

1

XML Paper 2 Specification

3
4 XPS Specification and Reference Guide

5
6
7
8 Working Draft 1.4

9
10 January 2009

11
12
13 © 2008–2009 Ecma International. All rights reserved.

14
15
16
17 This document is "a work in progress" and was produced by Ecma Technical Committee TC46.

1	Contents	
2	1. SCOPE	1
3	2. CONFORMANCE	3
4	2.1 Requirements Terminology	3
5	2.2 Implementation Conformance	3
6	2.3 Instantiating Error Conditions	4
7	3. NORMATIVE REFERENCES	5
8	4. DEFINITIONS	9
9	5. NOTATIONAL CONVENTIONS	13
10	5.1 Document Conventions	13
11	5.2 Diagrams	13
12	6. ACRONYMS AND ABBREVIATIONS	15
13	7. GENERAL DESCRIPTION	17
14	8. XPS DOCUMENT FORMAT	19
15	8.1 How This Standard Is Organized	19
16	8.2 Package	21
17	9. PARTS AND RELATIONSHIPS	23
18	9.1 Fixed Payload	23
19	9.1.1 Fixed Payload Relationships.....	26
20	9.1.2 FixedDocumentSequence Part.....	27
21	9.1.3 FixedDocument Part.....	27
22	9.1.4 FixedPage Part.....	28
23	9.1.5 Image Parts.....	28
24	9.1.6 Thumbnail Parts	34
25	9.1.7 Font Parts	35
26	9.1.8 Remote Resource Dictionary Parts	39
27	9.1.9 PrintTicket Parts.....	40
28	9.1.10 SignatureDefinitions Part	42
29	9.1.11 DocumentStructure Part	42
30	9.1.12 StoryFragments Part	43
31	9.2 Part Naming Recommendations	43
32	9.3 XPS Document Markup	45
33	9.3.1 Support for Versioning and Extensibility	46
34	9.3.2 XML Usage	46
35	9.3.3 Markup Model	47
36	9.3.4 Whitespace.....	49
37	9.3.5 Language	49
38	10. DOCUMENTS	51
39	10.1 <FixedDocumentSequence> Element	51
40	10.1.1 <DocumentReference> Element.....	51
41	10.2 <FixedDocument> Element	52
42	10.2.1 <PageContent> Element.....	52
43	10.2.2 <PageContent.LinkTargets> Element	53
44	10.2.3 <LinkTarget> Element.....	54

1 **10.3 <FixedPage> Element..... 55**

2 10.3.1 BleedBox Attribute..... 56

3 10.3.2 ContentBox Attribute 57

4 10.3.3 57

5 10.3.4 Media Orientation and Scaling 58

6 **10.4 <Canvas> Element 63**

7 **10.5 <Path> Element 66**

8 **10.6 <Glyphs> Element..... 67**

9 **11. GRAPHICS..... 69**

10 **11.1 <Path> Element 70**

11 11.1.1 <Path.Data> Element 74

12 11.1.2 <Path.Fill> Element..... 75

13 11.1.3 <Path.Stroke> Element..... 76

14 **11.2 Geometries and Figures 77**

15 11.2.1 Geometries 79

16 11.2.2 Figures 82

17 11.2.3 Abbreviated Geometry Syntax 91

18 **12. TEXT 99**

19 **12.1 <Glyphs> Element..... 100**

20 12.1.1 Glyph Metrics.....105

21 12.1.2 Mapping Code Units to Glyphs106

22 12.1.3 Indices Attribute110

23 12.1.4 UnicodeString Attribute112

24 12.1.5 StyleSimulations Attribute112

25 12.1.6 IsSideways Attribute113

26 12.1.7 DeviceFontName Attribute118

27 12.1.8 xml:lang Attribute119

28 12.1.9 CaretStops Attribute119

29 12.1.10 Optimizing Glyph Markup119

30 12.1.11 Glyph Markup Examples120

31 **12.2 <Glyphs.Fill> Element 123**

32 **13. BRUSHES 125**

33 **13.1 <SolidColorBrush> Element 126**

34 **13.2 <ImageBrush> Element..... 127**

35 **13.3 <VisualBrush> Element 131**

36 13.3.1 <VisualBrush.Visual> Element132

37 **13.4 Common Attributes for Tiling Brushes 135**

38 13.4.1 Viewbox, Viewport, ViewboxUnits, and ViewportUnits Attributes135

39 13.4.2 TileMode Attribute140

40 **13.5 <LinearGradientBrush> Element 151**

41 13.5.1 SpreadMethod Attribute.....153

42 13.5.2 <LinearGradientBrush.GradientStops> Element156

43 **13.6 <RadialGradientBrush> Element 157**

44 13.6.1 SpreadMethod Attribute.....160

45 13.6.2 <RadialGradientBrush.GradientStops> Element164

46 **13.7 <GradientStop> Element 165**

47 **13.8 Using a Brush as an Opacity Mask 166**

48 **14. COMMON PROPERTIES 169**

49 **14.1 Opacity 170**

50 **14.2 Resources and Resource References 170**

1	14.2.1	<FixedPage.Resources> Element.....	171
2	14.2.2	<Canvas.Resources> Element	172
3	14.2.3	<ResourceDictionary> Element	173
4	14.2.4	Resource References	176
5	14.2.5	Scoping Rules for Resolving Resource References.....	177
6	14.2.6	Support for Markup Compatibility	178
7	14.3	Clipping.....	178
8	14.3.1	<Canvas.Clip> Element	178
9	14.3.2	<Path.Clip> Element	179
10	14.3.3	<Glyphs.Clip> Element.....	180
11	14.4	Positioning Content	181
12	14.4.1	<MatrixTransform> Element.....	182
13	14.4.2	<Canvas.RenderTransform> Element.....	185
14	14.4.3	<Path.RenderTransform> Element.....	186
15	14.4.4	<Glyphs.RenderTransform> Element	187
16	14.4.5	<PathGeometry.Transform> Element.....	188
17	14.4.6	<ImageBrush.Transform> Element.....	189
18	14.4.7	<VisualBrush.Transform> Element	190
19	14.4.8	<LinearGradientBrush.Transform> Element	193
20	14.4.9	<RadialGradientBrush.Transform> Element	194
21	14.5	OpacityMask.....	197
22	14.5.1	<Canvas.OpacityMask> Element	197
23	14.5.2	<Path.OpacityMask> Element	198
24	14.5.3	<Glyphs.OpacityMask> Element.....	199
25	15. COLOR.....	201	201
26	15.1	Color Support	201
27	15.1.1	sRGB Color Space	201
28	15.1.2	scRGB Color Space.....	202
29	15.1.3	Gray Color Space.....	202
30	15.1.4	CMYK Color Space.....	202
31	15.1.5	N-Channel Color Spaces.....	202
32	15.1.6	Named Color for Spot Colors and N-tone Images.....	202
33	15.1.7	Identifying Output-Ready Color Spaces Using ICC Profiles	202
34	15.1.8	ICC Profiles	203
35	15.2	Vector Color Syntax	204
36	15.2.1	sRGB Color Syntax.....	205
37	15.2.2	scRGB Color Syntax	206
38	15.2.3	Grayscale syntax	207
39	15.2.4	CMYK Color Syntax	207
40	15.2.5	N-Channel Color Syntax.....	208
41	15.2.6	Named Color Syntax	209
42	15.3	Colors in Raster Images.....	210
43	15.3.1	sRGB Raster Images	210
44	15.3.2	scRGB Raster Images	211
45	15.3.3	Gray Raster Images	211
46	15.3.4	CMYK Raster Images	212
47	15.3.5	N-channel Raster Images.....	212
48	15.3.6	Named Color Raster Images	213
49	15.3.7	Device Color Raster Images.....	213
50	15.3.8	Images and Color Profile Association.....	214
51	15.4	Registration Marks for Color Separations.....	215
52	15.5	Alpha and Gradient Blending	216
53	15.6	Color Rendering Intent	218
54	16. DOCUMENT STRUCTURE AND INTERACTIVITY.....	223	223
55	16.1	Document Structure Markup.....	223
56	16.1.1	DocumentStructure Part	223
57	16.1.2	StoryFragments Part	229

1 **16.2 Hyperlinks..... 243**

2 16.2.1 Hyperlink Activation243

3 16.2.2 Hyperlink Addressing244

4 16.2.3 Name Attribute244

5 16.2.4 FixedPage.NavigateUri Attribute.....245

6 **16.3 Selection 246**

7 **16.4 Accessibility 246**

8 16.4.1 Reading Order246

9 16.4.2 Screen Reader Applications247

10 16.4.3 Text Alternatives for Graphics and Images247

11 **17. XPS DOCUMENT PACKAGE FEATURES 249**

12 **17.1 Interleaving Optimizations 249**

13 17.1.1 Empty PrintTicket251

14 17.1.2 Optimizing Interleaving Order.....251

15 17.1.3 Consuming Interleaved Packages254

16 17.1.4 Consumers with Resource Constraints.....254

17 17.1.5 Interleaving Optimizations and Digital Signatures256

18 **17.2 Digital Signatures..... 256**

19 17.2.1 Signature Policy257

20 17.2.2 Signature Definitions.....260

21 **17.3 Core Properties 264**

22 **18. RENDERING RULES 265**

23 **18.1 Coordinate System and Rendering Placement 265**

24 18.1.1 Page Dimensions265

25 18.1.2 Rounding of Coordinates.....265

26 18.1.3 Transforms.....266

27 18.1.4 Pixel Center Location, Pixel Placement, and Pixel Inclusion266

28 18.1.5 Maximum Placement Error267

29 18.1.6 Pixel Placement for Glyphs267

30 18.1.7 Abutment of Shapes267

31 18.1.8 Clipping Behavior267

32 **18.2 Implementation Limits 268**

33 **18.3 Gradient Computations 269**

34 18.3.1 All Gradients.....269

35 18.3.2 Linear Gradients.....271

36 18.3.3 Radial Gradients.....273

37 **18.4 Opacity Computations..... 276**

38 18.4.1 Pre-Multiplied Alpha and Superluminous Colors278

39 **18.5 Composition Rules 279**

40 18.5.1 Optimization Guidelines280

41 18.5.2 Composition Examples281

42 **18.6 Stroke Rendering..... 285**

43 18.6.1 Stroke Edge Parallelization285

44 18.6.2 Phase Control285

45 18.6.3 Symmetry of Stroke Drawing Algorithms285

46 18.6.4 Rules for Dash Cap Rendering.....286

47 18.6.5 Rules for Line Cap Rendering.....288

48 18.6.6 Line Caps for Dashed Strokes289

49 18.6.7 Rules for Line Join Rendering.....290

50 18.6.8 Rules for Degenerate Line and Curve Segments.....294

51 18.6.9 Stroking and Fill Rule295

52 18.6.10 Mixing Stroked and Non-Stroked Segments295

53 18.6.11 Stroke Behavior with Multiple Path Figures295

54 18.6.12 Consistent Nominal Stroke Width295

55 **18.7 Brushes and Images 296**

56 18.7.1 Small Tiles296

1	18.7.2	Image Scaling	296
2	18.7.3	Tile Placement.....	296
3	18.7.4	Tiling Transparent Visual Brushes and Image Brushes	297
4	19.	ELEMENTS.....	299
5	19.1	ArcSegment.....	299
6	19.2	Canvas	300
7	19.3	Canvas.Clip.....	302
8	19.4	Canvas.OpacityMask	302
9	19.5	Canvas.RenderTransform	303
10	19.6	Canvas.Resources	303
11	19.7	Discard.....	303
12	19.8	DiscardControl.....	304
13	19.9	DocumentOutline.....	304
14	19.10	DocumentReference.....	305
15	19.11	DocumentStructure.....	305
16	19.12	DocumentStructure.Outline.....	305
17	19.13	FigureStructure	306
18	19.14	FixedDocument.....	306
19	19.15	FixedDocumentSequence	306
20	19.16	FixedPage	306
21	19.17	FixedPage.Resources	308
22	19.18	Glyphs.....	308
23	19.19	Glyphs.Clip	313
24	19.20	Glyphs.Fill	313
25	19.21	Glyphs.OpacityMask.....	313
26	19.22	Glyphs.RenderTransform.....	314
27	19.23	GradientStop	314
28	19.24	ImageBrush.....	315
29	19.25	ImageBrush.Transform	316
30	19.26	Intent	317
31	19.27	LinearGradientBrush	317
32	19.28	LinearGradientBrush.GradientStops	319
33	19.29	LinearGradientBrush.Transform	319
34	19.30	LinkTarget.....	319
35	19.31	ListItemStructure	320
36	19.32	ListStructure	320
37	19.33	MatrixTransform.....	321
38	19.34	NamedElement	321
39	19.35	OutlineEntry	322
40	19.36	PageContent.....	322
41	19.37	PageContent.LinkTargets	323

1 **19.38 ParagraphStructure** 323

2 **19.39 Path** 324

3 **19.40 Path.Clip** 329

4 **19.41 Path.Data** 329

5 **19.42 Path.Fill** 329

6 **19.43 Path.OpacityMask** 329

7 **19.44 Path.RenderTransform** 330

8 **19.45 Path.Stroke** 330

9 **19.46 PathFigure** 331

10 **19.47 PathGeometry** 331

11 **19.48 PathGeometry.Transform** 332

12 **19.49 PolyBezierSegment** 333

13 **19.50 PolyLineSegment** 333

14 **19.51 PolyQuadraticBezierSegment** 334

15 **19.52 RadialGradientBrush** 334

16 **19.53 RadialGradientBrush.GradientStops** 337

17 **19.54 RadialGradientBrush.Transform** 337

18 **19.55 ResourceDictionary** 337

19 **19.56 SectionStructure** 338

20 **19.57 SignBy** 339

21 **19.58 SignatureDefinition** 339

22 **19.59 SignatureDefinitions** 340

23 **19.60 SigningLocation** 340

24 **19.61 SolidColorBrush** 340

25 **19.62 SpotLocation** 341

26 **19.63 Story** 341

27 **19.64 StoryBreak** 342

28 **19.65 StoryFragment** 342

29 **19.66 StoryFragments** 343

30 **19.67 StoryFragmentReference** 344

31 **19.68 TableCellStructure** 344

32 **19.69 TableRowGroupStructure** 345

33 **19.70 TableRowStructure** 345

34 **19.71 TableStructure** 346

35 **19.72 VisualBrush** 346

36 **19.73 VisualBrush.Transform** 348

37 **19.74 VisualBrush.Visual** 349

38 **A. SCHEMAS – W3C XML** 351

39 **A.1 Signature Definitions** 351

40 **A.2 XPS Document** 353

41 **A.3 Resource Dictionary Key** 377

1	A.4 Document Structure	379
2	A.5 Discard Control	385
3	A.6 3D-Graphic Content	386
4	B. SCHEMAS – RELAX NG	389
5	B.1 Signature Definitions	389
6	B.2 XPS Document	391
7	B.3 Resource Dictionary Key	393
8	B.4 Document Structure	395
9	B.5 Discard Control	397
10	B.6 3D-Graphic Content	398
11	C. ABBREVIATED GEOMETRY SYNTAX ALGORITHM	399
12	D. STANDARD NAMESPACES AND CONTENT TYPES	411
13	D.1 XML Namespace URIs	411
14	D.2 Content Types	412
15	D.3 Relationship Types	413
16	E. CONFORMANCE REQUIREMENTS	415
17	E.1 Implementation Conformance	416
18	E.1.1 MUST Conformance Requirements	416
19	E.1.2 SHOULD Conformance Requirements	417
20	E.2 XPS Document Format	417
21	E.2.1 MUST Conformance Requirements	417
22	E.2.2 SHOULD Conformance Requirements	417
23	E.3 Parts and Relationships	417
24	E.3.1 MUST Conformance Requirements	417
25	E.3.2 SHOULD Conformance Requirements	425
26	E.3.3 OPTIONAL Conformance Requirements	430
27	E.4 Documents	431
28	E.4.1 MUST Conformance Requirements	431
29	E.4.2 SHOULD Conformance Requirements	433
30	E.4.3 OPTIONAL Conformance Requirements	434
31	E.5 Graphics	434
32	E.5.1 MUST Conformance Requirements	434
33	E.5.2 SHOULD Conformance Requirements	435
34	E.5.3 OPTIONAL Conformance Requirements	435
35	E.6 Text	435
36	E.6.1 MUST Conformance Requirements	435
37	E.6.2 SHOULD Conformance Requirements	437
38	E.6.3 OPTIONAL Conformance Requirements	438
39	E.7 Brushes	439
40	E.7.1 MUST Conformance Requirements	439
41	E.8 Common Properties	439
42	E.8.1 MUST Conformance Requirements	439
43	E.8.2 OPTIONAL Conformance Requirements	440
44	E.9 Color	441
45	E.9.1 MUST Conformance Requirements	441
46	E.9.2 SHOULD Conformance Requirements	446
47	E.9.3 OPTIONAL Conformance Requirements	448
48	E.10 Document Structure and Interactivity	450

1 E.10.1 MUST Conformance Requirements450

2 E.10.2 SHOULD Conformance Requirements451

3 E.10.3 OPTIONAL Conformance Requirements453

4 **E.11 XPS Document Package Features 455**

5 E.11.1 MUST Conformance Requirements455

6 E.11.2 SHOULD Conformance Requirements457

7 E.11.3 OPTIONAL Conformance Requirements459

8 **E.12 Rendering Rules 460**

9 E.12.1 MUST Conformance Requirements460

10 E.12.2 SHOULD Conformance Requirements461

11 E.12.3 OPTIONAL Conformance Requirements464

12 **E.13 Additional Conformance Requirements 466**

13 E.13.1 MUST Conformance Requirements466

14 **E.14 3D Graphic Content 467**

15 E.14.1 MUST Conformance Requirements467

16 E.14.2 SHOULD Conformance Requirements468

17 E.14.3 OPTIONAL Conformance Requirements468

18 **F. 3D GRAPHIC CONTENT 469**

19 **F.1 Brush3D 472**

20 **G. BIBLIOGRAPHY 475**

21 **H. INDEX 477**

22

1 List of Figures

2	Figure 8–1. Package-based XPS Document format.....	21
3	Figure 11–1. Fill using EvenOdd algorithm.....	81
4	Figure 11–2. Fill using NonZero algorithm	81
5	Figure 11–3. Arc choice A.....	85
6	Figure 11–4. Arc choice B.....	85
7	Figure 11–5. Arc choice C.....	85
8	Figure 11–6. Arc choice D	85
9	Figure 12–1. Glyph metrics	105
10	Figure 12–2. Upright (usually horizontal) glyph metrics.....	105
11	Figure 12–3. Sideways (usually vertical) glyph metrics	106
12	Figure 17–1. A sample signature spot	263
13	Figure 18–1. Extreme curvatures and dash rendering	285
14	Figure 18–2. Flat dash caps.....	286
15	Figure 18–3. Square dash caps	287
16	Figure 18–4. Round dash caps	287
17	Figure 18–5. Triangular dash caps.....	288
18	Figure 18–6. Overlapping dash segments	288
19	Figure 18–7. Flat start line cap, flat end line cap	289
20	Figure 18–8. Square start line cap, square end line cap	289
21	Figure 18–9. Triangular start line cap, triangular end line cap	289
22	Figure 18–10. Round start line cap, round end line cap	289
23	Figure 18–11. Stroke start or end point within a dash for flat dash caps.....	289
24	Figure 18–12. Stroke start or end point within a dash for non-flat dash caps	289
25	Figure 18–13. Stroke start or end point within a gap for flat dash caps	290
26	Figure 18–14. Stroke start or end point within a gap for not-flat dash caps	290
27	Figure 18–15. Round line join with right angle	290
28	Figure 18–16. Round line join with acute angle	291
29	Figure 18–17. Round line join with obtuse angle	291
30	Figure 18–18. Beveled line join with right angle	291
31	Figure 18–19. Beveled line join with acute angle	292
32	Figure 18–20. Beveled line join with obtuse angle	292
33	Figure 18–21. Mitered line join with right angle and miter limit of 1.0.....	293
34	Figure 18–22. Mitered line join with acute angle and miter limit of 1.0.....	293
35	Figure 18–23. Mitered line join with obtuse angle and miter limit of 1.0.....	293
36	Figure 18–24. Mitered line join with right angle and miter limit of 2.0.....	294
37	Figure 18–25. Mitered line join with acute angle and miter limit of 2.0.....	294
38	Figure 18–26. Mitered line join with acute angle and miter limit of 10.0.....	294

