55  
 24 uniqueness of .....51  
 25 width of .....53, 267  
 26 width, advisory .....51  
 27 paragraph.....*See* document structure, paragraph  
 28 part .....**viii**  
 29 part name .....**viii**  
 30 parts..... *See also* XPS Document format  
 31 core properties .....*See* Open Packaging Conventions,  
 32 specification  
 33 digital signature  
 34 certificate  
 35 part.....*See* Open Packaging Conventions,  
 36 specification  
 37 digital signature origin.....*See* Open Packaging  
 38 Conventions, specification  
 39 DiscardControl.....20, 256  
 40 DocumentStructure.....20, 38, 224  
 41 FixedDocument .....19, 24  
 42 FixedDocumentSequence.....19, 23  
 43 FixedPage .....20, 24  
 44 font.....20, 31–36  
 45 ICC profile .....20  
 46 image .....20, 24–31  
 47 naming recommendations .....40–42  
 48 PrintTicket .....20, 36  
 49 remote resource dictionary.....20, 36  
 50 SignatureDefinitions.....20, 38  
 51 StoryFragments .....20, 39, 230  
 52 thumbnail .....20, 30  
 53 thumbnail, package .....*See* Open Packaging  
 54 Conventions, specification
- 55 XML digital signature ..... *See* Open Packaging  
 56 Conventions, specification  
 57 path  
 58 clipping ..... *See* clipping  
 59 described ..... 67, 68  
 60 fill brush..... 69  
 61 geometry, reference to..... 69, 74  
 62 opacity ..... *See* opacity  
 63 opacity mask.....*See* opacity mask  
 64 shape ..... *See* geometry  
 65 stroke brush..... 69  
 66 stroke control .....*See* stroke  
 67 transformation.....*See* transformation  
 68 usage described ..... 73  
 69 payload.....**viii**, *See* XPS Document format  
 70 physical imageable size.**viii**, *See* PrintTicket keywords,  
 71 PageImageableSize  
 72 physical media size.....**viii**, *See* PrintTicket keywords,  
 73 PageMedaSize  
 74 physical model ..... *See* Open Packaging Conventions  
 75 physical organization..... *See* Open Packaging  
 76 Conventions, interleaving; ZIP  
 77 pixel..... *See* layout, pixel  
 78 pixel snapping ..... *See* stroke, line  
 79 PNG ..... *See* image, PNG  
 80 portrait orientation ..... *See* page, orientation of  
 81 positioning content *See* layout; transformation, matrix  
 82 primary fixed payload root..... **viii**  
 83 Print Schema .....*See* PrintTicket  
 84 printing  
 85 bleed area ..... *See* page, bleed area  
 86 content area .....*See* page, content area  
 87 device fonts ..... *See* font, device font  
 88 digital signature ..... *See* digital signature  
 89 discard control.....*See* discard control  
 90 font, print and preview restricted *See* font, licensing  
 91 rights  
 92 interleaving..... *See* Open Packaging Conventions,  
 93 interleaving  
 94 layout .....*See* layout  
 95 orientation ..... *See* PrintTicket keywords  
 96 PrintTicket..... *See* PrintTicket; PrintTicket keywords  
 97 resource constraints .....*See* discard control  
 98 scaling ..... *See* PrintTicket keywords  
 99 PrintTicket ..... **viii**  
 100 described ..... 36  
 101 empty PrintTicket, markup of ..... 251  
 102 mapping content levels to parts ..... 36  
 103 namespace.....*See* namespace  
 104 parts.....*See* parts  
 105 processing requirements ..... 37  
 106 relationships ..... *See* relationships  
 107 PrintTicket keywords  
 108 DocumentImpositionColor ..... 221  
 109 ICMRenderingIntent ..... 221  
 110 JobDigitalSignatureProcessing ..... 261  
 111 PrintInvalidSignatures ..... 261

1	PrintInvalidSignaturesWithErrorReport	261
2	PrintOnlyValidSignatures	261
3	namespace	See namespace
4	PageBlackGenerationProcessing	221
5	PageBlendColorSpace	220
6	PageColorManagement	219
7	PageDeviceColorSpaceProfileURI	219
8	PageDeviceColorSpaceUsage	219
9	PageImageableSize	56, 58, 59, 60
10	PageMediaSize	55, 56, 58, 60, 267
11	PageOrientation	56, 58
12	PageScaling	58–60
13	producer	ix
14	bleed size	ix, 55
15	content size	ix, 55
16	described	15
17	implementation burden	409–51
18	media size	ix, 55
19	property	See XPS Document format, properties
<hr/>		
20	<b>R</b>	
21	radial gradient	See brush, radial gradient brush
22	raster graphics	See image
23	reading order	See accessibility, reading order
24	relationship	ix
25	StartPart	x
26	relationships	See also XPS Document format
27	core properties	22
28	digital signature	22
29	digital signature certificate	22
30	digital signature definitions	22
31	digital signature origin	22
32	DiscardControl	22
33	DocumentStructure	22
34	external prohibited	22
35	namespace	See namespace
36	PrintTicket	22
37	purpose of	22
38	relationship types, reference table	407
39	required resource	23
40	restricted font	23, 32, 35
41	StartPart	19, 23
42	StoryFragments	23
43	thumbnail	23
44	usage of	23
45	relationships part	ix
46	remote resource dictionary	ix
47	rendering	See also layout
48	rules described	267–300
49	required part	ix
50	resource	See resource dictionary, resource definition
51	resource constraints	See discard control
52	resource definition	See resource dictionary, resource definition
53	definition	
54	resource dictionary	ix
55	described	173, 176
56	example	173, 175, 177, 178, 180
57	markup compatibility usage	181
58	remote	See resource part, remote resource dictionary
59	dictionary	
60	resource definition	ix
61	described	172, 176
62	key	
63	described	173
64	namespace	See namespace
65	on canvas	61
66	on geometry	78
67	on glyphs	98
68	on image brush	128
69	on linear gradient brush	151
70	on matrix transformation	185
71	on path	69
72	on radial gradient brush	158
73	on solid color brush	126
74	on visual brush	131
75	uniqueness	180
76	language	177
77	locating	181
78	namespace prefixes, interpreting	177
79	referencing previously-defined resources	176, 177, 180
80	usefulness of path, glyphs, and canvas as	173
81	sharing	See resource part, remote resource dictionary
82	specifying	172, 173–77
83	where usable	173
84	resource part	
85	color profile	See color, color profile
86	font	See font
87	image	See image
88	reference to	22, 24
89	relationships to	See relationships
90	remote resource dictionary	20, 22, 173, 177
91	required resource	22
92	usage of	22
93	resource reference	ix
94	described	173, 180
95	example	180
96	scope	180
97	syntax	180
98	rotating content	See transformation, matrix
99		
100		
<hr/>		
101	<b>S</b>	
102	scalable vector graphics (SVG)	See geometry,
103	abbreviated syntax	
104	scaling	
105	content	See transformation, matrix
106	for print	See page:scaling for print
107	scRGB	See color, color space, scRGB
108	search	See find

## K Index

1	section .....See document structure, section	58	pixel snapping ..... 69
2	segment .....See geometry, segment	59	thin stroke anti-aliasing behavior..... 269
3	selection	60	zero-length, avoided ..... 82
4	behavior .....246	61	multi-figure path, behavior with..... 298
5	caret stop .....98, 119	62	phase control ..... 288
6	document structure, enabled by .....223, 246	63	segments, mixed stroked and non-stroked ..... 298
7	fragment type, behavior depending on .....246	64	thickness ..... 69
8	mentioned ..... 111, 119, 223		
9	order .....246		
10	recommended .....246		
11	shear .....See transformation, matrix	65	<b>T</b>
12	signature definitions ..... See digital signature	66	table .... See document structure, table, See document
13	signature spot .....x, See digital signature, signature	67	structure, table
14	definitions	68	table of contents .....See document structure, outline
15	simple ordering .....x	69	text
16	size terminology ..... See page, size terminology	70	baseline.....See text, glyph, baseline
17	skewing content .....See transformation, matrix	71	bidirectional..... 98, 115
18	sRGB ..... See color, color space, sRGB	72	bold ..... See text, style simulation
19	starting part ..... See relationships, StartPart	73	clipping ..... See clipping
20	story ..... See document structure, story	74	fill brush ..... 98, 124
21	story fragment ..... See document structure, story	75	font .....See also font
22	fragment	76	device font, reference to ..... 98, 118
23	stream .....x	77	reference to ..... 98
24	stretching content .....See transformation, matrix	78	glyph
25	stroke	79	advance width .....104, 105, 109, 110
26	brush ..... See path, stroke brush	80	baseline ..... 104
27	consistent nominal width ..... See stroke, hairline	81	black box ..... 104