39

1 List of Tables

2	Table 9–1. XPS Document parts	23
3	Table 9–2. Fixed payload relationships	26
4	Table 9–3. Supported JPEG APPn markers	29
5	Table 9–4. Support for ancillary PNG chunks	30
6	Table 9–5. Supported TIFF tags	30
7	Table 9–6. Supported Windows Media Photo features	34
8	Table 9–7. Guidelines for OpenType font embedding	36
9	Table 9–8. Cmap table selection	39
10	Table 11–1. Arc segment definition	84
11	Table 11–2. Commands	92
12	Table 12–3. Glyph specifications	110
13	Table 12–4. Portions of the cluster specification	111
14	Table 12–5. IsSideways and BidLevel effects on origin placement	115
15	Table 13–1. Brush types	125
16	Table 13–2. Common attributes for <ImageBrush> and <VisualBrush> elements	135
17	Table 14–1. Common property attributes	169
18	Table 14–2. Common property elements	170
19	Table 15–1. Syntax summary	204
20	Table 15–2. Color Space Pixel Format Defaults	215
21	Table 15–3. Recommended ICC rendering intent usage	220
22	Table 16–1. StoryFragments part elements	229
23	Table 16–2. Unicode character categories	245
24	Table 18–1. Recommended minimum processing requirements	268
25	Table 18–2. Opacity computation symbols	276
26	Table D–1. Package-wide namespaces	411
27	Table D–2. XPS Document namespaces	411
28	Table D–3. Package-wide content types	412
29	Table D–4. XPS Document content types	412
30	Table D–5. Package-wide relationship types	413
31	Table D–6. XPS Document relationship types	413
32	Table E–1. Implementation MUST conformance requirements	416
33	Table E–1. Implementation SHOULD conformance requirements	417
34	Table E–1. XPS Document format MUST conformance requirements	417
35	Table E–1. XPS Document format SHOULD conformance requirements	417
36	Table E–2. Parts and Relationships MUST conformance requirements	417
37	Table E–3. Parts and Relationships SHOULD conformance requirements	425
38	Table E–4. Parts and Relationships OPTIONAL conformance requirements	430
39	Table E–5. Document MUST conformance requirements	431
40	Table E–6. Document SHOULD conformance requirements	433
41	Table E–7. Document OPTIONAL conformance requirements	434
42	Table E–8. Graphics MUST conformance requirements	434
43	Table E–9. Graphics SHOULD conformance requirements	435
44	Table E–10. Graphics OPTIONAL conformance requirements	435
45	Table E–11. Text MUST conformance requirements	435
46	Table E–12. Text SHOULD conformance requirements	437
47	Table E–13. Text OPTIONAL conformance requirements	438
48	Table E–14. Brushes MUST conformance requirements	439
49	Table E–15. Common properties MUST conformance requirements	439
50	Table E–16. Common properties OPTIONAL conformance requirements	440
51	Table E–17. Color MUST conformance requirements	441
52	Table E–18. Color SHOULD conformance requirements	446

1	Table E-19. Color OPTIONAL conformance requirements.....	448
2	Table E-20. Document structure MUST conformance requirements.....	450
3	Table E-21. Document structure SHOULD conformance requirements.....	451
4	Table E-22. Document structure OPTIONAL conformance requirements	453
5	Table E-23. XPS Document package feature MUST conformance requirements	455
6	Table E-24. XPS Document package feature SHOULD conformance requirements	457
7	Table E-25. XPS Document package feature OPTIONAL conformance requirements	459
8	Table E-26. Rendering rules MUST conformance requirements.....	460
9	Table E-27. Rendering rules SHOULD conformance requirements.....	461
10	Table E-28. Rendering rules OPTIONAL conformance requirements	464
11	Table E-29. Additional MUST conformance requirements.....	466
12	Table E-30. 3D Graphic Content MUST conformance requirements	467
13	Table E-31. 3D Graphic Content SHOULD conformance requirements	468
14	Table E-32. 3D Graphic Content OPTIONAL conformance requirements	468

15

1 List of Examples

2	Example 9–1. A typical XPS Document.....	25
3	Example 9–2. XPS Document part naming.....	45
4	Example 9–3. Property attribute syntax	48
5	Example 9–4. Property element syntax	49
6	Example 10–1. <FixedDocumentSequence> usage.....	51
7	Example 10–2. <FixedDocument> usage	52
8	Example 10–3. <PageContent> usage	53
9	Example 10–4. <PageContent.LinkTargets> usage.....	54
10	Example 10–5. Fixed page markup.....	56
11	Example 10–6. Canvas composition.....	65
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.....	80
16	Example 11–5. <ArcSegment> usage.....	85
17	Example 11–6. <PolyBezierSegment> usage	87
18	Example 11–7. <PolyLineSegment> usage.....	88
19	Example 11–8. <PolyQuadraticBezierSegment> usage	90
20	Example 11–9. Closed <PathFigure> usage.....	90
21	Example 11–10. A path described using abbreviated syntax	94
22	Example 11–11. Smooth Bézier curve.....	95
23	Example 11–12. Relative commands and curves	96
24	Example 12–1. One-to-one cluster map	107
25	Example 12–2. Many-to-one cluster map	107
26	Example 12–3. One-to-many cluster map	108
27	Example 12–4. Many-to-many cluster map.....	109
28	Example 12–5. Using indices to specify advance width.....	111
29	Example 12–6. Using the Indices attribute to specify glyph replacement for a cluster	112
30	Example 12–7. Text with positive uOffset and vOffset Indices values.....	115
31	Example 12–8. Right-to-left text (odd BidiLevel)	115
32	Example 12–9. Sideways text (IsSideways set to true)	116
33	Example 12–10. Vertical text.....	117
34	Example 12–11. Japanese vertical text	117
35	Example 12–12. Using the CaretStops attribute to determine a valid caret stop position	119
36	Example 12–13. Basic italic font.....	120
37	Example 12–14. Italic font using StyleSimulations attribute.....	120
38	Example 12–15. Kerning	121
39	Example 12–16. Ligatures	121
40	Example 12–17. Cluster maps	122
41	Example 13–1. <SolidColorBrush> usage.....	127
42	Example 13–2. <ImageBrush> usage.....	129
43	Example 13–3. <VisualBrush.Visual> usage	133
44	Example 13–4. ViewboxUnits and ViewportUnits attribute usage	136
45	Example 13–5. Tiling brush base image and rendering.....	136
46	Example 13–6. Tiling brush Viewport adjustments.....	137
47	Example 13–7. Tiling brush viewbox adjustments.....	138
48	Example 13–8. Image brush with a Viewbox larger than the image	139
49	Example 13–9. Image brush with TileMode value of None	140
50	Example 13–10. Visual brush with TileMode value of None	142
51	Example 13–11. Image brush with a TileMode value of Tile	143
52	Example 13–12. Visual brush with a TileMode value of Tile.....	144

1	Example 13–13. Image brush with a TileMode value of FlipX	145
2	Example 13–14. Visual brush with a TileMode value of FlipX	146
3	Example 13–15. Image brush with a TileMode value of FlipY.....	147
4	Example 13–16. Visual Brush with a TileMode value of FlipY	148
5	Example 13–17. Image brush with a TileMode value of FlipXY.....	149
6	Example 13–18. Visual brush with a TileMode value of FlipXY	150
7	Example 13–19. <LinearGradientBrush> usage	152
8	Example 13–20. Linear gradient brush with a SpreadMethod value of Pad	153
9	Example 13–21. Linear gradient brush with a SpreadMethod value of Reflect.....	154
10	Example 13–22. Linear gradient brush with a SpreadMethod value of Repeat	155
11	Example 13–23. A radial gradient brush	159
12	Example 13–24. RadialGradientBrush usage	159
13	Example 13–25. Radial gradient brush with a SpreadMethod value of Pad	160
14	Example 13–26. Radial gradient brush with a SpreadMethod value of Reflect.....	162
15	Example 13–27. Radial gradient brush with a SpreadMethod value of Repeat	163
16	Example 13–28. Opacity mask with linear gradient.....	166
17	Example 13–29. Opacity mask with radial gradient.....	167
18	Example 14–1. <FixedPage.Resources> usage	171
19	Example 14–2. <Canvas.Resources> usage	172
20	Example 14–3. Resource dictionary markup	174
21	Example 14–4. A remote resource dictionary and reference.....	175
22	Example 14–5. Using a resource reference to fill a brush.....	176
23	Example 14–6. Using scoping rules	177
24	Example 14–7. Canvas clip markup and rendering.....	178
25	Example 14–8. <Path.Clip> usage	180
26	Example 14–9. <Glyphs.Clip> usage	181
27	Example 14–10. Matrix scaling.....	183
28	Example 14–11. Matrix reversing the x axis	183
29	Example 14–12. Matrix reversing the y axis	183
30	Example 14–13. Matrix skewing.....	183
31	Example 14–14. Matrix Rotating	183
32	Example 14–15. Matrix positioning	183
33	Example 14–16. <MatrixTransform> usage	184
34	Example 14–17. Using abbreviated matrix transformation syntax.....	185
35	Example 14–18. <Canvas.RenderTransform> usage.....	185
36	Example 14–19. <Path.RenderTransform> usage.....	186
37	Example 14–20. <Glyphs.RenderTransform> usage	187
38	Example 14–21. <PathGeometry.Transform> usage.....	188
39	Example 14–22. <ImageBrush.Transform> usage.....	189
40	Example 14–23. <VisualBrush.Transform> usage	190
41	Example 14–24. <VisualBrush.Transform> usage with tiling behavior.....	192
42	Example 14–25. <LinearGradientBrush.Transform> usage	193
43	Example 14–26. <RadialGradientBrush.Transform> usage	195
44	Example 14–27. <Canvas.OpacityMask> usage	197
45	Example 14–28. <Path.OpacityMask> usage	198
46	Example 14–29. <Glyphs.OpacityMask> usage	200
47	Example 16–1. Document structure markup	224
48	Example 16–2. Document outline markup	226
49	Example 16–3. Simple multi-story document.....	229
50	Example 16–4. Story flowing back and forth across a page boundary	229
51	Example 16–5. Content structure spanning pages	231
52	Example 16–6. StoryFragments part markup.....	236
53	Example 16–7. Story fragments markup using a fragment name.....	237

1	Example 16–8. A relative, internal, named-address hyperlink	244
2	Example 16–9. A relative internal page address hyperlink	244
3	Example 17–1. Optimized interleaving for a single-threaded parsing architecture	251
4	Example 17–2. Optimized interleaving for a multi-threaded parsing architecture	253
5	Example 17–3. A DiscardControl part.....	255
6	Example 17–4. A SignatureDefinitions part.....	260
7	Example 18–1. Path opacity behavior for overlapping path figures	281
8	Example 18–2. Opacity behavior of path stroke intersections.....	281
9	Example 18–3. Opacity behavior of paths with stroked edges	282
10	Example F–1. 3D graphics content in FixedPage.fpage	470
11	Example F–2. 3D graphics content in FixedPage.fpage	472
12		

1. Scope

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

6 A primary goal is to ensure the interoperability of independently created software and hardware
7 systems that produce or consume XPS content. This Standard defines the ~~formal~~
8 that [systems processing XPS Documents](#) ~~producers and consumers~~ must satisfy in order to
9 achieve interoperability.

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

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

2. Conformance

2.1 Requirements Terminology ~~Language Notes~~

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

- **MUST:** This word, or the adjective "REQUIRED", means that the item is an absolute requirement of the Standard.
- **SHOULD:** This word, or the adjective "RECOMMENDED", means that there might exist valid reasons in particular circumstances to ignore this item, but the full implications should be understood and the case carefully weighed before choosing a different course.
- **MAY:** This word, or the adjective "OPTIONAL", means that this item is truly optional.

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

Requirements are documented inline in this Standard, and each requirement is denoted by a letter (M – MUST; S – SHOULD; O – OPTIONAL) and a unique rule number of the form m.n, where m and n are positive integers, all enclosed in brackets ([...]).

[Example: [M1.2] is a MUST requirement, [S2.4] is a SHOULD requirement, and [O3.9] is a MAY requirement. end example]

For convenient reference, these rules are collected in §Annex A.

2.2 ~~Software~~ Implementation Conformance

This Standard includes the implementation requirements that systems processing XPS content must satisfy in order to achieve conforming interoperability. An implementation is a consumer, or a producer, or both a consumer and a producer.

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

In order for a consumer to be considered conformant, the following rules apply:

- It MUST interpret and process the contents of XPS Document instances in a manner conforming to this Standard [M0.1]. A consumer is NOT REQUIRED to interpret or process all of the content in an XPS Document instance [M0.2].
- It SHOULD instantiate an error condition when XPS Document content not conforming to this Standard is encountered [S0.1].
- It MUST NOT instantiate an error condition in response to XPS Document content conforming to this Standard [M0.3].
- When "OPTIONAL" or "RECOMMENDED" features contained within XPS Document instances are accessed by a consumer, the consumer MUST interpret and process those features in a manner conforming to this Standard [M0.4].

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

In order for a producer to be considered conformant, the following rules apply:

- Any XPS Document instances it creates MUST conform to this Standard [M0.5].
- It MUST NOT introduce any non-conforming XPS Document content when modifying an XPS Document instance [M0.6].
- When a producer chooses to use an "OPTIONAL" or "RECOMMENDED" feature in an XPS Document instance, then the producer MUST create or modify that feature in a manner conforming to this Standard [M0.7].
- ~~• It MUST NOT generate any new, non-conforming instances of a documented format.~~
- ~~• It MUST NOT introduce any non-conformance when modifying an instance of a documented format.~~

~~An application can be both a consumer and a producer.~~

~~Conformance requirements are documented inline in this specification, and each requirement is denoted by a letter (M—MUST; S—SHOULD; O—OPTIONAL) and a rule number of the form m.n, where m and n are positive integers, all enclosed in brackets ([...]). [Example: [M1.2] is a MUST requirement, [S2.4] is a SHOULD requirement, and [O3.9] is a MAY requirement. end example] For convenient reference, these rules are collected in §F.~~

2.3 Instantiating Error Conditions

XPS Documents are intended to address the requirements of a wide range of scenarios. The methods and effects of instantiated error conditions in response to conformance rule violations are implementation-defined.

[Note: Implementors are encouraged to instantiate error conditions to indicate non-conformant XPS Documents where users can be expected to be able to act on the error information. Implementors are strongly encouraged to fail gracefully when processing non-compliant XPS Documents to ensure that non-compliant XPS Document instances, and non-compliant XPS producers, do not proliferate. end note]

3. Normative References

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

Adobe Photoshop® TIFF Technical Notes, March 22, 2002.

<http://partners.adobe.com/public/developer/en/tiff/TIFFphotoshop.pdf>

BNF of Generic URI Syntax. World Wide Web Consortium.

http://www.w3.org/Addressing/URL/5_URI_BNF.html

Digital Compression and Coding of Continuous-tone Still Images. International

Telecommunication Union (ITU). 1993. <http://www.w3.org/Graphics/JPEG/itu-t81.pdf>

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

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

Exchangeable Image File Format for Digital Still Cameras: Exif Version 2.2. Technical Standardization Committee on AV & IT Storage Systems and Equipment. Japan Electronic

Industry Development Association. 2002. <http://www.jeita.or.jp>

Extensible Markup Language (XML) 1.0 (Fourth Edition). Bray, Tim, Eve Maler, Jean Paoli, C. M. Sperberg-McQueen, and François Yergeau (editors). World Wide Web Consortium. 2006.

<http://www.w3.org/TR/2006/REC-xml-20060816/>

HTML 4.01 Specification. Jacobs, Ian, Arnaud Le Hors, and Dave Raggett (editors). World Wide

Web Consortium. 1999. <http://www.w3.org/TR/1999/REC-html401-19991224/>

ICC.1:2001-04 *File Format for Color Profiles*. International Color Consortium. 2001.

http://www.color.org/ICC_Minor_Revision_for_Web.pdf

IEC 61966:1999, *Multimedia systems and equipment - Colour measurement and management - Part 2-1: Colour management - Default RGB colour space - sRGB*

IEC 61966:2003, *Multimedia systems and equipment - Colour measurement and management - Part 2-2: Colour management - Extended RGB colour space - scRGB*

ISO 15076-1, *Image technology colour management — Architecture, profile format, and data structure — Part 1: Based on ICC.1:2004-10*

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

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

- 1 ISO/IEC FCD 19775-1r1:200x *Information technology — Computer graphics and image*
2 *processing — Extensible 3D (X3D) — Part 1: Architecture and base components.*
- 3 ISO/IEC 19776-1:2005 *Information technology — Computer graphics and image processing —*
4 *Extensible 3D (X3D) encodings — Part 1: XML encoding.*
- 5 ISO/IEC 19776-2:2005 *Information technology — Computer graphics and image processing —*
6 *Extensible 3D (X3D) encodings — Part 2: Classic VRML encoding.*
- 7 ISO/IEC 19776-3:2007 *Information technology — Computer graphics, image processing and*
8 *environmental representation — Extensible (X3D) encodings — Part 3: Compressed binary*
9 *encoding.*
- 10 *JPEG File Interchange Format, Version 1.02.* Hamilton, Eric. World Wide Web Consortium. 1992.
11 <http://www.w3.org/Graphics/JPEG/jfif3.pdf>
- 12 *Namespaces in XML 1.0 (Second Edition).* Bray, Tim, Dave Hollander, Andrew Layman, and
13 Richard Tobin (editors). World Wide Web Consortium. 2006. [http://www.w3.org/TR/2006/REC-](http://www.w3.org/TR/2006/REC-xml-names-20060816/)
14 [xml-names-20060816/](http://www.w3.org/TR/2006/REC-xml-names-20060816/)
- 15 *Portable Network Graphics (PNG) Specification.* Duce, David (editor). Second Edition. World
16 Wide Web Consortium. 2003. <http://www.w3.org/TR/2003/REC-PNG-20031110>
- 17 RFC 2045, *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message*
18 *Bodies.* Borenstein, N., and N. Freed. The Internet Society. 1996.
19 <http://www.ietf.org/rfc/rfc2045.txt>.
- 20 RFC 2119 — *Key words for use in RFCs to Indicate Requirement Levels.* Bradner, S. The
21 Internet Society. 1997. <http://www.rfc-editor.org>
- 22 RFC 3066 — *Tags for the Identification of Languages.* Alvestrand, H. The Internet Society. 2001.
23 <http://www.rfc-editor.org>
- 24 RFC 4234 — *Augmented BNF for Syntax Specifications: ABNF.* Crocker, D. (editor). The Internet
25 Society. 2005. <http://www.rfc-editor.org>
- 26 *A Standard Default Color Space for the Internet—sRGB, Version 1.10.* Anderson, Matthew,
27 Srinivasan Chandrasekar, Ricardo Motta, and Michael Stokes. World Wide Web Consortium.
28 1996. <http://www.w3.org/Graphics/Color/sRGB>
- 29 *TIFF, Revision 6.0.* Adobe Systems Incorporated. 1992.
30 <http://partners.adobe.com/public/developer/en/tiff/TIFF6.pdf>
- 31 *Unicode Character Database, Revision 4.0.0.* Davis, Mark and Ken Whistler. The Unicode
32 Consortium. 2003. <http://www.unicode.org/Public/4.0-Update/UCD-4.0.0.html>
- 33 *The Unicode Standard, Version 4.0.* The Unicode Consortium. Boston, MA: Addison-Wesley,
34 2003, ISBN 0-321-18578-1.
- 35 *XML Base.* Marsh, Jonathan. World Wide Web Consortium. 2001.
36 <http://www.w3.org/TR/2001/REC-xmlbase-20010627/>
- 37 *XML Schema Part 1: Structures, Second Edition.* Beech, David, Murray Maloney, Noah
38 Mendelsohn, and Henry S. Thompson (editors). World Wide Web Consortium. 2004.
39 <http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/>

- 1 *XML Schema Part 2: Datatypes, Second Edition*. Biron, Paul V. and Ashok Malhotra (editors).
- 2 World Wide Web Consortium. 2004. <http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/>

4. Definitions

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

alpha blending — Transparently blending two elements when rendering.

consumer — A piece of software or a device that [interprets and processes](#) ~~reads~~ XPS packages.

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

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

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

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

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

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

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

digital signature, noncompliant — A digital signature that does not conform to the XPS Document signing rules.

digital signature, questionable — A digital signature that conforms to the XPS Document signing rules but has a problem during validation of the signature such as the inability to contact the certificate authority to validate its authenticity or the markup contains markup compatibility elements and attributes that can change the representation of the signed content.

digital signature, valid — A digital signature that conforms to the XPS Document signing rules and is not a broken digital signature or questionable digital signature.

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

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

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

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

- 1 **fixed payload root** — The root of a fixed payload is the FixedDocumentSequence part.
- 2 **FixedDocument part** — A common, easily indexed root for all pages within an XPS Document.
- 3 **FixedDocumentSequence part** — The part that assembles a set of FixedDocument parts
4 within the fixed payload.
- 5 **FixedPage part** — The part that contains all of the visual elements to be rendered on a page.
- 6 [implementation-defined behavior — Behavior specified by each implementation and not by](#)
7 [this Standard.](#)
- 8 **named color** — An industry-defined color specification that identifies a particular color in a
9 well-defined color [schema](#)[system](#), usually for purposes of printing.
- 10 **named element** — An element in the document structure markup that refers to an element in
11 the fixed-page markup with a specified name.
- 12 **ordering, interleaved**— The layout style of a physical package where parts are broken into
13 pieces and “mixed-in” with pieces from other parts. When delivered, interleaved packages help
14 improve the performance of the consumer processing the package.
- 15 **ordering, simple** — Simple ordering the parts in the package are laid out with a defined
16 ordering. When such a package is delivered in a purely linear fashion, starting with the first
17 byte in the package through to the last that, all of the bytes for the first part arrive first, then
18 all of the bytes for the second part, and so on.
- 19 **package** — A logical entity that holds a collection of parts.
- 20 **package model** — Defines a package abstraction that holds a collection of parts.
- 21 **package relationship** — A relationship whose target is a part and whose source is the
22 package as a whole. Package relationships are found in the package relationships part named
23 “/_rels/.rels”.
- 24 **part** — A stream of bytes with a MIME content type and associated common properties.
25 Typically corresponds to a file (as on a file system), a stream (as in a compound file), or a
26 resource (as in an HTTP URI).
- 27 **part name** — A part name is used to refer to a part in the context of a package, typically as
28 part of a URI. By definition, the part name is the path component of a pack URI.
- 29 **payload** — A complete collection of interdependent parts and relationships within a package.
- 30 **physical imageable size** — Represents the area of a page that is printable by a specific
31 device.
- 32 **physical media size** — Represents the physical media on which the content will be printed.
- 33 **physical model** — Defines the mapping between the components of the package model to the
34 features of a particular physical format based on the ZIP specification.
- 35 **piece** — A portion of a part. Pieces of different parts can be interleaved together. The individual
36 pieces are named using a unique mapping from the part name. Pieces are not addressable in
37 the package model.
- 38 **primary fixed payload root** — The fixed payload root that is referenced by the XPS package
39 StartPart relationship.

- 1 **PrintTicket part** — A PrintTicket part provides the settings used when a package is printed.
2 PrintTicket parts can be attached to the entire package, or at lower levels in the structure, such
3 as individual pages.
- 4 **producer** — A piece of software or a device that [creates or modifies](#) ~~writes~~ XPS packages.
- 5 **producer bleed size** — Represents the overflow (or “bleed”) box used by the producer for
6 registration and layout.
- 7 **producer content size** — Represents the content bounding box specified by the producer.
- 8 **producer media size** — Represents the physical media on which the content will be printed.
- 9 **property** — A characteristic of a markup element, referred to as an attribute of the element.
- 10 **property attribute** — An XPS Document property value can be expressed as either a property
11 attribute or a property element.
- 12 **property element** — An XPS Document property value can be expressed as either a property
13 attribute or a property element.
- 14 **property value** — The value of a property, expressed as an XML attribute, an XML child
15 element, or an entry in the resource dictionary.
- 16 **relationships** — A relationship represents the kind of connection between a source part and a
17 target part in a package. Relationships make the connections between parts directly
18 discoverable without looking at the content in the parts, and without altering the parts
19 themselves. See also, package relationship.
- 20 **relationships part** — A part containing an XML representation of relationships.
- 21 **required part** — A part, such as an image or font, that is referenced from other parts, and is
22 required for valid processing of the referencing part.
- 23 **resource definition** — A shareable property value, with a name, defined within a resource
24 dictionary. Any property value defined by fixed page markup can be held in a resource
25 dictionary. Each resource definition has a key that is unique within the scope of the resource
26 dictionary.
- 27 **resource dictionary** — A resource dictionary holds resources. Each resource in a resource
28 dictionary carries a name. The resource’s name can be used to reference the resource from a
29 property’s XML attribute.
- 30 **resource dictionary, remote** — A part containing a resource dictionary.
- 31 **resource reference** — An attribute whose value refers to an entry in a resource dictionary.
32 Resource references appear in the format “{StaticResource RscName}” where RscName
33 corresponds to a matching entry in the resource dictionary with an x:Key attribute value.
- 34 **signature definition** — The means by which XPS Document authors provide co-signature
35 requirements and workflow-specific signature information.
- 36 **signature spot** — A visual element that indicates that a digital signature has been applied or
37 requested.
- 38 **signing rules** — The set of rules that define whether a particular digital signature is compliant
39 with the XPS Document **signature** policy.

- 1 **story** — A block of individually readable content in an XPS Document.
- 2 **story fragment** — A portion of a story that appears within the scope of a single fixed page.
- 3 **stream** — A linearly ordered sequence of bytes.
- 4 **thumbnail** — An images that helps end-users identify parts of a package or a package as a
5 whole.
- 6 **X3D** — A 3D graphic content stream conforming to ISO standards 19775-1r1:200x, 19776-
7 1:2005, 19776-2:2005, and 19776-3:2007.
- 8 **XPS Document** — A package that contains a discoverable fixed payload and is a format for
9 storing paginated documents defined by the XPS Standard.
- 10 **XPS Document StartPart relationship** — The specific relationship type that identifies the
11 root of a fixed payload within an XPS Document.
- 12 **ZIP Archive** — A physical ZIP file that is displayed by the file system. A ZIP archive contains
13 **ZIP items**.

5. Notational Conventions

5.1 Document Conventions

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

Definition terms are formatted like *this*.



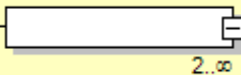
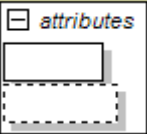


Syntax descriptions and code are formatted in `monospace` type.

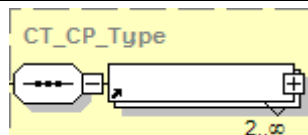
Replaceable items are formatted in *monospace cursive* type.

5.2 Diagrams

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

12

Symbol	Description
	Required element. This box represents an element that MUST 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 The following acronyms and abbreviations are used throughout this Standard:

3 IEC — the International Electrotechnical Commission

4 ISO — the International Organization for Standardization

5 W3C — World Wide Web Consortium

1 **7. General Description**

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

5 This Standard 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 XML-based
8 XPS Document format. These clauses describe the XML markup that defines the
9 composition of documents and the appearance of each page. They also include rendering
10 rules that enable devices and applications to display and print XPS Documents with full
11 fidelity in a wide range of environments and scenarios.
- 12 3. XPS Document Markup Reference (clause 19), which presents a consolidated reference of
13 XPS Document markup elements and their attributes.
- 14 4. Annexes (**Error! Reference source not found.**–0), which contain additional technical
15 details and schemas, as well as convenient reference information.

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

20 Clauses 1–5 and 7–19, and annexes **Error! Reference source not found.**–A and A–0, form a
21 normative part of this Standard; and the clause 6, annex A, examples, notes, and the index,
22 are informative.

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

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

8. XPS Document Format

This Standard describes how the XPS Document format is organized internally and rendered externally. It is built upon the principles described in the OPC Standard. XPS Documents MUST observe all conformance requirements [M1.1] and **SHOULD observe all** recommendations [S1.1] of that Standard, except where indicated otherwise [M1.1]. **The information presented here is intended both for producers and consumers.**~~The information presented here is intended both for producers, which emit content in the XPS Document format, and consumers, which access and render or process the contents of an XPS Document.~~

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

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

Finally, the XPS Document format implements the common package features specified by the OPC Standard that support digital signatures and core properties. Implementers should note that the XPS Document format does not define support for encryption, or other forms of content protection, other than that required for Embedded Font Obfuscation.

8.1 How This Standard Is Organized

This subclause is informative

Clause	Description
Parts and Relationships (§9)	This clause describes how XPS Documents use the packaging model (as described in the OPC Standard) 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.

Clause	Description
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.
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 can 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. [<i>Example: Such markings might be tables or paragraphs. 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 Standard) 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.1)	This annex includes the W3C XSD schema for the Signature Definitions part.
XPS Document Schema (§A.2A.1)	This annex includes the W3C XSD schema for the FixedDocument, FixedDocumentSequence, and FixedPage parts.
Resource Dictionary Key Schema (§A.3A.1)	This annex provides the W3C XSD schema for the resource dictionary Key attribute, used by several elements in the XPS Document schema.
Document Structure Schema (§A.4A.1)	This annex provides the W3C XSD schema for the DocumentStructure and StoryFragments parts.
Discard Control Schema (§A.5A.1)	This annex includes the W3C XSD schema for the DiscardControl part for interleaving.

Clause	Description
3D-Graphic Content Schema (§A.6)	This annex includes the optional W3C XSD schema for 3D-Graphics support.
Abbreviated Geometry Syntax Algorithm (§B)	This annex provides a sample algorithm for interpreting the abbreviated geometry syntax provided to succinctly describe geometric regions in a single attribute.
scRGB Gamut Boundary Definition (§D)	The scRGB color space’s gamut boundary used by this Standard is defined in this annex.
Standard Namespaces and Content Types (§A)	This annex defines all of the XML namespace names, content types, and relationship types used by all XPS Document parts and relationships.
Conformance Requirements (§A)	This annex assembles all the conformance requirements specified throughout the previous clauses and annexes into a comprehensive list for reference purposes.

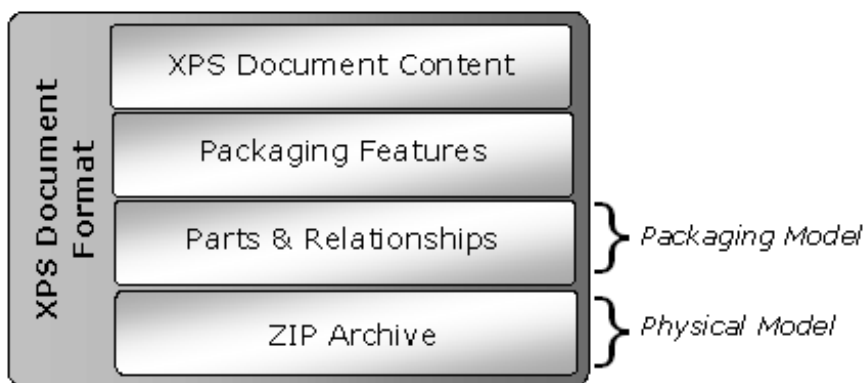
1 **End of informative text**

2 **8.2 Package**

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

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

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



10

9. Parts and Relationships

The packaging conventions described in the OPC Standard can be used to carry any payload. A *payload* is a complete collection of interdependent parts and relationships within a package. This Standard defines a particular payload that contains a static or “fixed-layout” representation of paginated content: the fixed payload.

A package that holds at least one fixed payload and follows the rules described in this Standard is referred to as an *XPS Document*. Producers and consumers of XPS Documents can implement their own parsers and rendering engines based on this Standard.

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

9.1 Fixed Payload

A payload that has a FixedDocumentSequence root part is known as a *fixed payload*. A *fixed payload root* is a FixedDocumentSequence part that references FixedDocument parts that, in turn, reference FixedPage parts.

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

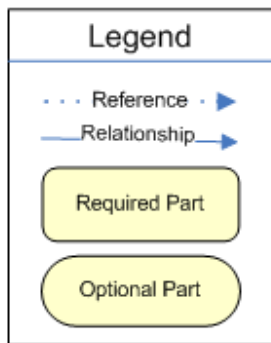
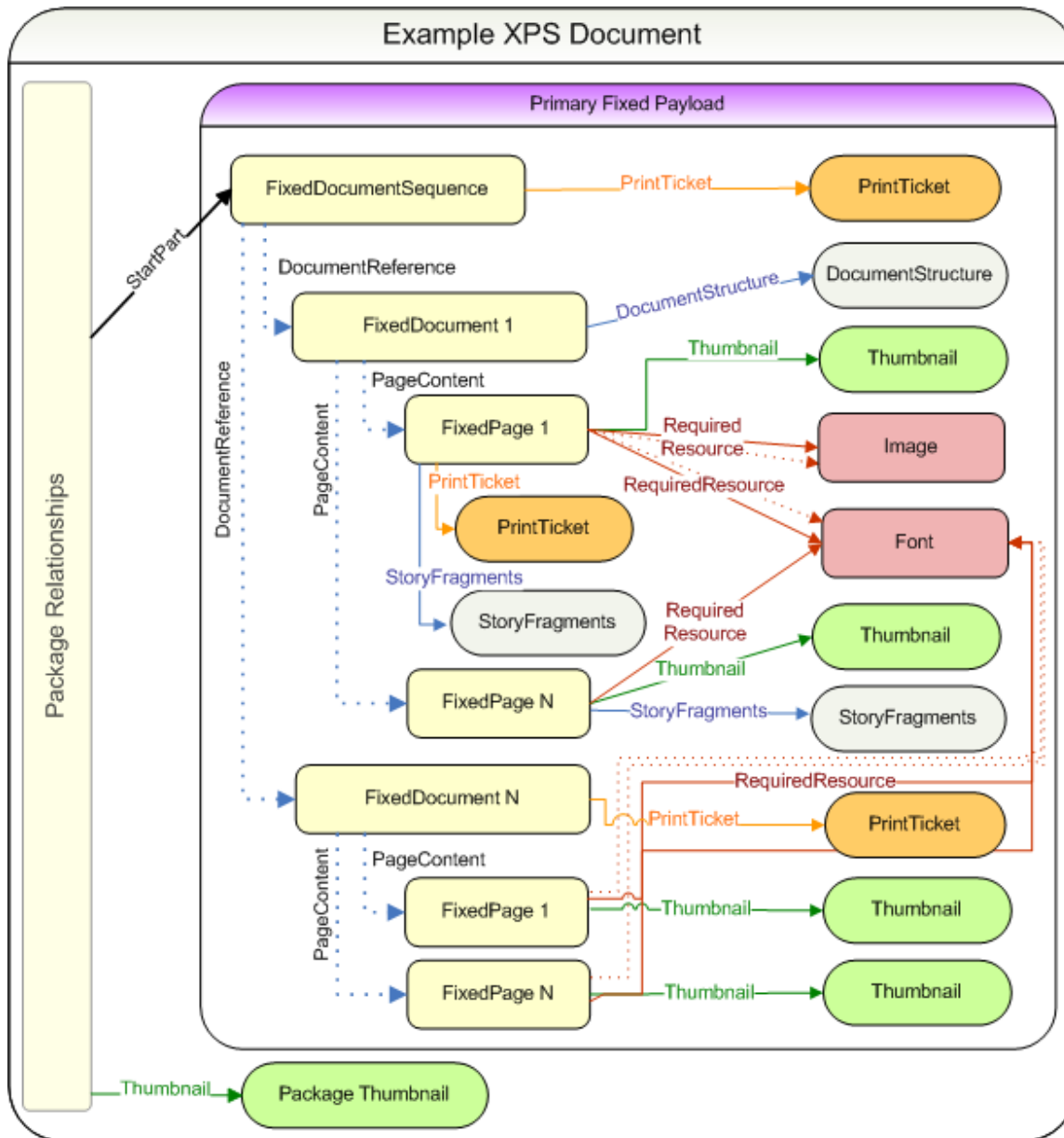
The payload includes the full set of parts required for processing the FixedDocumentSequence part. All content to be rendered **MUST** be contained in the XPS Document [M2.1]. The payload for an XPS Document may include additional parts not defined by this Standard. Consumers **MUST** ignore parts in valid XPS Documents that they do not understand [M2.84]. The parts that can be found in an XPS Document are listed in Table 9–1. Relationships and content types for these parts are defined in §A. Each part **MUST** use *only* the appropriate content type specified in §A [M2.2].

Table 9–1. XPS Document parts

Name	Description	Required/Optional
FixedDocumentSequence (§9.1.2)	Specifies a sequence of fixed documents.	REQUIRED [M2.3]
FixedDocument (§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 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]

9.1.1 Fixed Payload Relationships

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

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

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

Relationship types are defined in §A.

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 part to a PrintTicket part.	OPTIONAL [O2.15]
Required Resource	Relationship from a FixedPage part to a required resource, including	REQUIRED for each resource referenced

	Font, Image, ColorProfile, and Remote Resource Dictionary parts. Required resources can be shared between pages.	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 Document for
 2 any of the following relationship types: DiscardControl, DocumentStructure, PrintTicket,
 3 Required Resource, Restricted Font, StartPart, StoryFragments, and Thumbnail. Consumers
 4 that access the target part of any relationship with one of these relationship types MUST
 5 [instantiate an error condition](#) ~~generate an error~~ if the part is not included in the XPS Document
 6 [M2.77].

7 **9.1.2 FixedDocumentSequence Part**

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

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

15 Fixed document sequence markup specifies each fixed document in the fixed payload in
 16 sequence, using <DocumentReference> elements. The order of <DocumentReference>
 17 elements determines document order and MUST be preserved ~~by consumers that are also~~
 18 ~~producers~~ [M2.15]. Each <DocumentReference> element MUST reference a FixedDocument
 19 part by relative URI [M2.80]. For more information, see §10.1.

20 The content type of the FixedDocumentSequence part is defined in §A.

21 **9.1.3 FixedDocument Part**

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

1 The markup in the FixedDocument part specifies the pages of a document in sequence using
2 <PageContent> elements. The order of <PageContent> elements determines page order and
3 MUST be preserved ~~by consumers that are also producers~~ [M2.16]. Each <PageContent>
4 element MUST reference a FixedPage part by relative URI [M2.81]. For more information,
5 see §10.2.

6 The content type of the FixedDocument part is defined in §A,

7 **9.1.4 FixedPage Part**

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

13 Each FixedPage part specifies the contents of a page within a <FixedPage> element using
14 <Path> and <Glyphs> elements (using various brush elements) and the <Canvas> grouping
15 element. The <ImageBrush> and <Glyphs> elements (or their child or descendant elements)
16 can reference Image parts or Font parts by URI. They MUST reference these parts by relative
17 URI [M2.82]. For more information, see §10.3.

18 The content type of the FixedPage part is defined in §A

19 **9.1.5 Image Parts**

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

23 Images are included in XPS Documents with an <ImageBrush> element and an ImageSource
24 attribute to reference a part with the appropriate content type. For more information, see §D.2.
25 Fixed pages MUST use a Required Resource relationship to each Image part referenced
26 [M2.10]. For more information, see §D.3.

27 XPS Documents support the following image formats:

- 28 • JPEG
- 29 • PNG
- 30 • TIFF
- 31 • Windows Media Photo

32 Color profiles MAY be embedded in image files [O2.18]. See §15.

33 [For images that have a constant opacity, producers SHOULD NOT use the image format alpha](#)
34 [channel; the Opacity attribute in the <ImageBrush> element SHOULD be used instead \[S2.37\].](#)

35 **9.1.5.1 JPEG Images**

36 It is RECOMMENDED that JPEG image part names end with the extension “.jpg” [S2.6]. JPEG
37 image parts MUST contain images that conform to the JPEG Standard [M2.17]. Consumers
38 SHOULD support JPEG images that contain JFIF-specified APP0 and ICC-specified APP2 markers
39 [S2.34]. Consumers MUST support JPEG images that contain the EXIF-specified APP1 marker
40 and interpret the EXIF color space correctly [M2.78].