28	contour rendering .....288	82	cluster map .....105–8, 109, 110, 111
29	dash	83	indices
30	cap .....289	84	described ..... 109
31	flat .....289	85	reference to non-existent glyph ..... 109
32	round .....290	86	restriction of length ..... 109
33	square .....289	87	specifying ..... 98
34	style .....69	88	syntax ..... 109
35	triangle .....290	89	metrics ..... 104
36	offset .....69	90	offset .....105, 109, 110, 115
37	overlapping .....291	91	origin ..... 104, 105
38	style .....69	92	origin, sideways ..... 104
39	drawing algorithm, symmetry of .....288	93	side-bearing, bottom ..... 105
40	edge parallelization .....288	94	side-bearing, left ..... 104
41	fill rule, independence from .....298	95	side-bearing, right ..... 104
42	hairline .....298	96	side-bearing, top ..... 105
43	line	97	glyphs usage for ..... 97
44	cap .....291	98	italic ..... See text, style simulation
45	end .....69	99	markup ..... 98
46	flat .....292	100	markup examples ..... 121
47	for dashed stroke .....292	101	markup optimization ..... 120
48	round .....292	102	opacity ..... See opacity
49	square .....292	103	opacity mask ..... See opacity mask
50	start .....69	104	position ..... See also text, glyph, offset
51	triangle .....292	105	horizontal text ..... 98
52	join	106	vertical text ..... 112
53	bevel .....294	107	sideways (vertical) ..... 98
54	miter .....295–97	108	advance width ..... 109, 113
55	round .....293	109	bidirectional text, intersection with ..... 114
56	style .....69	110	described ..... 112
57	miter limit .....69	111	example, sideways ..... 116

1	example, vertical.....	116, 117	58	of tiles .....	194
2	horizontal text, including .....	113	59	of tiles, example.....	196
3	origin calculation.....	113	60	of visual brush.....	131, 194
4	vertical glyphs, preference for .....	113	61	transparency .....	See opacity
5	size.....	98	62	TrueType collection (TTC) font .....	See font:TrueType
6	style simulation .....	98, 112	63	collection (TTC)	
7	transformation .....	See transformation	64	TrueType font.....	See font:TrueType
8	underline .....	See path			
9	Unicode string .....	See also text, glyph			
10	escaping open brace character .....	111	65	<b>V</b>	
11	mapping code units to glyphs .....	105	66	vector graphics .....	See path
12	normalization prohibited .....	111	67	versioning.....	See markup compatibility
13	specifying .....	98			
14	Unicode control marks, inclusion of .....	111	68	<b>W</b>	
15	Unicode scalar value, split into code units.....	105	69	whitespace .....	See XPS Document format, XML,
16	unmappable code unit behavior .....	111	70	whitespace	
17	usage of.....	111	71	Windows Color System (WCS). See color, color profile,	
18	UTF-16 code units, consisting of .....	105	72	Windows Color System (WCS) profile	
19	vertical.....	See text, sideways (vertical)	73	Windows Media Photo.....	See image, Windows Media
20	thread .....	See document structure, story	74	Photo	
21	thumbnail.....	x			
22	described .....	20, 30			
23	formats .....	See image			
24	parts .....	See parts			
25	relationship.....	23			
26	usage .....	30			
27	TIFF.....	See image, TIFF	75	<b>X</b>	
28	transformation		76	XML .....	See XPS Document format, XML
29	composability .....	268	77	XML namespaces.....	See namespace; XPS Document
30	described .....	185, 268	78	format, XML, namespaces	
31	effective transform.....	See transformation,	79	XML Paper Specification	
32	composability		80	document format.....	See XPS Document format
33	elements applied to.....	171, 172	81	organization of.....	16
34	matrix		82	XPS Document format .....	x, 16, 19
35	abbreviated syntax.....	186	83	content types .....	406
36	abbreviated syntax example .....	189	84	described .....	1, 49
37	described .....	185	85	example .....	21
38	example .....	187	86	extensibility.....	See markup compatibility
39	inverting x-axis .....	186	87	fixed payload.....	vii, 19
40	inverting y-axis.....	186	88	fixed payload root.....	vii, 19
41	multiplying .....	185	89	illustrated.....	18
42	positioning .....	187	90	language .....	See language
43	rotating .....	187	91	markup elements.....	See XPS elements
44	scaling .....	186	92	parts.....	18, 19