1 *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

2 Consumers MUST ensure that they can distinguish between the uses of those markers listed in
3 Table 9–3 and other data that is recorded using the same markers [M2.85].

4 *[Note: The APP0 marker is also used for JFXX (extended JFIF). The APP1 marker is also used*
5 *for XMP metadata. The APP2 marker is also used for EXIF FlashPix extensions. These are not*
6 *intended to be exhaustive lists of alternative uses of those markers. end note]*

7 *[Note: Implementers of consumers might wish to support additional APPn markers, such as*
8 *APP13 (Photoshop 3.0 extension) and APP14 (Adobe DCT Filters in PostScript Level 2*
9 *extension). end note]*

10 In cases where a consumer encounters a JPEG image with conflicting resolution information in
11 different markers, the order of precedence is as follows:

- 12 1. The EXIF tag
- 13 2. The JFIF tag
- 14 3. Any other APPn tags supported by the consumer
- 15 4. A default value of 96 [dots per inch \(dpi\)](#) (as described in §13.4.1)

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

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

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

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

27 If no color profile is embedded in the JPEG image or stored in a separate part associated with
28 the JPEG image according to the mechanisms described in §15.3.8, then the default color space
29 MUST be treated as defined in §15.3.8 [M8.30].

30 **9.1.5.2 PNG Images**

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

1 *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]

2 If no color profile is embedded in the PNG image or stored in a separate part associated with
 3 the PNG image according to the mechanisms described in §15.3.8, then the default color space
 4 MUST be treated as defined in §15.3.8 [M8.30].

5 **9.1.5.3 TIFF Images**

6 It is RECOMMENDED that TIFF image part names end with the extension “.tif” [S2.9]. TIFF
 7 image parts MUST contain images that conform to the TIFF specification [M2.25]. XPS
 8 Document consumers MUST support baseline TIFF 6.0 with some extensions, as noted in Table
 9 9-5 [M2.26]. These tags MUST be supported for the specified image types [M2.26]. If
 10 consumers encounter a tag that is not included below, they SHOULD ignore that tag [S2.10].

11 *Table 9-5. Supported TIFF tags*

Image type	Tags
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, 7, or 32773) ImageLength ImageWidth ResolutionUnit (1, 2, or 3) RowsPerStrip StripByteCounts StripOffsets XResolution YResolution
Palette color	BitsPerSample (1, 4, or 8)

images	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; <i>or if SamplesPerPixel = 4: 8,8,8,8 or 16,16,16,16</i>) Compression (1, 5, 7, or 32773) ExtraSamples (0, 1, or 2. Required if SamplesPerPixel = 4; must not be present otherwise) ICC Color Profile [tag 34675] ImageLength ImageWidth PhotometricInterpretation (2) PlanarConfiguration (1) ResolutionUnit (1, 2, or 3) RowsPerStrip SamplesPerPixel (3 or 4) StripByteCounts StripOffsets XResolution YResolution

CMYK images (TIFF extension)	BitsPerSample (8,8,8,8 or 16,16,16,16; <i>or</i> if SamplesPerPixel = 5: 8,8,8,8,8 or 16,16,16,16,16) Compression (1, 5, 7, or 32773) ExtraSamples (0, 1, or 2. Required if SamplesPerPixel = 5; must not be present otherwise) ICC Color Profile [tag 34675] ImageLength ImageWidth InkSet (1) NumberOfInks (4) PhotometricInterpretation (5) PlanarConfiguration (1) ResolutionUnit (1, 2, or 3) RowsPerStrip SamplesPerPixel (4 or 5) StripByteCounts StripOffsets XResolution YResolution
------------------------------------	---

1 If the TIFF image contains multiple image file directories (IFDs), consumers MUST use only the
 2 first IFD and ignore all others [M2.27].

3 If the ResolutionUnit tag is set to 1 (no units), XResolution and YResolution are interpreted in
 4 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 associated with
 6 the TIFF image according to the mechanisms described in §15.3.8, then the default color space
 7 MUST be treated as defined in §15.3.8 [M8.30].

8 The following features of the TIFF specification MUST be supported in addition to the tags
 9 described in Table 9–5:

- 10 • Baseline TIFF (Sections 1–10) with the exception of the following tags [M2.26]:
 - 11 ○ CellLength
 - 12 ○ CellWidth
 - 13 ○ GrayResponseCurve
 - 14 ○ GrayResponseUnit
 - 15 ○ MaxSampleValue
 - 16 ○ MinSampleValue
 - 17 ○ Orientation
 - 18 ○ Thresholding
- 19 • CCITT bilevel encodings (Section 11) [M2.28]
- 20 • CMYK images (Section 16) [M2.29]

- 1 • Associated alpha data (Section 18) [M2.30]
- 2 ◦ ExtraSamples tag value of 0: The data in this channel MUST be ignored [M2.83]
- 3 ◦ ExtraSamples tag value of 1: Treat alpha as pre-multiplied alpha (see §18.4.1 for
- 4 details)
- 5 ◦ 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 ◦ 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 space MUST
- 12 be interpreted correctly [M2.79].

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

17 XPS Document consumers SHOULD mitigate the effect of badly formed TIFF files in the
18 following ways [S2.11]:

- 19 • Test with as many different TIFF images as possible.
- 20 • Correct common mistakes in TIFF images, such as:
 - 21 ◦ Not all BitsPerSample hold the same value
 - 22 ◦ Number of BitsPerSample does not match SamplesPerPixel
 - 23 ◦ PhotometricInterpretation 1 or 2 (instead of 3) used when BitsPerSample is set to
 - 24 "8,8,8"
 - 25 ◦ When the ExtraSamples tag is missing and SamplesPerPixel is not consistent with the
 - 26 PhotometricInterpretation tag then ExtraSamples values should be given the value 0.
- 27 • Implement a recovery strategy when a problematic TIFF image is encountered.

28 [*Note*: Over time, TIFF-consuming implementations have developed a certain tolerance for such
29 deviations by attempting to deduce the intent of the TIFF image author and correct for
30 apparent errors or deviations.

31 Many TIFF images in circulation today deviate from the TIFF Specification. *end note*]

32 **9.1.5.4 Windows Media Photo Images**

33 It is RECOMMENDED that Windows Media Photo image part names end with the extension
34 ".wdp" [S2.12]. Windows Media Photo image files MUST conform to the Windows Media Photo
35 specification [M2.35]. XPS Documents support Windows Media Photo images with the
36 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*		
	32-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 separate part
4 associated with the Windows Media Photo image according to the mechanisms described
5 in §15.3.8, then the default color space MUST be treated as defined in §15.3.8 [M8.30].

6 **9.1.6 Thumbnail Parts**

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

9 Thumbnail images MAY be attached using a relationship to the FixedPage parts [O2.19]. Each
10 FixedPage part MUST NOT have more than one thumbnail part attached [M2.36]. Relationships
11 to thumbnail parts are defined in §A. It is RECOMMENDED that if thumbnails are used for
12 pages, a thumbnail SHOULD be included for each page in the document [S2.13].

13 Although the OPC Standard allows thumbnails to be attached to any part, XPS Document
14 consumers SHOULD only process thumbnails associated via a package relationship from the
15 package as a whole or via a relationship from a FixedPage part [S2.14]. These thumbnails
16 MUST be in either JPEG or PNG format [M2.37]. Thumbnails attached to any other part SHOULD
17 be ignored by XPS Document consumers [S2.14]. The content types of thumbnail parts are
18 specified in §D.2.

19 For more information about the relationship type for thumbnail parts, see §D.3.

1 **9.1.7 Font Parts**

2 Fonts are stored in font parts. XPS Documents **MUST** support the OpenType font format, which
3 includes TrueType and CFF fonts [M2.39]. To support portability, Unicode-encoded fonts
4 **SHOULD** be used (see §9.1.7.5 for additional information) [S2.15].

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

8 If the referenced font part is a TrueType Collection, the fragment portion of the URI indicates
9 the font face to be used. The use of URI fragments is specified in the BNF of Generic URI
10 Syntax specification. The fragment contained in the FontURI attribute value **MUST** be an integer
11 between 0 and n-1, inclusive, where n is the number of font faces contained in the TrueType
12 Collection [M2.38]. The syntax for the integer value is expressed as:

13
14 fontface = *DIGIT

15 [*Example: To reference the first font face in the font part “../Resources/Fonts/CJKSuper.ttc”,*
16 *the value of the FontUri attribute is “../Resources/Fonts/CJKSuper.ttc#0”. end example*] If no
17 fragment is specified, the first font face is used in the same way as if the URI had specified
18 “#0”. If the fragment is not recognised as a valid integer, consumers **SHOULD generate**
19 [instantiate](#) an error [condition](#) [S2.35].

20 Content types for fonts differ depending on whether the font is non-obfuscated or obfuscated
21 (see §9.1.7.2). Content types are summarized in §A.

22 Fixed pages **MUST** use a Required Resource relationship to each Font parts referenced [M2.10].
23 For more information, see §A.

24 **9.1.7.1 Subsetting Fonts**

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

30 **9.1.7.2 OpenType Font Embedding**

31 Protecting the intellectual property of font vendors is a goal of the XPS Document format.
32 Therefore, producers **MUST** observe the guidelines and mechanisms described below in order to
33 honor the licensing rights specified in OpenType fonts [M2.40]. It is not the responsibility of
34 consumers to enforce font licensing intent, although consumers **MUST** be able to process XPS
35 Documents using any combination of these embedding and obfuscation mechanisms, even if
36 produced in violation of these guidelines [M2.41].

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

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

1 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 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 AutomationProperties.Name attribute of the <Path> element [S2.18].	Render embedded images.
2 / 0x0004	For preview and print embedding, font can 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 and §D.3 for details. –MUST NOT modify or edit the XPS Document markup or hierarchical structure starting from the <FixedDocument> element [M2.43].	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. [M2.49]	MUST NOT extract or install permanently [M2.50].
9 / 0x0200	Bitmap embedding only.	MUST do embedded font obfuscation [M2.51]	MUST NOT extract or install permanently

	(see §9.1.7.3). [M2.52]. 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].
--	--

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 embedded
3 fonts. Specifically, embedded font obfuscation prevents end-users from using standard ZIP
4 utilities to extract fonts from XPS Document files and install them on their systems.

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

6 Embedded font obfuscation achieves the following goals:

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

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

15 Although the licensing intent allows embedding of non-obfuscated fonts and installation of the
16 font on a remote client system under certain conditions, this is NOT RECOMMENDED in XPS
17 Documents [S2.19]. However, there are vertical solutions in which implementations might
18 benefit from un-obfuscated font embedding. In these cases, implementations could omit
19 obfuscation or extract and install the embedded font.

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

- 22 1. Generate a 128-bit GUID (Globally Unique Identifier) for the font to be obfuscated.
23 Instead of a true GUID, a 128-bit random number MAY be used [O2.21]. The 16 bytes of
24 the 128-bit GUID are referred to in the following text by the placeholder names B_{00} , B_{01} ,
25 B_{02} , B_{03} ; B_{10} , B_{11} ; B_{20} , B_{21} ; B_{30} , B_{31} , B_{32} , B_{33} , B_{34} , B_{35} , B_{36} , and B_{37} . The order in which
26 bytes are assigned to these placeholders does not matter, as long as it is consistent for
27 obfuscation and de-obfuscation.
- 28 2. Generate a part name for the obfuscated font using the GUID. The last segment of the
29 part name MUST be of the form " $B_{03}B_{02}B_{01}B_{00}-B_{11}B_{10}-B_{21}B_{20}-B_{30}B_{31}-B_{32}B_{33}B_{34}B_{35}B_{36}B_{37}$ " or
30 " $B_{03}B_{02}B_{01}B_{00}-B_{11}B_{10}-B_{21}B_{20}-B_{30}B_{31}-B_{32}B_{33}B_{34}B_{35}B_{36}B_{37}.ext$ " where each B_x represents a
31 placeholder for one byte of the GUID, represented as two hex digits [M2.54]. The part
32 name MAY have an arbitrary extension (identified by the placeholder ".ext") [O2.22]. It is
33 RECOMMENDED that the extension for TrueType fonts be ".odttf" and for TrueType
34 collections be ".odttc" [S2.20].

3. The content type for the part containing the obfuscated font MUST match the definition in §A [M2.2].
4. Perform an XOR operation on the first 32 bytes of the binary data of the font part with the array consisting of the bytes referred to by 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} , B_{01} , B_{02} , and B_{03} , in that order and repeating the array once. The result is an obfuscated font.
5. Store the obfuscated font in a part with the generated name.

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

1. If the content type of the part containing the font is not the obfuscated font content type as specified in A, process the font without any de-obfuscation steps.
2. For font parts with the obfuscated font content type as specified in A, de-obfuscate the font by following these rules:
 - a. Remove the extension from the last segment of the name of the part containing the font.
 - b. Convert the remaining characters of the last segment to a GUID using the byte ordering described above.
 - c. Perform an XOR operation on the first 32 bytes of the binary data of the obfuscated font part with the array consisting of the bytes referred to by 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} , B_{01} , B_{02} , and B_{03} , in that order and repeating the array once. The result is a non-obfuscated font.
 - d. Use the non-obfuscated font for the duration of the document processing, but do not leave any local or otherwise user-accessible copy of the non-obfuscated font.

9.1.7.4 Print and Preview Restricted Fonts

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

When editing content, producers MUST NOT edit a document where the FixedDocument part has a Restricted Font relationship [M2.43]. When editing content, producers MUST instantiate an error condition when encountering any font with the print and preview restriction bit set for which no Restricted Font relationship has been added to the FixedDocument part [M2.12]. ~~Consumers that are also producers MUST NOT edit a document where the FixedDocument part has a Restricted Font relationship [M2.43]. When invoking editing functionality, consumers that are also producers MUST treat as an error any font with the print and preview restriction bit set for which no Restricted Font relationship has been added to the FixedDocument part [M2.12].~~

Consumers that are not also producers MUST consider an XPS Document valid even if the producer failed to properly set the Restricted Font relationship [M2.12].

9.1.7.5 Non-Standard Font Compatibility Encoding

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

1 *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

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

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

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

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

14 When processing <Glyphs> elements that use this compatibility encoding, character codes in
 15 the range 0x20-0xff are mapped to PUA codepoints. Therefore, character codes in the range
 16 0x80-0x9f are not considered non-printable Unicode control codes.

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

- 19 • Search is unpredictable
- 20 • Copy and paste functionality is unpredictable
- 21 • Glyph rendering is unpredictable, especially between different consumers

22 Producers SHOULD NOT use this non-standard encoding and they SHOULD write PUA
 23 codepoints to the UnicodeString attribute [S2.15].

24 **9.1.8 Remote Resource Dictionary Parts**

25 *A remote resource dictionary* allows producers to define resources that can be reused across
 26 many pages, such as a brush. This is stored in a Remote Resource Dictionary part. For more
 27 information, see §14.2.3.1.

9.1.9 PrintTicket Parts

This Standard provides a mechanism for including user intent and device configuration settings within an XPS Document as PrintTicket parts. *PrintTicket parts* enable the association of settings with parts within an XPS Document. The format to be used for PrintTickets is implementation-defined. This Standard defines how to associate those PrintTicket parts with XPS Documents. If the consumer understands the content of the PrintTicket, then the PrintTicket part SHOULD be processed when the XPS Document is printed [S2.36]. PrintTicket parts can be attached only to FixedDocumentSequence, FixedDocument, and FixedPage parts, and each of these parts MUST attach no more than one PrintTicket [M2.59]. ~~PrintTicket parts provide user intent and device configuration information to printing consumers. PrintTicket parts MUST be processed when the XPS Document is printed [M2.58]. PrintTicket parts can be attached only to FixedDocumentSequence, FixedDocument and FixedPage parts and each of these parts MUST attach no more than one PrintTicket [M2.59]. PrintTickets can provide override settings to be used when printing the part to which they are attached.~~

~~9.1.9.1 — PrintTicket Format~~

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

~~9.1.9.2~~ 9.1.9.1 Mapping PrintTicket Parts to Fixed Payload Parts

XPS Documents contain a hierarchy of FixedDocumentSequence, FixedDocument, and FixedPage parts, as defined in §10. The association of PrintTickets with FixedDocumentSequence, FixedDocument, and FixedPage parts reflects this hierarchy and enables the scope of settings specified in PrintTicket parts to be limited to the FixedDocumentSequence, FixedDocument, and FixedPage parts within the XPS Document. Domain-specific implementations are responsible for specifying how the settings provided in the PrintTicket parts are scoped.

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

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

1 ~~Within an XPS Document, there is a direct mapping between the PrintTicket levels and the fixed~~
2 ~~payload parts:~~

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

15 ~~PrintTicket parts are associated with parts via the PrintTicket relationship defined in §E.~~

16 ~~9.1.9.3 — Processing PrintTicket Parts~~

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

18 ~~When processing a PrintTicket, consumers MUST first remove all levels of PrintTicket content~~
19 ~~not applicable to the current element [M2.63] (see §9.1.9.2).~~

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

24 ~~Print settings are expressed by scoped Print Schema elements. Elements can interact between~~
25 ~~different levels in the PrintTicket hierarchy. Elements that interact between levels MUST be~~
26 ~~specified at the root of each level ticket [M2.59]. A keyword merge conflict between PrintTicket~~
27 ~~settings is defined as the same root-level Print Schema element denoted by the same name~~
28 ~~attribute value appearing in multiple level tickets. There are two options for interactions:~~

29 ~~1. — If there is no merge conflict, a prefix-scoped element MUST be pushed down, or~~
30 ~~inherited, from a more general ticket to a more specific ticket [M2.66]. This case is isomorphic~~
31 ~~to the case where both tickets contain an identical element.~~

32 ~~2. — If there is a merge conflict, the setting from the most-specific ticket MUST take~~
33 ~~precedence [M2.67]. That is, a page-scoped setting in a page-level PrintTicket overwrites an~~
34 ~~identical page-scoped setting in a document-level or job-level PrintTicket. Likewise, a~~
35 ~~document-scoped setting in a document-level PrintTicket overwrites an identical document-~~
36 ~~scoped setting in a job-level PrintTicket.0.~~

37 ~~To determine the print settings in the XPS Document the following algorithm should be applied:~~

- 38 ~~1. — Validate the job-level PrintTicket associated with the fixed document sequence by~~
39 ~~merging and validating the PrintTicket with the default PrintTicket for the print consumer.~~
40 ~~The default PrintTicket represents the default configuration of the print consumer's state;~~
41 ~~in the case of a print consumer, the state is the current device and print driver~~
42 ~~configuration. Call the resulting ticket the "validated job-level PrintTicket." If no job-level~~
43 ~~PrintTicket is supplied, use the default PrintTicket.~~

- ~~2. For each FixedDocumentX referenced by the fixed document sequence, perform the following steps:~~
- ~~a. Merge and validate the document level PrintTicket associated with FixedDocumentX with the validated job level PrintTicket from Step 1.~~
 - ~~b. Call the resulting ticket the "validated DocumentX level PrintTicket." If no document level PrintTicket is supplied for the current fixed document, the validated job level PrintTicket from Step 1 should be used.~~
- ~~3. For each FixedPageY referenced by each FixedDocumentX in the fixed document sequence, perform the following steps: 0.~~
- ~~a. Merge and validate the page level PrintTicket associated with FixedPageY with the validated DocumentX level PrintTicket from Step 2.~~
 - ~~b. Call the resulting ticket the "validated PageXY level PrintTicket." If no page level PrintTicket is supplied for the current fixed page, the validated DocumentX level PrintTicket from Step 2 should be used.~~

9.1.10 SignatureDefinitions Part

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

Digital signature definitions are stored in a SignatureDefinitions part. A FixedDocument part refers to a SignatureDefinitions part using a relationship of the SignatureDefinitions type. For more information, see §A.

The SignatureDefinitions part is OPTIONAL [O2.6]. Signature definitions MUST conform to the Signature Definitions schema as defined in §A.1 [M2.86M2.72].

For more information on digital signature support in XPS Documents, see §17.

9.1.11 DocumentStructure Part

Explicitly authored document structure information is stored in the DocumentStructure part. This part contains the document outline and defines the framework for every element in fixed pages in terms of semantic blocks, each of which is called a *story*. A story is split into StoryFragments parts, which contain content structure markup that defines semantic blocks such as paragraphs and tables. For more information, see §16.

Document structure markup contains a root <DocumentStructure> element. See §16 for markup details. The <DocumentStructure> element uses the Document Structure namespace specified in §D.1.

The DocumentStructure part is referenced by relationship from the FixedDocument part, with the relationship type as specified in §A. The content type of the DocumentStructure part is also specified in §A.

Consumers MAY provide an algorithmic construction of the structure of an XPS Document based on a page-layout analysis [O2.27], but they MUST NOT use such a method to derive structure for any part of the XPS Document included in the DocumentStructure part [M2.68]. A consumer capable of calculating reading order from the layout of the document MUST use the reading

1 order specified in the DocumentStructure part, even though the derived order might be
2 perceived as preferable to the specified order [M2.68].

3 **9.1.12 StoryFragments Part**

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

6 StoryFragments part markup contains a root <StoryFragments> element. See §16 for markup
7 details. The <StoryFragments> element uses the Document Structure namespace specified
8 in §D.1.

9 The StoryFragments part is referenced by relationship from its associated FixedPage part. The
10 content type of the StoryFragments part is specified in §D.2.

11 **9.2 Part Naming Recommendations**

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

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

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

- 23 • The FixedDocumentSequence part name SHOULD contain only one segment, and that
24 segment SHOULD have the extension ".fdseq". [*Example: "/FixedDocSeq.fdseq" end*
25 *example*] [S2.24].
- 26 • A FixedDocument part name SHOULD contain three segments, using "/Documents/*n*/" in
27 the first two segments and the extension ".fdoc" [S2.25]. Here, *n* SHOULD be a numeral
28 that represents the ordinal position of the fixed document in the fixed document
29 sequence [S2.25]. [*Example: The fixed document referenced by the Source attribute of*
30 *the third <DocumentReference> child of the <FixedDocumentSequence> element could*
31 *be "/Documents/3/FixedDocument.fdoc". end example*]
- 32 • A FixedPage part name SHOULD contain four segments, using "/Documents/*n*/Pages/" as
33 the first three segments and the extension ".fpage" on the last segment [S2.26]. Here, *n*
34 represents the fixed document that includes this page. [*Example: The third page of the*
35 *second document might be "/Documents/2/Pages/3.fpage". end example*]
- 36 • Resource parts MAY be named to indicate whether their intended use is at the document
37 level or as a shared resource for all documents [O2.28]. A resource that is specific to a
38 particular document SHOULD have a part name that begins with the three segments
39 "/Documents/*n*/Resources/" where *n* is the particular fixed document [S2.27]. A
40 resource intended to be shared across documents SHOULD begin with the segment
41 "/Resources/" and SHOULD have a final segment that is a globally unique identifier
42 followed by the appropriate extension for that resource [S2.27]. [*Example:*
43 *"/Resources/Fonts/63B51F81-C868-11D0-999C-00C04FD655E1.odttf" end example*]

- 1 A Font part name SHOULD append the segment "Fonts/" to the resource part name prefix
 2 specified above [S2.27]. [*Example*: A font might be named
 3 "/Documents/1/Resources/Fonts/Arial.ttf" or "/Resources/Fonts/F2ABC7B7-C60D-4FB9-
 4 AAE4-3CA0F6C7038A.odttf". *end example*]
- 5 An Image part name SHOULD append the segment "Images/" to the resource part name
 6 specified above [S2.27]. [*Example*: An image might be named
 7 "/Documents/3/Resources/Images/dog.jpg" or "/Resources/Images/E0D79307-846E-
 8 11CE-9641-444553540000.jpg". *end example*]
- 9 A Remote Resource Dictionary part name SHOULD append the segment "Dictionaries/" to
 10 the resource part name specified above [S2.27]. Remote resource dictionaries SHOULD
 11 also use the ".dict" extension [S2.27]. [*Example*: A resource dictionary might be named
 12 "/Documents/2/Resources/Dictionaries/Shapes.dict" or
 13 "/Resources/Dictionaries/0DDF3BE2-E692-15D1-AB06-B0AA00BDD685.dict". *end*
 14 *example*]
- 15 • Any DocumentStructure part name SHOULD contain four segments using
 16 "/Documents/*n*/Structure/" as the first three segments and the extension ".struct"
 17 [S2.28]. Here *n* represents the fixed document that this structure is associated with.
 18 [*Example*: The DocumentStructure part for the first document in a fixed document
 19 sequence could be "/Documents/1/Structure/DocStructure.struct". *end example*]
 - 20 • Any StoryFragments part name SHOULD contain five segments using
 21 "/Documents/*n*/Structure/Fragments" as the first four segments and the extension
 22 ".frag" [S2.29]. Here *n* represents the fixed document that these parts are associated
 23 with. [*Example*: A StoryFragment part associated with the third page of the second
 24 document in a fixed document sequence could be
 25 "/Documents/2/Structure/Fragments/3.frag". *end example*]
 - 26 • ICC profile part names SHOULD contain four segments, using "/Documents/*n*/Metadata/"
 27 as the first three segments, where *n* is the fixed document that uses these parts
 28 [S2.30]. If an ICC profile part is shared across documents, the part name SHOULD
 29 contain two segments, using "/Metadata/" as the first segment and a second segment
 30 that is a string representation of a globally unique identifier, followed by an extension
 31 [S2.30]. ICC profiles SHOULD use an appropriate extension for the color profile type.
 32 [S2.30] [*Example*: ".icm" *end example*]
 - 33 • Thumbnail part names SHOULD contain four segments, using "/Documents/*n*/Metadata/"
 34 as the first three segments, where *n* is the fixed document that uses the thumbnail
 35 [S2.31]. If the Thumbnail part relates to the package as a whole, the part name
 36 SHOULD contain two segments, using "/Metadata/" as the first segment and a second
 37 segment that is a string representation of a globally unique identifier, followed by an
 38 extension [S2.31]. Thumbnails SHOULD use an extension appropriate to the image type,
 39 either ".png" or ".jpg" [S2.31]. [*Example*: A Thumbnail part for a particular fixed page
 40 might be "/Documents/1/Metadata/5.png". *end example*]
 - 41 • PrintTicket part names associated with the entire job SHOULD be associated via
 42 relationship with the FixedDocumentSequence part and contain two segments, using
 43 "/Metadata/" as the first segment [S2.32]. PrintTicket parts associated with a particular
 44 fixed document or fixed page SHOULD contain four segments, using
 45 "/Documents/*n*/Metadata/" as the first three segments, where *n* is the fixed document
 46 that uses these parts [S2.32]. PrintTicket parts [based on XML](#) SHOULD use the
 47 extension ".xml" [S2.32]. [*Example*: A PrintTicket associated with the entire job could be
 48 "/Metadata/Job_PT.xml" and a PrintTicket associated with a single page might be
 49 "/Documents/1/Metadata/Page2_PT.xml". *end example*]

- 1 • The names of any non-standard parts that are associated with a particular fixed
2 document SHOULD contain four segments, using “/Documents/*n*/Other/” as the first
3 three segments. Here, *n* is the fixed document to which the part belongs [S2.33].

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

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

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

27 *end example]*

28 **9.3 XPS Document Markup**

29 XPS Document Markup is used to describe the content of fixed pages within an XPS Document.
30 This XML-based markup has been designed to address the requirements for describing
31 graphical content within electronic paper documents. The graphical primitives described by the
32 elements, attributes and attribute values in the markup are completely sufficient for
33 representing document content as acquired from, or output to, physical paper by a variety of
34 document devices and applications. The XPS Document Markup has also been developed
35 consistent with the independent development of compatible systems that produce or consume
36 XPS Documents.

37 The design of XPS Document Markup reflects the tradeoffs between the following two,
38 sometimes competing, goals:

- 39 1. XPS Document markup should be parsimonious; that is, it should include only the minimum
40 set of primitive operations and markup constructs necessary to render text and graphics
41 with full fidelity. Redundancy in the Standard increases the opportunity for independent
42 implementations, such as printer-resident raster image processors (RIPs), viewers, and
43 interactive applications, to introduce accidental incompatibilities. Redundancy also increases
44 the cost of implementation and testing, and, typically, the required memory footprint.
- 45 2. XPS Document markup should be compact; that is, the most common graphical primitives
46 for vector graphics and text-rendering should have compact representations. Inefficient

1 representations compromise the performance of systems handling XPS Documents. As byte-
2 count increases, so does communication time. Although compression can be used to
3 improve communication time, it cannot eliminate the performance loss caused by inefficient
4 representations.

5 **9.3.1 Support for Versioning and Extensibility**

6 XPS Document markup has been designed in anticipation of the evolution of this Standard. It
7 also allows third parties to extend the markup. XPS Document markup incorporates the Markup
8 Compatibility and Extensibility Standard incorporated by the Office Open XML Standard.

9 The following parts MAY include elements and attributes defined in the Markup Compatibility
10 and Extensibility Standard [O2.29]:

- 11 • DocumentStructure
- 12 • FixedDocument
- 13 • FixedDocumentSequence
- 14 • FixedPage
- 15 • Relationships
- 16 • Remote Resource Dictionary
- 17 • SignatureDefinitions
- 18 • StoryFragments

19 Consumers of these parts MUST support the Markup Compatibility and Extensibility Standard
20 [M2.69]. Before attempting to validate one of these parts against a schema, processors MUST
21 remove all markup compatibility elements and attributes and all ignorable elements and
22 attributes not defined in the expected version of XPS Document markup [M2.69].

23 Markup compatibility elements and attributes that appear in one XPS Document part do not
24 carry through to a second part via an inline URI reference in the XML markup. Likewise the
25 markup compatibility mechanisms do not carry through from part to part via relationship.

26 **9.3.2 XML Usage**

27 All XML content of the parts defined in this Standard MUST conform to the following validation
28 rules:

- 29 1. XML content MUST be encoded using either UTF-8 or UTF-16. If any such part includes an
30 encoding declaration (as defined in §4.3.3 of the XML Standard), that declaration MUST
31 NOT name any encoding other than UTF-8 or UTF-16 [M2.70].
- 32 2. The XML 1.0 Standard allows for the usage of Data Type Definitions (DTDs), which
33 enable Denial of Service attacks, typically through the use of an internal entity expansion
34 technique. As mitigation for this potential threat, DTD content MUST NOT be used in the
35 XML markup defined in this Standard, and consumers MUST [instantiate an error condition](#)
36 [when encountering](#) ~~treat the presence of~~ DTD content ~~as an error~~ [M2.71].
- 37 3. If the XML content contains the Markup Compatibility and Extensibility namespace, as
38 described in the Markup Compatibility and Extensibility Standard, it MUST be processed
39 to remove Markup Compatibility and Extensibility elements and attributes, ignorable
40 namespace declarations, and ignored elements and attributes before applying further
41 validation rules below [M2.69].

- 1 4. XML content MUST be valid against the corresponding W3C XSD schema defined in this
2 Standard. In particular, the XML content MUST NOT contain elements or attributes drawn
3 from namespaces that are not explicitly defined in the corresponding XSD unless the XSD
4 allows elements or attributes drawn from any namespace to be present in particular
5 locations in the XML markup [M2.72].
- 6 5. XML content MUST NOT contain elements or attributes drawn from "xml" or "xsi"
7 namespaces unless they are explicitly defined in the W3C XSD schema or by other means
8 in the Standard [M2.73].

9 **9.3.3 Markup Model**

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

14 Fixed page markup is expressed using elements and attributes and is based on a higher-level
15 abstract model of contents and properties. Some fixed page elements can hold "contents,"
16 which are expressed as child elements. Properties can be expressed either as attributes or child
17 elements.

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

20 **9.3.3.1 Namespaces**

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

- 22 • The XPS Document namespace, the principal namespace used for elements and
23 attributes in fixed page markup. For more information, see §A.
- 24 • The Resource Dictionary Key namespace, which allows certain XPS Document elements
25 to be included in a resource dictionary, as described in §14.2.
- 26 • The Markup Compatibility namespace, which supports the Markup Compatibility and
27 Extensibility Standard as defined in the OPC Standard.

28 **9.3.3.2 Properties**

29 A *property* is a characteristic of an element. XPS Document property values can be expressed
30 either as property attributes or property elements. *Property values* can be stored in a resource
31 dictionary and referenced by an attribute that uses a special syntax to express its value. For
32 more information, see §14.2.

33 Properties MUST NOT be set more than once, regardless of the syntax used to specify the value
34 [M2.74]. In certain cases, they can be specified using either property attributes or property
35 elements. Consumers MUST [instantiate an error condition when encountering ~~tr~~-properties](#)
36 that are specified in both ways ~~as an error~~ [M2.74].

37 Some properties are common to several fixed page elements. For more information, see §14.

38 **9.3.3.2.1 Composable Property Values**

39 Some fixed page properties are composable, meaning that the page marking effect is
40 determined by combining the property value of a given element with that of its parent and
41 ancestor elements. [Example: A <Path> element with an Opacity value of 0.5 nested inside a

1 <Canvas> element with an Opacity value of 0.5 results in an effective 25% opacity of the
2 <Path> element when rendered. *end example*]

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

8 For more information, see §18.1.3, and §18.5.

9 **9.3.3.2.2 Property Attribute Syntax**

10 Some property values can be expressed using simple XML attribute syntax, that is, with a text
11 string. The value of properties used to describe geometries can be expressed using an
12 abbreviated syntax. For more information, see §11.2.3.

13 *Example 9-3. Property attribute syntax*

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

```
15 <!-- Property Attribute Syntax -->  
16 <SolidColorBrush Color="#FF0000" />
```

17 *end example*]

18 **9.3.3.2.3 Property Element Syntax**

19 Some property values can also be expressed using a child element to describe the property
20 value. These property elements are included to enable usage of the markup compatibility
21 mechanisms described in the Markup Compatibility and Extensibility Standard. The element
22 name is derived from a combination of a parent element name and the property name,
23 separated by a dot (.) character.

24 The order of child property elements is significant: they MUST occur before any contents of the
25 parent element and they MUST appear in the sequence specified in the schema [[M2.87](#)~~M2.72~~].

1 *Example 9–4. Property element syntax*

2 When specifying Clip and RenderTransform properties of the canvas, both must appear before
3 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 whitespace
23 character is allowed, multiple whitespace characters MAY be used [O2.30]. Attributes that
24 specify comma-delimited attribute values MAY, unless specified otherwise, OPTIONALLY include
25 whitespace characters preceding or following the comma [O2.31]. XPS Document markup MUST
26 NOT use the xml:space attribute [M2.75]. Additionally, where the XPS Document schema
27 specifies attributes of types that allow whitespace collapsing, leading and trailing whitespace in
28 the attribute value MAY be used along with other whitespace that relies on the whitespace
29 collapsing behavior specified in the XML Schema Standard [O2.32].

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

31 **9.3.5 Language**

32 Language information supports the following features:

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

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

1 another application. This scenario applies to most Indic-language fonts, some East Asian-
2 language fonts, and others. *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 xml:lang
5 attribute, the value of which is inherited by child and descendant elements [M2.76].

6 -This attribute is defined in the W3C XML Standard.

7 xml:lang is REQUIRED for <FixedPage> elements [M2.88]. ~~and xml:lang~~ MAY be used with
8 <Canvas>, <Path>, and <Glyphs> elements [O2.33]. ~~it is not valid~~ xml:lang MUST NOT be used
9 on any other fixed page markup element [M2.89M2.72]. xml:lang is also REQUIRED for the
10 <DocumentOutline> element for document structure [M2.90]. ~~xml:lang~~ ~~and is~~ OPTIONAL for the
11 <OutlineEntry> element [O2.34M2.72]. When the language of the contents is unknown and is
12 required, the value "und" (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**

diagram	
annotation	Specifies a sequence of fixed documents.

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**

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>required</td> <td></td> <td></td> <td>Specifies the URI of the fixed document content. The specified URI MUST refer to a FixedDocument part within the XPS Document [M3.2].</td> </tr> </tbody> </table>	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].
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.
------------	---

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

10.2 <FixedDocument> Element

element **FixedDocument**

diagram	
annotation	Binds an ordered sequence of fixed pages together into a single multi-page document.

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

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

```

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

13 *end example]*

10.2.1 <PageContent> Element

element **PageContent**

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>required</td> <td></td> <td></td> <td>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].</td> </tr> </tbody> </table>	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].
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-The width of the pages contained in the page content.
	Height	ST_GEOne				Typical-The height of the 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. The number
2 of pages in the document can be determined by counting the number of <PageContent>
3 elements.

4 The <PageContent> element has a single required attribute, Source, which refers to a
5 FixedPage part. It can optionally include advisory Height and Width attributes to indicate the size
6 of a single page. (The authoritative height and width are specified by the fixed page.) The
7 Height and Width attribute values allow consumers such as viewers to make initial visual layout
8 estimates quickly, without loading and parsing all of the individual fixed pages. These
9 consumers then update the page dimensions when the fixed page is loaded, if they differ.

10 The <PageContent> element has one allowable child element, <PageContent.LinkTargets>, and
11 it MUST NOT contain more than a single child element [M3.21M2.72].

12 Producers MUST NOT produce markup where a <PageContent> element references the same
13 fixed page referenced by any other <PageContent> element in the entire XPS Document, even
14 in other fixed documents within the fixed payload [M3.6].

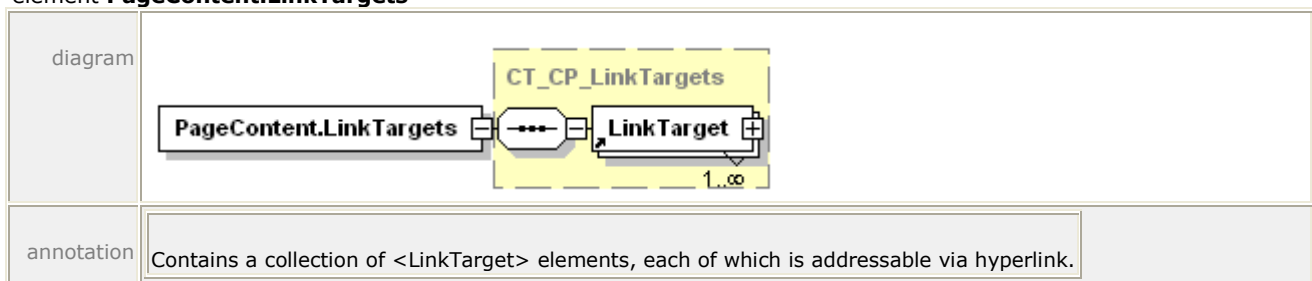
15 *Example 10-3. <PageContent> usage*

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

20 *end example]*

21 10.2.2 <PageContent.LinkTargets> Element

22 element **PageContent.LinkTargets**



23 The <PageContent.LinkTargets> element defines the list of link targets that specify each named
24 element on the page that can be addressed by hyperlink.

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

2 In the following markup, `Pages/2.fpage` contains two `<LinkTarget>` elements with Name
3 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

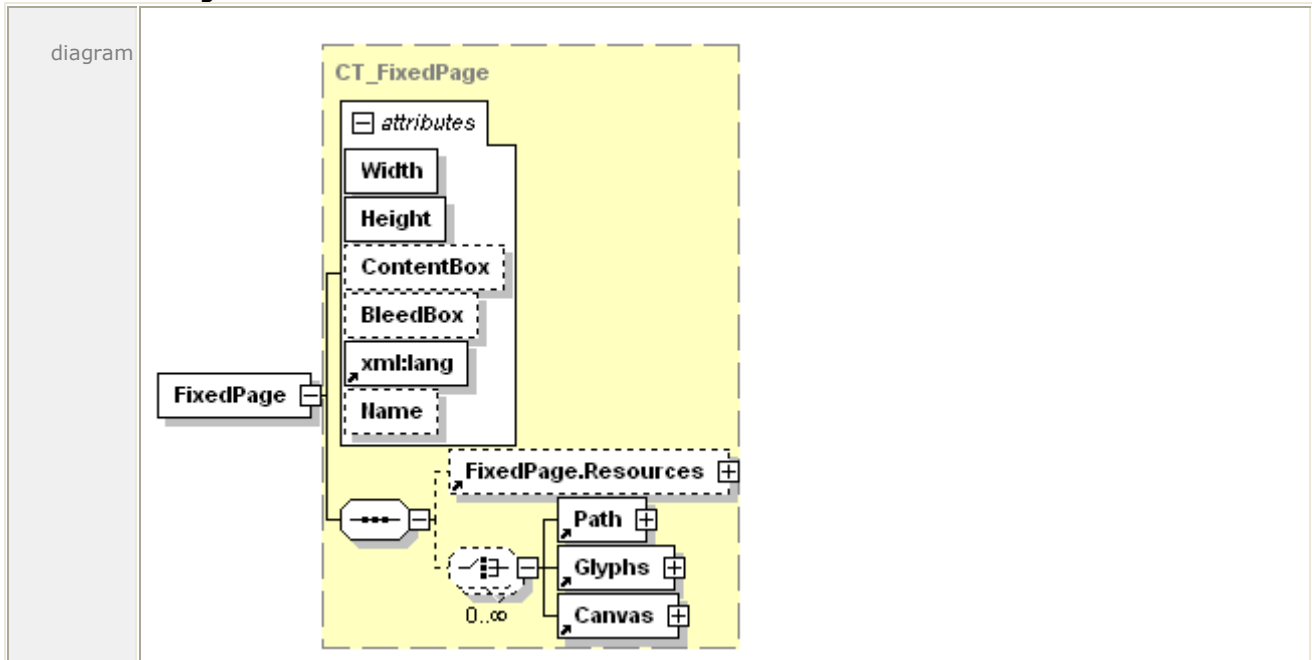
15 element **LinkTarget**

diagram						
attributes	Name	Type	Use	Default	Fixed	Annotation
	Name	<u>ST_Name</u>	required			Contains a string value that identifies the current element as a named, addressable point in the document for the purpose of hyperlinking.
annotation	Specifies an addressable point on the page.					

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

1 **10.3 <FixedPage> Element**

2 element **FixedPage**



attributes	Name	Type	Use	Default	Fixed	Annotation
	Width	ST_GEOne	required			Width of the page, expressed as a real number in units of the effective coordinate space.
	Height	ST_GEOne	required			Height of the page, expressed as a real number in units of the effective coordinate space.
	ContentBox	ST_ContentBox				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	ST_BleedBox				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 a
 2 FixedPage part. The fixed page contains the elements that together form the basis for all
 3 markings rendered on the page: <Paths>, <Glyphs>, and the optional <Canvas> grouping
 4 element.

5 The fixed page MUST specify a height, width, and default language [[M3.22](#)~~M2.72~~].

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

8 *Example 10-5. Fixed page markup*