45	skewing .....	186	93	payload .....	19
46	specifying .....	185	94	properties	
47	mentioned .....	267	95	attribute syntax.....	45
48	non-invertible transform, rendering of elements		96	composability .....	45, 171
49	with.....	268	97	described .....	45
50	of brush .....	125	98	element syntax .....	45
51	of canvas.....	61, 189	99	model .....	44
52	of geometry.....	78, 192	100	ordering.....	46
53	of glyphs .....	98, 191	101	property attribute .....	ix
54	of image brush.....	128, 193	102	property element .....	ix
55	of linear gradient brush .....	151, 197	103	property value .....	ix, 45
56	of path .....	69, 190	104	property.....	ix
57	of radial gradient brush .....	158, 198	105	relationship types .....	407
			106	relationships .....	18, 22

## K Index

1	root.....	49	55	<LinkTarget>.....	52, 326
2	versioning.....	See markup compatibility	56	<PageContent.LinkTargets>.....	51, 330
3	XML		57	<PageContent>.....	50, 329
4	DTDs prohibited.....	44	58	page-level	
5	markup design.....	42	59	<ArcSegment>.....	82, 301
6	markup model.....	44–47	60	<Canvas.Clip>.....	182, 305
7	namespaces.....	44, 405	61	<Canvas.OpacityMask>.....	200, 305
8	Unicode encodings permitted.....	43	62	<Canvas.RenderTransform>.....	189, 306
9	usage.....	43	63	<Canvas.Resources>.....	175, 306
10	whitespace.....	47	64	<Canvas>.....	61, 302
11	XML and XSI namespace usage.....	44	65	<FixedPage.Resources>.....	173, 311
12	XML schema (XSD)		66	<FixedPage>.....	53, 310
13	characteristics.....	44	67	<Glyphs.Clip>.....	184, 317
14	discard control schema.....	391	68	<Glyphs.Fill>.....	124, 317
15	document structure schema.....	385	69	<Glyphs.OpacityMask>.....	202, 318
16	resource dictionary key schema.....	383	70	<Glyphs.RenderTransform>.....	191, 319
17	signature definitions schema.....	359	71	<Glyphs>.....	98, 312
18	validity requirement.....	44	72	<GradientStop>.....	166, 319
19	XPS Document schema.....	361	73	<ImageBrush.Transform>.....	193, 322
20	XPS elements		74	<ImageBrush>.....	127, 320
21	digital signature		75	<LinearGradientBrush.GradientStops> ...	157, 325
22	<Intent>.....	264, 322	76	<LinearGradientBrush.Transform>.....	197, 325
23	<SignatureDefinition>.....	262, 348	77	<LinearGradientBrush>.....	151, 323
24	<SignatureDefinitions>.....	262, 349	78	<MatrixTransform>.....	185, 327
25	<SignBy>.....	265, 348	79	<Path.Clip>.....	183, 337
26	<SigningLocation>.....	265, 349	80	<Path.Data>.....	74, 337
27	<SpotLocation>.....	263, 350	81	<Path.Fill>.....	75, 337
28	discard control		82	<Path.OpacityMask>.....	201, 338
29	<Discard>.....	256, 306	83	<Path.RenderTransform>.....	190, 338
30	<DiscardControl>.....	256, 307	84	<Path.Stroke>.....	76, 338
31	document structure		85	<Path>.....	68, 331
32	<DocumentOutline>.....	226, 307	86	<PathFigure>.....	81, 339
33	<DocumentStructure.Outline>.....	226, 309	87	<PathGeometry.Transform>.....	192, 341
34	<DocumentStructure>.....	224, 308	88	<PathGeometry>.....	78, 339
35	<FigureStructure>.....	242, 309	89	<PolyBezierSegment>.....	86, 341
36	<ListItemStructure>.....	242, 326	90	<PolyLineSegment>.....	87, 342
37	<ListStructure>.....	241, 327	91	<PolyQuadraticBezierSegment>.....	88, 342
38	<NamedElement>.....	243, 328	92	<RadialGradientBrush.GradientStops> ...	165, 345
39	<OutlineEntry>.....	227, 328	93	<RadialGradientBrush.Transform>.....	198, 346
40	<ParagraphStructure>.....	239, 330	94	<RadialGradientBrush>.....	158, 343
41	<SectionStructure>.....	239, 348	95	<ResourceDictionary>.....	176, 346
42	<Story>.....	228, 351	96	<SolidColorBrush>.....	126, 350
43	<StoryBreak>.....	239, 352	97	<VisualBrush.Transform>.....	194, 358
44	<StoryFragment>.....	235, 352	98	<VisualBrush.Visual>.....	133, 358
45	<StoryFragmentReference>.....	229, 353	99	<VisualBrush>.....	131, 356
46	<StoryFragments>.....	234, 353			
47	<TableCellStructure>.....	241, 354			
48	<TableRowGroupStructure>.....	240, 355	100	<b>Z</b>	
49	<TableRowStructure>.....	240, 355	101	ZIP	
50	<TableStructure>.....	240, 355	102	archive.....	x, 18
51	document-level		103	item.....	x
52	<DocumentReference>.....	49, 308	104	utilities.....	33
53	<FixedDocument>.....	50, 309			
54	<FixedDocumentSequence>.....	49, 310			