```

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

```

18 *end example]*

19 **10.3.1 BleedBox Attribute**

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

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

- 26 • The BleedBox BleedOriginX value MUST be less than or equal to 0 [M3.7].
- 27 • The BleedBox BleedOriginY value MUST be less than or equal to 0 [M3.8].
- 28 • The BleedBox BleedWidth value MUST be greater than or equal to the fixed page Width
 29 attribute value plus the absolute value of the Bleedbox BleedOriginX value [M3.9].
- 30 • The BleedBox BleedHeight value MUST be greater than or equal to the fixed page Height
 31 attribute value plus the absolute value of the BleedBox BleedOriginY value [M3.10].

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

10.3.2 ContentBox Attribute

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

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

- The ContentBox ContentOriginX value MUST be greater than or equal to 0 and less than the fixed page Width attribute value [M3.11].
- The ContentBox ContentOriginY value MUST be greater than or equal to 0 and less than the fixed page Height attribute value [M3.12].
- The ContentBox ContentWidth value MUST be less than or equal to the difference between the fixed page Width attribute value and the ContentBox ContentOriginX value [M3.13].
- The ContentBox ContentHeight value MUST be less than or equal to the difference between the fixed page Height attribute value and the ContentBox ContentOriginY value [M3.14].

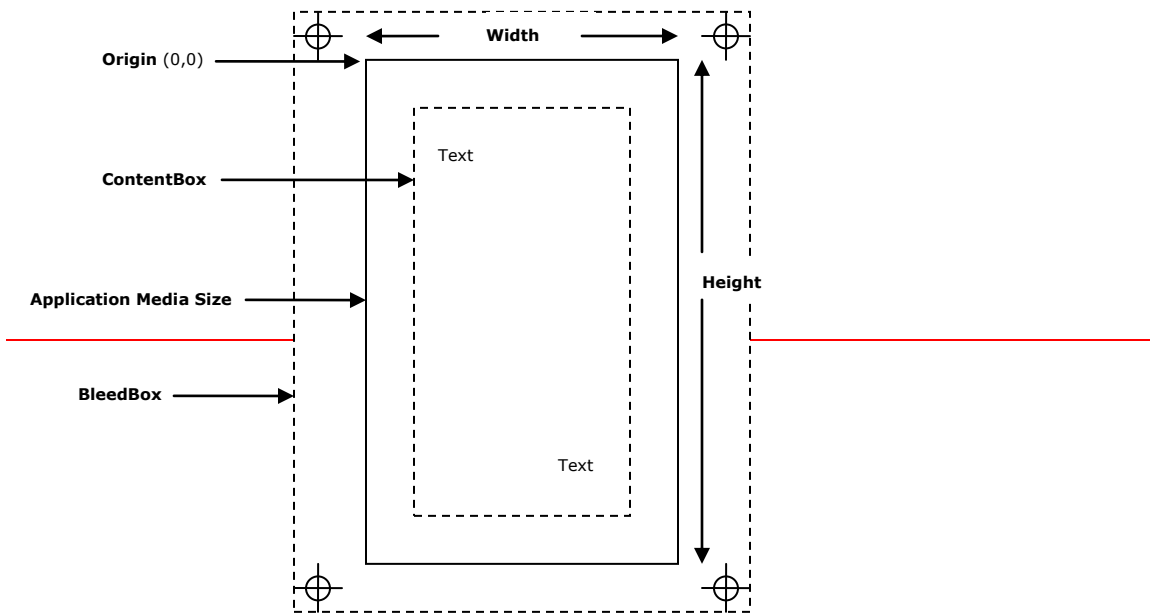
If the ContentBox attribute is omitted, the default value is "0,0,Width,Height".

10.3.3 Page Size Terminology

The following terminology is used to describe page sizes:

- The *producer media size* represents the media size that is used by the producer for layout purposes. This size is described by the Height and Width attributes.
- The *producer bleed size* represents the overflow or "bleed" box used by the producer for registration and layout. This size is described by the BleedBox attribute. If the BleedBox attribute is not present, then the producer bleed size is defined as the producer media size.
- The *producer content size* represents the content bounding box specified by the producer. This size is described by the ContentBox attribute. If the ContentBox attribute is not present, then the producer content size is defined as the producer media size.
- The *physical media size* represents the physical media on which the content will be printed. This size is described by the PageMediaSize keyword in the Print Schema. The PageMediaSize is page-orientation agnostic. This means that the representation of physical media size is defined by two dimensions, PageMediaSizeWidth and PageMediaSizeHeight. For more information, see the Print Schema specification.
- The *physical imageable size* represents the area that is printable by a specific device. This size is described by the PageImageableSize keyword in the Print Schema. The PageImageableSize is relative to the physical media size (PageMediaSize keyword) and the orientation (PageOrientation keyword). For more information, see the Print Schema specification.

Figure 10-1. Page regions



10.3.4 Media Orientation and Scaling

When rendering a FixedPage for printing, consumers are responsible for mapping from FixedPage content to the physical media. Differences in device capabilities and device configuration result in a large number of permutations for the mapping. The positioning, scaling, orientation, and clipping of FixedPage content when mapping to physical media MAY be controlled by settings provided in the PrintTicket [O3.1]. In the absence of settings provided in the PrintTicket, the mapping of FixedPage content to the physical media is implementation-defined.

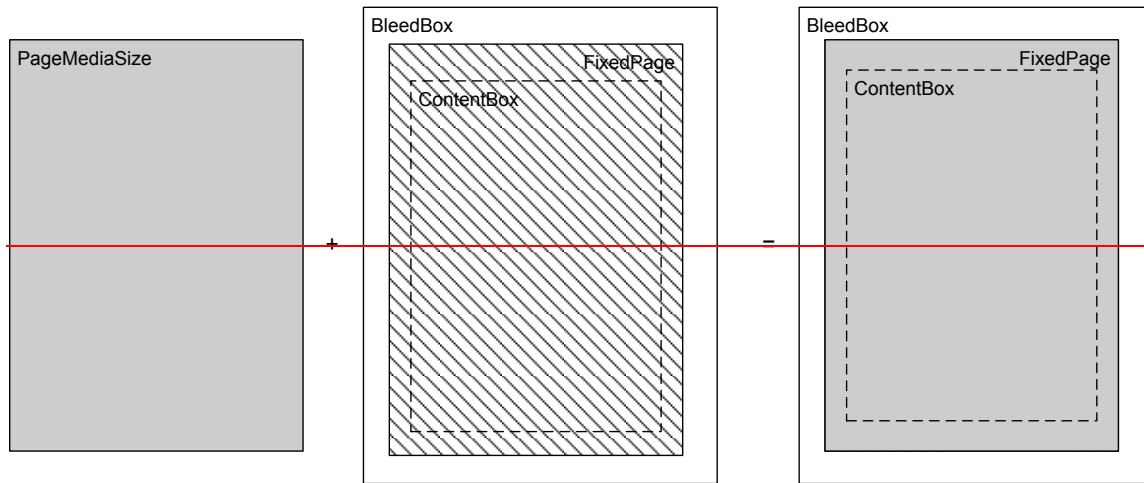
By default, consumers SHOULD clip to the FixedPage Width and Height [S3.5]; consumers MAY provide implementation-defined mechanisms to select alternative clipping strategies [O3.2]. [Note: For example, an implementation can provide a PrintTicket setting to allow control of consumer clipping of FixedPage content to one of the defined bounding boxes. end note]

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

~~For media orientation, the PageMediaSize, PageOrientation and the width and height of the <FixedPage> element determine the rendering of the fixed page. 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 [S3.4].~~

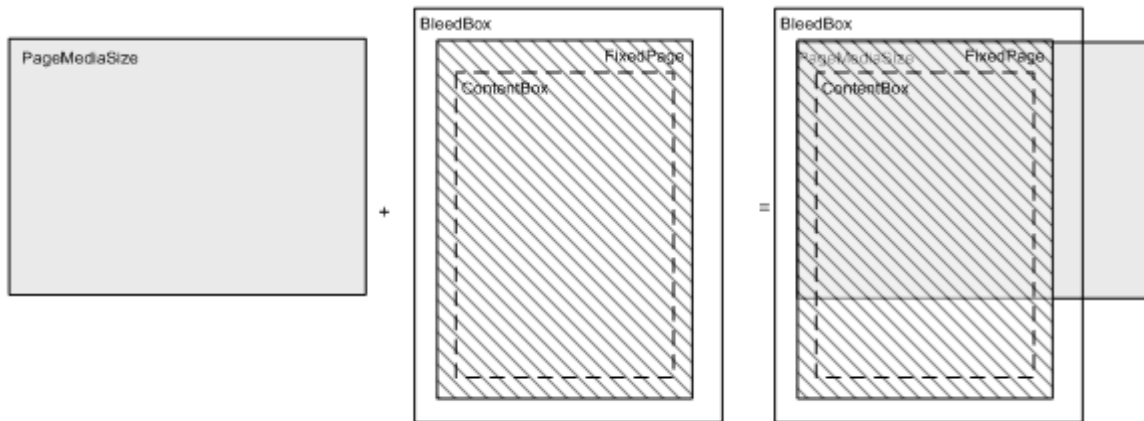
~~[Example: Consider a fixed page with a width of 816 (8.5") and a height of 1056 (11"). If the PrintTicket specifies the PageMediaSize value as Letter and the PageOrientation value as Portrait then no clipping occurs, as shown below.~~

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

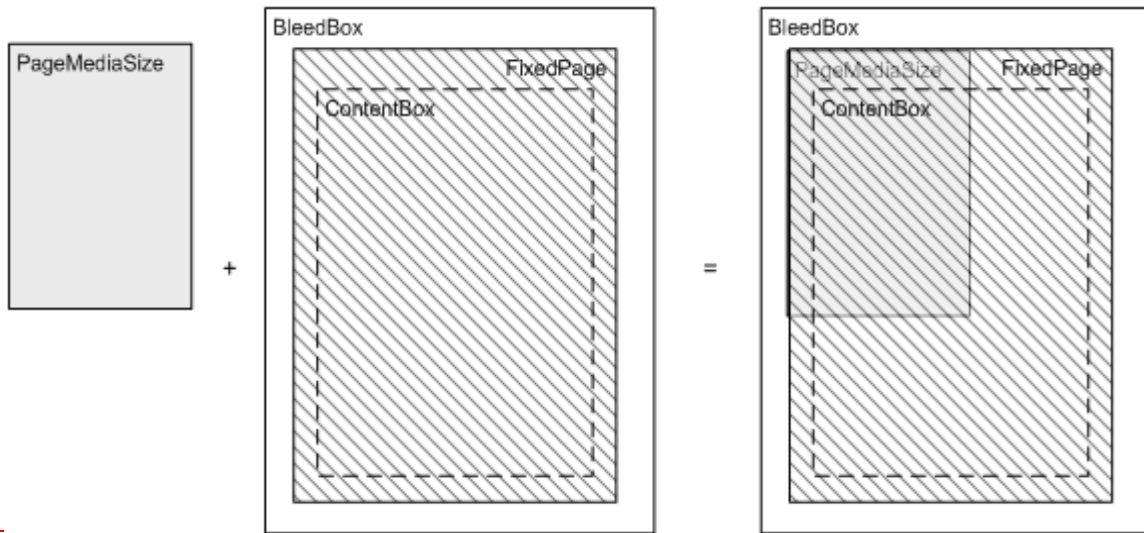


2
 3 *If the PrintTicket specifies the PageMediaSize value as Letter and the PageOrientation value as*
 4 *Landscape the bottom of the fixed page content is clipped, as shown below.*

5 *Figure 10-3. Matching PrintTicket and fixed page size with differing orientation*



6
 7 *Finally, if the PrintTicket specifies the PageMediaSize value as 4x6 and the PageOrientation*
 8 *value as Portrait, the bottom and right portions of the fixed page content are clipped, as shown*
 9 *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 differently*
5 *than those intended for portrait orientation printing; their Width attribute holds the larger value,*
6 *their Height attribute holds the smaller value.*

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

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

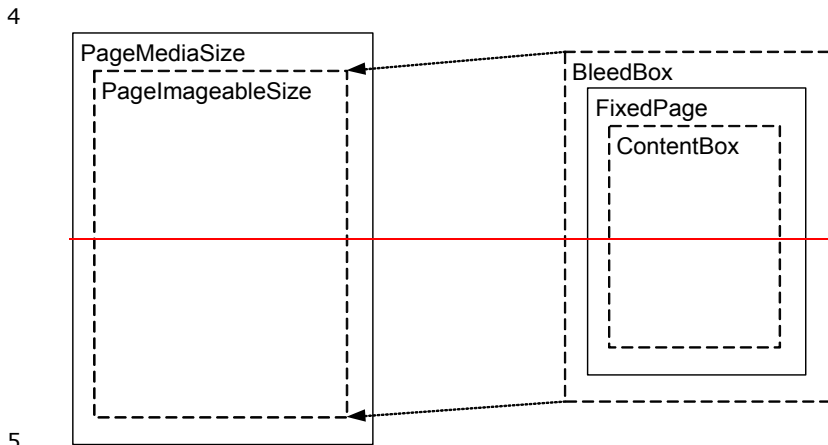
14 *For media scaling, the following page scaling options determine the rendering of the fixed page:*

- 15 • *FitApplicationBleedSizeToPageImageableSize*
- 16 • *FitApplicationContentSizeToPageImageableSize*
- 17 • *FitApplicationMediaSizeToPageImageableSize*
- 18 • *FitApplicationMediaSizeToPageMediaSize*

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

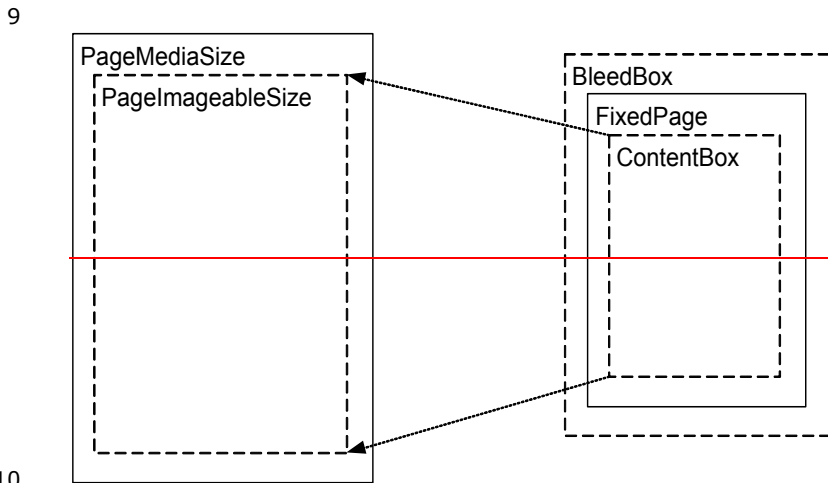
1 **10.3.4.1 – FitApplicationBleedSizeToPageImageableSize**

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



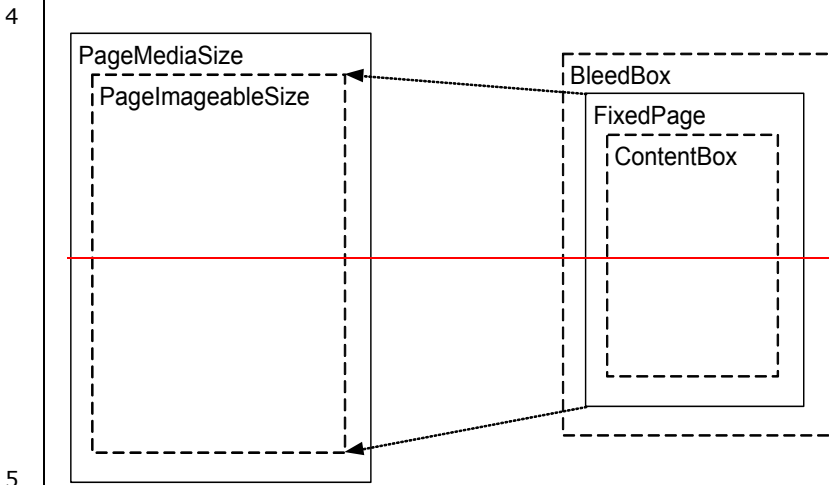
6 **10.3.4.2 – FitApplicationContentSizeToPageImageableSize**

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



1 **10.3.4.3 — FitApplicationMediaSizeToPageImageableSize**

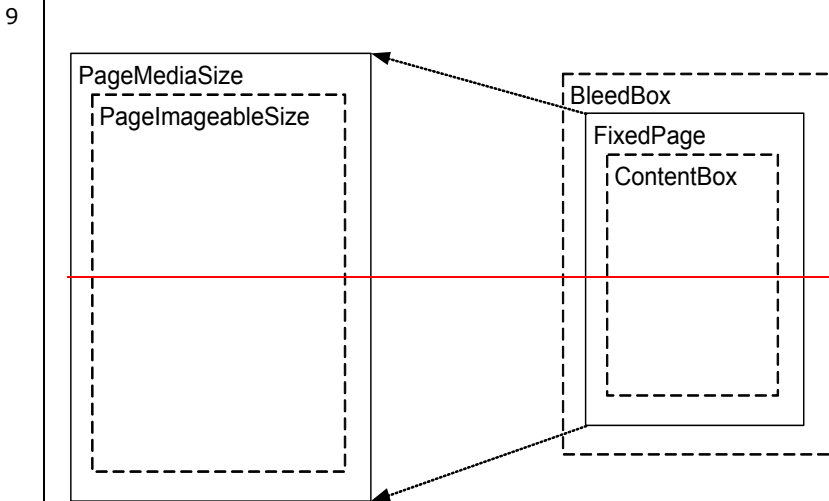
2 Consumers ~~MUST~~ scale the height and width (producer media size) to the PageImageableSize,
3 preserving the aspect ratio [M3.18]. See the Print Schema PageScaling definition.



5

6 **10.3.4.4 — FitApplicationMediaSizeToPageMediaSize**

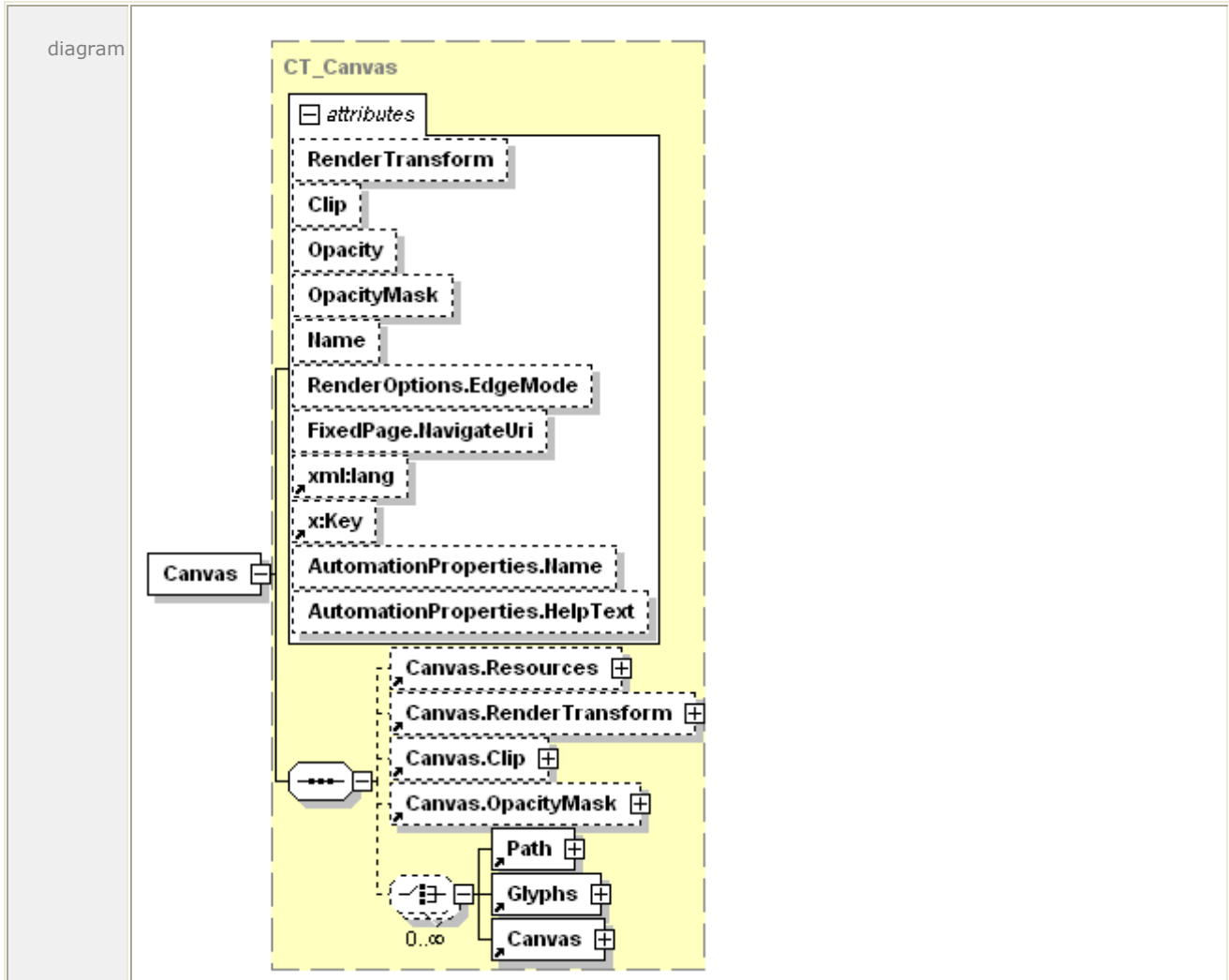
7 Consumers ~~MUST~~ scale the height and width (producer media size) to the PageMediaSize,
8 preserving the aspect ratio [M3.19]. See the Print Schema PageScaling definition.



10

1 **10.4 <Canvas> Element**

2 element **Canvas**



attributes	Name	Type	Use	Default	Fixed	Annotation
	RenderTransform	ST_RscRefMatrix				Establishes a new coordinate frame for the child and descendant elements of the canvas, such as another canvas. Also affects clip and opacity mask.
	Clip	ST_RscRefAbbrGeomF				Limits the rendered region of the element.
	Opacity	ST_ZeroOne		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	<u>ST_RscRef</u>				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	<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.
RenderOptions.EdgeMode	<u>ST_EdgeMode</u>				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 The <Canvas> element groups elements together. [*Example: <Glyphs> and <Path> elements*
2 can be grouped in a canvas in order to be identified as a unit (as a hyperlink destination) or to
3 apply a composed property value to each child and ancestor element. *end example*]

4 Some properties of the <Canvas> element are composable and affect the rendering of child
5 elements. This includes the coordinate space of the canvas. For details, see §14.

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

10 *Example 10–6. Canvas composition*

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

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

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

```

21 <Path>
22   <Path.Fill>
23     <SolidColorBrush Color="#808080" />
24   </Path.Fill>
25   <Path.Data>
26     <PathGeometry>
27       <PathFigure StartPoint="0,0" IsClosed="true">
28         <PolyLineSegment Points="200,0 200,100 0,100 0,0" />

```

```

1         </PathFigure>
2     </PathGeometry>
3 </Path.Data>
4 </Path>
5
6 <Canvas Opacity="0.5" RenderTransform="0.75,0,0,0.75,25,46">
7     <Path>
8         <Path.Fill>
9             <SolidColorBrush Color="#0000FF" />
10        </Path.Fill>
11        <Path.Data>
12            <PathGeometry>
13                <PathFigure StartPoint="0,0" IsClosed="true">
14                    <PolyLineSegment Points="200,0 200,100 0,100 0,0" />
15                </PathFigure>
16            </PathGeometry>
17        </Path.Data>
18    </Path>
19    <Glyphs
20        FontUri=" ../Resources/Fonts/times.ttf"
21        OriginX="1"
22        OriginY="100"
23        UnicodeString="EXAMPLE"
24        FontRenderingEmSize="42"
25        RenderTransform="1.0,0,0,2.0,0,-100">
26        <Glyphs.Fill>
27            <SolidColorBrush Color="#FFFFFF" />
28        </Glyphs.Fill>
29    </Glyphs>
30 </Canvas>

```

31 This markup is rendered as follows:



32

33 *end example]*

34 10.5 <Path> Element

35 The <Path> element specifies a geometry that can be filled with a brush. For more information,
 36 see §11.1.

1 **10.6 <Glyphs> Element**

2 The <Glyphs> element is used to represent a run of uniformly-formatted text from a single
3 font. The <Glyphs> element provides information for accurate rendering and supports search
4 and selection features in XPS Document consumers. For more information, see §12.1.

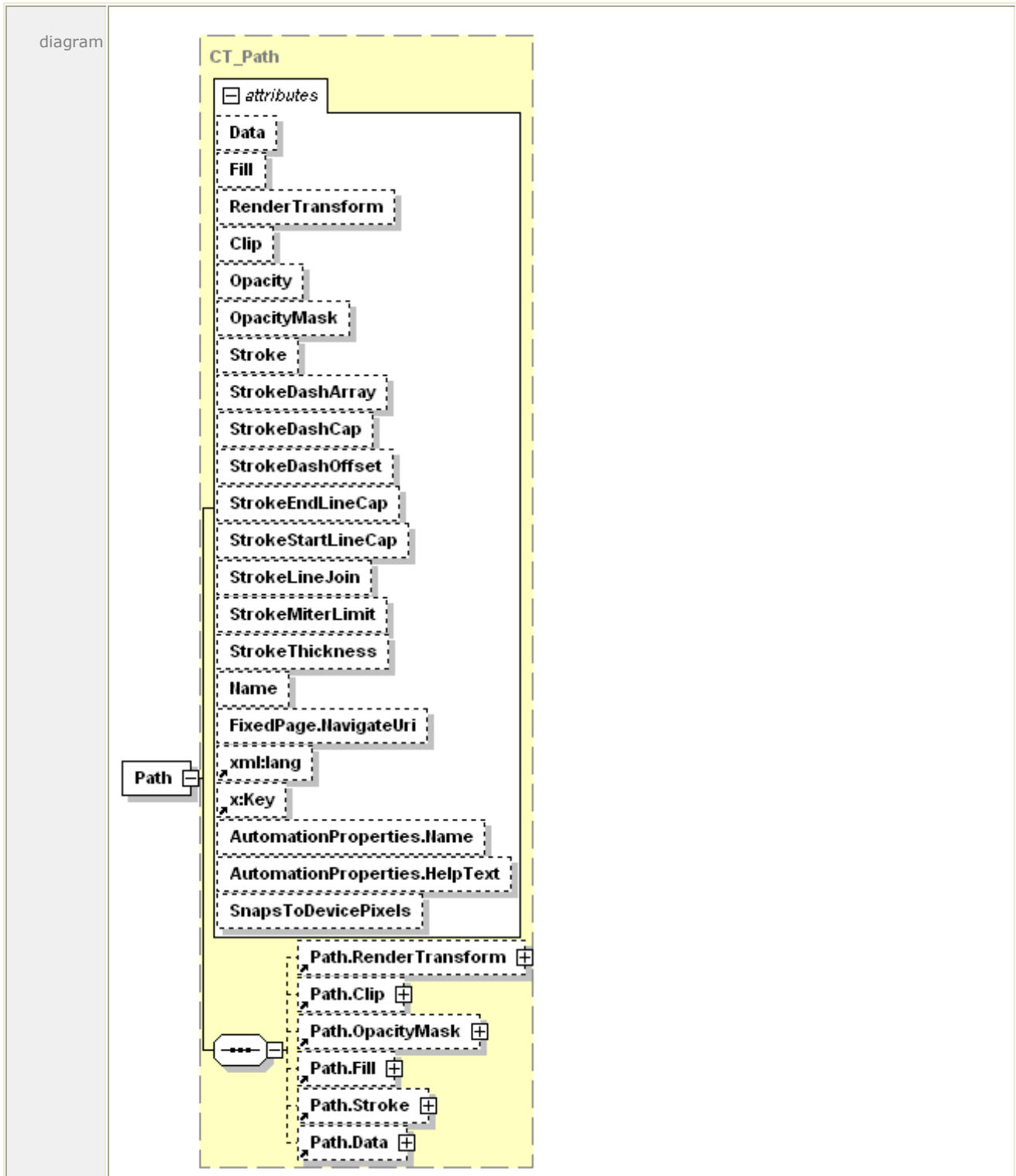
5

1 **11. Graphics**

2 Vector graphics are created using the <Path> element. A full set of properties is available to
3 describe the visual characteristics of the graphic. These characteristics include the fill, opacity,
4 clipping, rendering transformation, and various stroke details including thickness, fill, line join
5 style, line miter limit, line cap style, dash style, and dash cap style. The description of the
6 geometric area of the path (the geometry) is described by the Data property. Raster images are
7 included in fixed page markup by specifying a <Path> element filled with an <ImageBrush>.
8 3D graphics content is included in fixed-page markup by specifying a <Path> element filled
9 with a <Brush3D>. The <Brush3D> element usage with Markup Compatibility is defined in
10 Annex A.

1 **11.1 <Path> Element**

2 element **Path**



attributes	Name	Type	Use	Default	Fixed	Annotation
	Data	ST_RscRefAbbrGeomF				Describes the geometry of the path.
	Fill	ST_RscRefColor				Describes the brush used to paint the geometry specified by the Data property of the path.
	RenderTransform	ST_RscRefMatrix				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	ST_RscRefAbbrGeomF				Limits the rendered region of the element.
	Opacity	ST_ZeroOne		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	ST_RscRef				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	ST_RscRefColor				Specifies the brush used to draw the stroke.
	StrokeDashArray	ST_EvenArrayPos				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	ST_DashCap		Flat		Specifies how the ends of each dash are drawn. Valid values are Flat, Round, Square, and Triangle.
StrokeDashOffset	ST_Double		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	ST_LineCap		Flat		Defines the shape of the end of the last dash in a stroke. Valid values are Flat, Square, Round, and Triangle.
StrokeStartLineCap	ST_LineCap		Flat		Defines the shape of the beginning of the first dash in a stroke. Valid values are Flat, Square, Round, and Triangle.
StrokeLineJoin	ST_LineJoin		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	ST_GEOne		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	ST_GEZero		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 fixed page. It
 2 defines a single vector graphic to be rendered on a page. Some properties of the <Path>
 3 element are composable, meaning that the markings rendered to the page are determined by a
 4 combination of the property and all of the like-named properties of its parent and ancestor
 5 elements.

6 The Data property contains a geometric description of the area on which to apply a given effect.
 7 This description can take one of two forms: verbose or abbreviated. In the verbose form, the
 8 geometry is described in the <Path.Data> property element using the elements described
 9 in §11.2. In abbreviated form, it is described using abbreviated syntax in the Data attribute. For
 10 more information, see §11.2.3.

11 The <Path.Fill> property element describes the appearance of the area specified by the Data
 12 property. It contains a brush (see §13) that is used to fill the described areas. These can
 13 include a solid color, an image, a gradient, 3D graphics (Annex A), or a vector drawing pattern.

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

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

19 The transparency of the rendered <Path> element is controlled by the Opacity attribute. More
 20 complex transparency descriptions can be defined using the OpacityMask attribute to control the
 21 transparency of the brushes described by the Fill and Stroke properties.

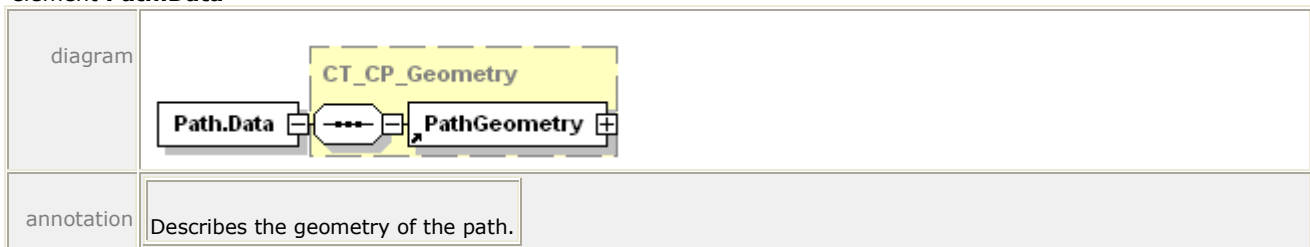
22 Consumers or viewers that perform anti-aliasing MAY “snap” those control points of the path
 23 that are situated on the path bounding box to whole device pixels if the ignorable
 24 SnapsToDevicePixels attribute is specified as true [O4.1].

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

28 For details on the Clip, Opacity, OpacityMask, and RenderTransform properties, see §14.

29 **11.1.1 <Path.Data> Element**

30 element **Path.Data**



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

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

```

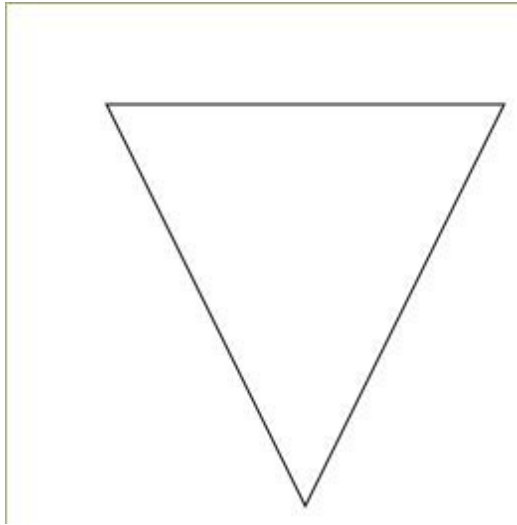
34     <Path Stroke="#000000" StrokeThickness="1">
35         <Path.Data>
36             <PathGeometry>
  
```

```

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

```

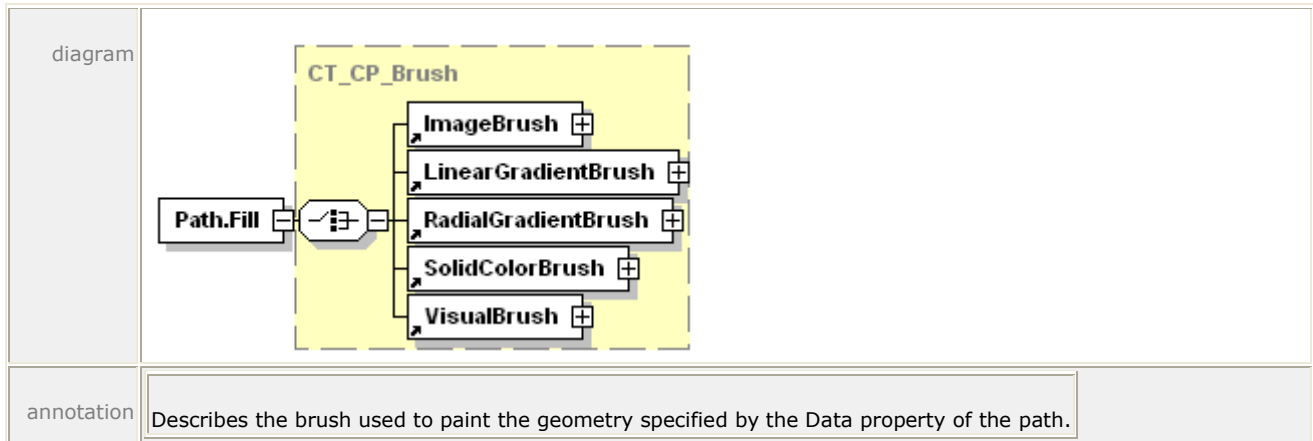
7 This markup produces the following results:



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 described by
 13 the Data property. This can be a solid color, an image, a gradient, or a 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>

```

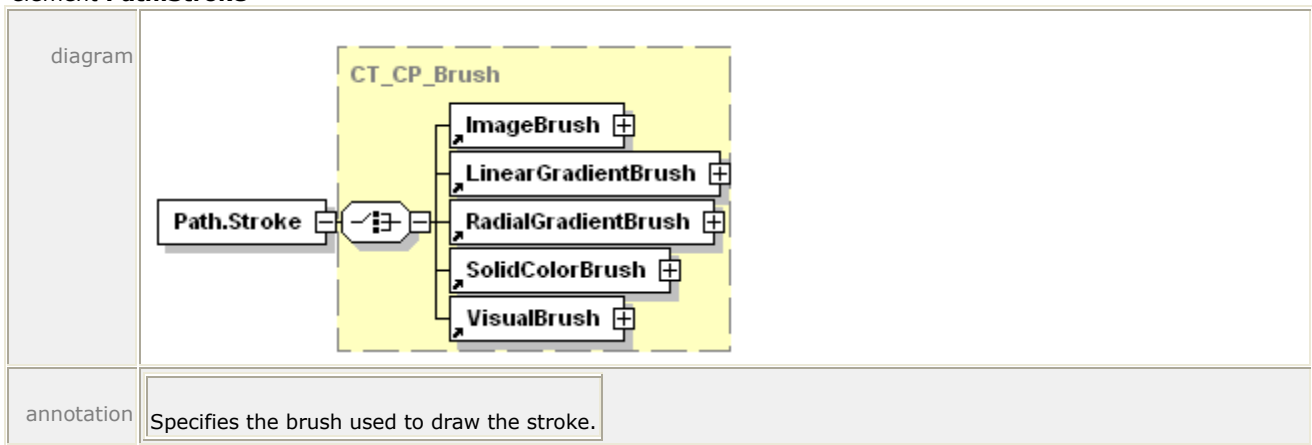
16 This markup produces the following result:



17
18 *end example]*

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

20 element **Path.Stroke**



1 The <Path.Stroke> property element describes the border of the path's geometry.
 2 <Path.Stroke> contains a brush. Only those segments of the path figure in the <Path.Data>
 3 element that set the IsStroked attribute to true (the default value if omitted) are stroked. If
 4 IsClosed is set to true, an extra segment will be stroked, connecting the last point in the path
 5 figure with the first point in the path figure.

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

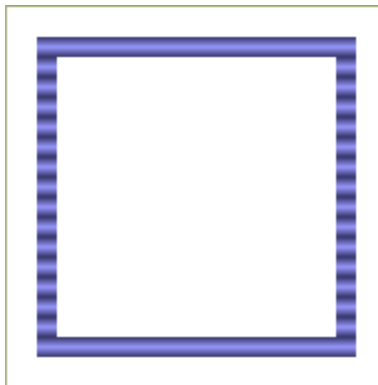
9 For more information, see §18.6.

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

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

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

23 This markup produces the following results:



24

25 *end example]*

26 11.2 Geometries and Figures

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

28 The smallest atomic unit in a geometry is a segment. Segments can be lines or curves. One or
 29 more segments are combined into a path figure definition. A path figure is a single shape
 30 comprised of continuous segments. One or more path figures collectively define an entire path
 31 geometry. A path geometry MAY define the fill algorithm to be used on the component path
 32 figures [[O4.2M2.72](#)].

- 1 A single path geometry can be used in the Data property of the <Path> element to describe its
- 2 overall geometry. A path geometry can also be used in the Clip property of the <Canvas>,
- 3 <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**

diagram						
attributes	Name	Type	Use	Default	Fixed	Annotation
	Figures	ST_AbbrGeom				Describes the geometry of the path.
	FillRule	ST_FillRule		EvenOdd		Specifies how the intersecting areas of geometric shapes are combined to form a region. Valid values are EvenOdd and NonZero.
	Transform	ST_RscRefMatrix				Specifies the local matrix transformation that is applied to all child and descendant elements of the path geometry before it is used for filling, clipping, or stroking.
	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 [M4.2].
annotation	Contains a set of <PathFigure> elements.					

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

8 The union of the path figures defines the interior of the path geometry according to the FillRule
 9 attribute as described in §11.2.1.2.

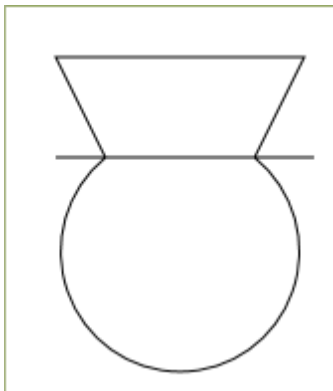
1 *Example 11-4. <PathGeometry> usage*

```

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

```

22 This markup produces the following results:



23

24 *end example]*

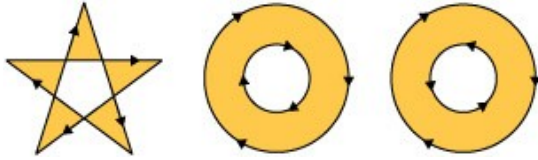
25 **11.2.1.2 FillRule Attribute**

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

30 **11.2.1.2.1 EvenOdd Fill Algorithm**

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

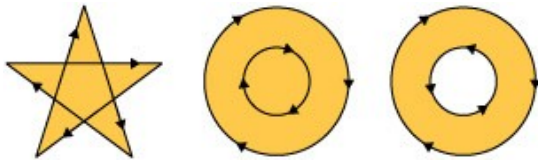
1 *Figure 11-1. Fill using EvenOdd algorithm*



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

10 *Figure 11-2. Fill using NonZero algorithm*



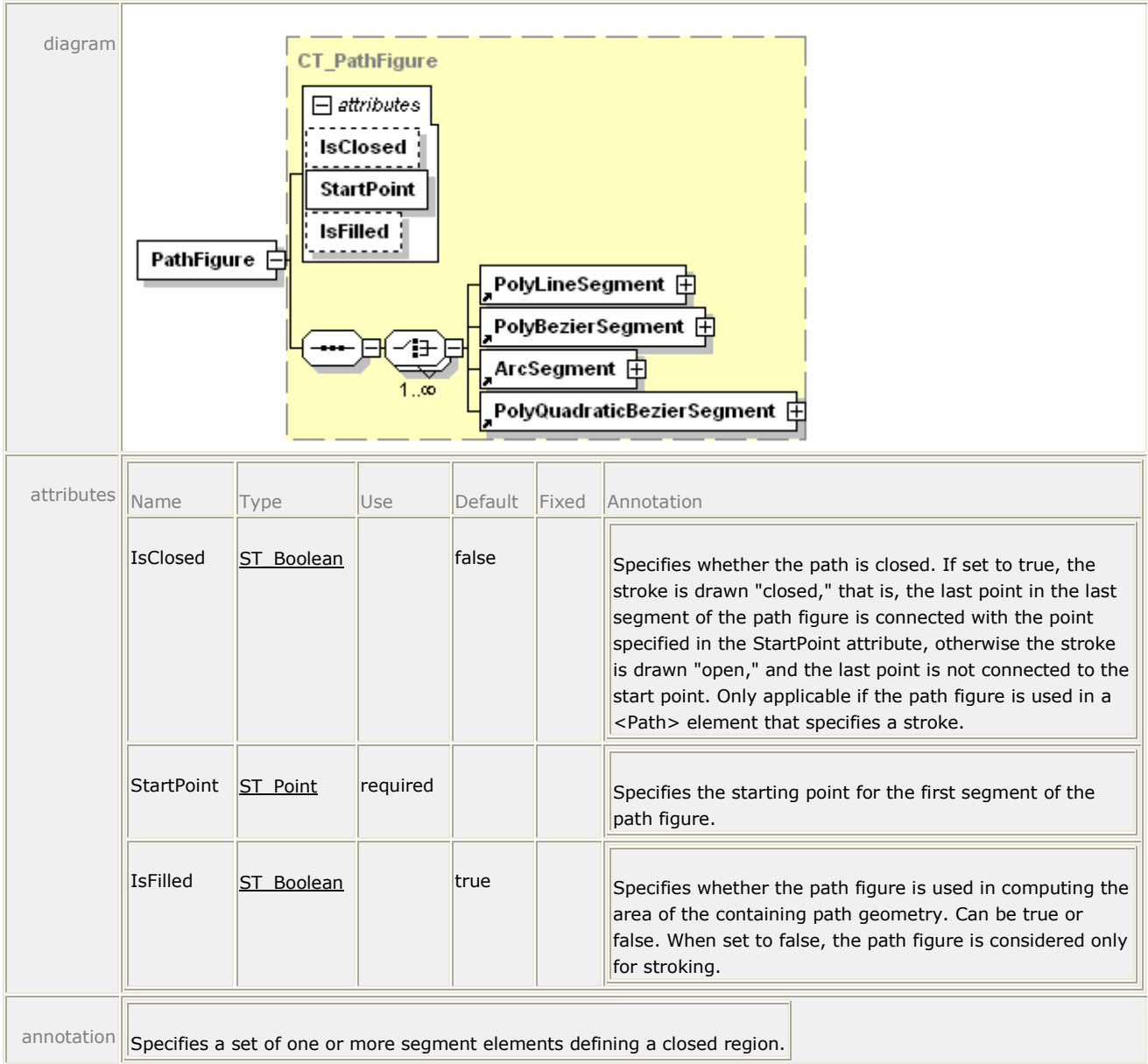
12 **11.2.1.3 Figures Attribute**

13 The <PathGeometry> element’s Figures attribute can be used to describe the path figures the
 14 geometry contains using abbreviated syntax (see §11.2.3) with the exception that the FillRule
 15 command MUST NOT be used [[M4.4M2-72](#)].

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. The
 5 segment elements define the shape of the path figure. The initial point of the first segment
 6 element is specified as the StartPoint attribute of the path figure. The last point of each segment
 7 element is the first point of the following segment 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]. If they are
 7 specified as zero-length, they are not drawn. For full details of the behavior in cases such as
 8 those involving line caps, see §18.

9 **11.2.2.2 <ArcSegment> Element**

10 element **ArcSegment**

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

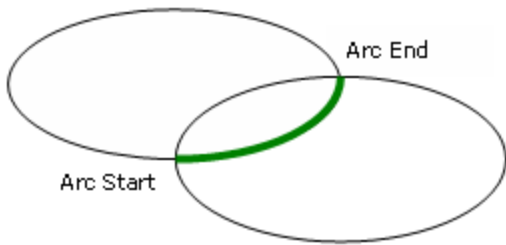
1 The <ArcSegment> element describes an elliptical arc. It is geometrically defined by the
 2 intersection of two ellipses that have the same x radius and y radius. The ellipses intersect at
 3 the starting and ending points of the arc.

4 *Table 11-1. 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. [<i>Example: "100,50" end example</i>]
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*

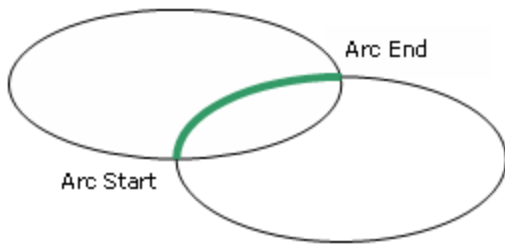
2 `IsLargeArc = false; SweepDirection = Counterclockwise`



3

4 *Figure 11-4. Arc choice B*

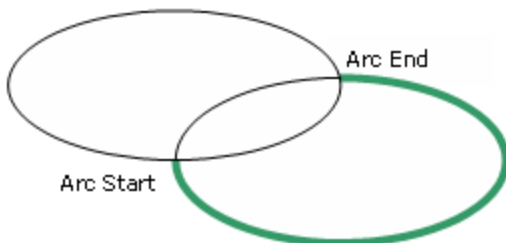
5 `IsLargeArc = false; SweepDirection = Clockwise`



6

7 *Figure 11-5. Arc choice C*

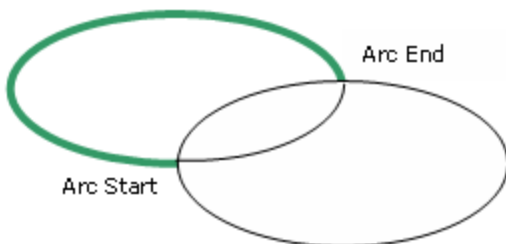
8 `IsLargeArc = true; SweepDirection = Counterclockwise`



9

10 *Figure 11-6. Arc choice D*

11 `IsLargeArc = true; SweepDirection = Clockwise`



12

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

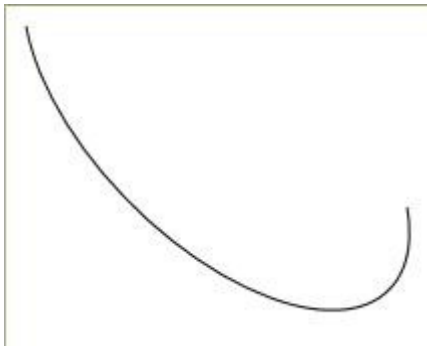
14 `<Path Stroke="#000000" StrokeThickness="1">`
 15 `<Path.Data>`

```

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 values on an
18 <ArcSegment> element:

- 19 • If the arc is impossible to render given the combination of radii specified in the Size
20 attribute and the angle of rotation specified in the RotationAngle attribute, the ellipses are
21 scaled equally until there is exactly one solution that satisfies the arc requirements to
22 pass through the specified Point attribute.
- 23 • If the Point attribute is the same as the previous point in the path figure, the segment is
24 omitted.
- 25 • If either the x or y radius in the Size attribute is 0, the segment is rendered as a poly line
26 segment with a single line segment to the x,y coordinates specified by the Point
27 attribute.
- 28 • The x or y radius in the Size attribute MUST NOT be negative [[M4.5M2.72](#)].
- 29 • If the RotationAngle value is greater than 360, it is replaced by the value of the
30 RotationAngle modulo 360. If it is less than 0, it is replaced with a value normalized to the
31 range 0–360.

1 **11.2.2.3 <PolyBezierSegment> Element**2 element **PolyBezierSegment**

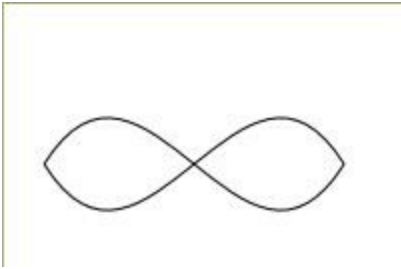
diagram						
attributes	Name	Type	Use	Default	Fixed	Annotation
	Points	<u>ST_Points</u>	required			Specifies control points for multiple Bézier segments. Coordinate values within each pair are comma-separated and additional whitespace can appear. Coordinate pairs are separated from other coordinate pairs by whitespace.
	IsStroked	<u>ST_Boolean</u>		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 curves are
4 drawn from the previous point in the path figure or the previous Bézier curve in the segment
5 and terminate at the third point (x_{3n}, y_{3n}) in the Points attribute (where n is the curve being
6 drawn). The tangents and curvature of each Bézier curve are controlled by the first two control
7 points (x_{3n-2}, y_{3n-2} and x_{3n-1}, y_{3n-1}) in the Points attribute. The Points attribute contains a multiple
8 of three whitespace-delimited pairs of comma-delimited x,y values.

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

```
10 <Path Stroke="#000000" StrokeThickness="1">
11 <Path.Data>
12 <PathGeometry>
13 <PathFigure StartPoint="20,80">
14 <PolyBezierSegment Points="70,0 120,160 170,80 120,0 70,160
15 20,80" />
16 </PathFigure>
17 </PathGeometry>
18 </Path.Data>
19 </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	<u>ST_Points</u>	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 can appear. Coordinate pairs are separated from other coordinate pairs by whitespace.
	IsStroked	<u>ST_Boolean</u>		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 arbitrary number
7 of individual vertices. The Points attribute defines the vertices and contains whitespace-
8 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>

```

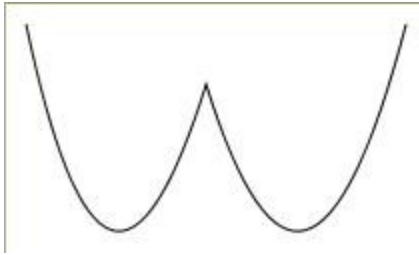

1 *Example 11-8. <PolyQuadraticBezierSegment> usage*

```

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 is drawn from
 17 the last point in the last segment of the `<PathFigure>` element to the `StartPoint` attribute of the
 18 `<PathFigure>` element. If the `IsClosed` attribute is omitted, its default setting is `false`.

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

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

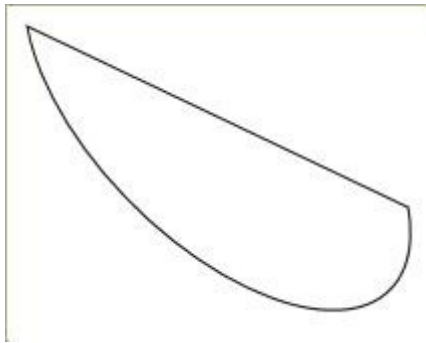
22 The following markup shows the arc segment as shown in Example 11-5 with the `IsClosed`
 23 attribute of the `<PathFigure>` element set to `true`.

```

24     <Path Stroke="#000000" StrokeThickness="1">
25         <Path.Data>
26             <PathGeometry>
27                 <PathFigure StartPoint="10,10" IsClosed="true">
28                     <ArcSegment
29                         Size="100,50"
30                         RotationAngle="45"
31                         IsLargeArc="true"
32                         SweepDirection="Counterclockwise"
33                         Point="200,100" />
34                 </PathFigure>
35             </PathGeometry>
36        </Path.Data>
37    </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 figures
6 comprised of multiple segments [[O4.3M2.72](#)]. A geometry is specified with an optional FillRule
7 command (not allowed in the Figures attribute of a <PathGeometry> element) followed by one
8 or more figure definitions. Figure definitions are specified with a Move command, a set of
9 drawing commands to create segments, and an optional Close command to create a closing
10 segment. Drawing commands include:

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

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

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

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

27 If entering more than one drawing command of the same type sequentially, the duplicate
28 command entry MAY be omitted [[O4.4M2.72](#)]. [*Example*: "L 100,200 300,400" is equivalent to
29 "L 100,200 L 300,400". *end example*] The current endpoint is determined as though each
30 command appeared individually.

31 Values specifying coordinates can be real numbers.

32 For more information, see §B.

1 Table 11-2. 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 [M4.6M2.72]. This command MUST NOT be specified in the value of the Figures attribute of the <PathGeometry> element [M4.7M2.72]. [Example: F 0 end example]	<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 [O4.5M2.72]. The first figure in a geometry MUST begin with a Move command [M4.8M2.72]. 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 end point of the previous figure [O4.6M2.72]. [Example: M 1.0,1.5 end example]	<PathFigure> StartPoint attribute
Line	L x, y or l x, y	Draws a straight line from the current point to the specified point. [Example: L 20,30 end example]	<PolyLineSegment> element
Horizontal Line	H x or h x	Draws a horizontal line from the current endpoint to the specified x coordinate. [Example: H 90 end example]	<PolyLineSegment> element
Vertical Line	V y or v y	Draws a vertical line from the current endpoint to the	<PolyLineSegment> element

Name	Syntax	Description	Non-Abbreviated Equivalent
	<code>v y</code>	specified <i>y</i> coordinate. [<i>Example: v 90 end example</i>]	
Cubic Bézier Curve	<code>C x₁, y₁ x₂, y₂ x₃, y₃</code> or <code>c x₁, y₁ x₂, y₂ x₃, y₃</code>	Draws a cubic Bézier curve from the current endpoint to the specified point (<i>x₃, y₃</i>) using the two specified control points (<i>x₁, y₁</i> and <i>x₂, y₂</i>). The first control point determines the initial direction (tangent) of the curve, and the second determines the terminating direction (tangent) of the curve. [<i>Example: C 100, 200 200, 400 300, 200 end example</i>]	<PolyBezierSegment> element
Quadratic Bézier Curve	<code>Q x₁, y₁ x₂, y₂</code> or <code>q x₁, y₁ x₂, y₂</code>	Draws a quadratic Bézier curve from the current endpoint to the specified point (<i>x₂, y₂</i>) using the specified control point (<i>x₁, y₁</i>). [<i>Example: q 100, 200 300, 200 end example</i>]	<PolyQuadraticBezierSegment> element
Smooth Cubic Bézier Curve	<code>S x₁, y₁ x₂, y₂</code> or <code>s x₁, y₁ x₂, y₂</code>	Draws a cubic Bézier curve from the current endpoint to the specified point (<i>x₂, y₂</i>). 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 <i>x₁, y₁</i> . [<i>Example: S 100, 200 200, 300 end example</i>]	<PolyBezierSegment> element
Elliptical Arc	<code>A x_r, y_r r_x fArc fSweep</code> <i>x, y</i> or <code>a x_r, y_r r_x fArc fSweep</code> <i>x, y</i>	Draws an elliptical arc from the current endpoint to the specified point (<i>x, y</i>). The size and orientation of the ellipse are defined by <i>x_r, y_r</i> , <i>r_x, r_y</i> defines the <i>x</i> radius, <i>y_r</i> defines the <i>y</i> radius, and <i>r_x</i> defines the <i>x</i> -axis rotation in degrees,	<ArcSegment> element

Name	Syntax	Description	Non-Abbreviated Equivalent
		<p>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. f_{Arc} and f_{Sweep} indicate which arc to use.</p> <p>Of the four candidate arc sweeps, two represent large arcs with sweeps of 180° or greater, and two represent smaller arcs with sweeps less than 180°.</p> <p>If f_{Arc} is 1, one of the two larger arc sweeps is chosen. If f_{Arc} is 0, one of the smaller arc sweeps is chosen. No other values of f_{Arc} are valid.</p> <p>If f_{Sweep} is 1, the arc is drawn in a positive-angle (clockwise) direction. If f_{Sweep} is 0, the arc is drawn in a negative-angle (counter-clockwise) direction. No other values of f_{Sweep} are valid. [Example: a 200,70 10 0 1 100,100 end example]</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 abbreviated
3 syntax:

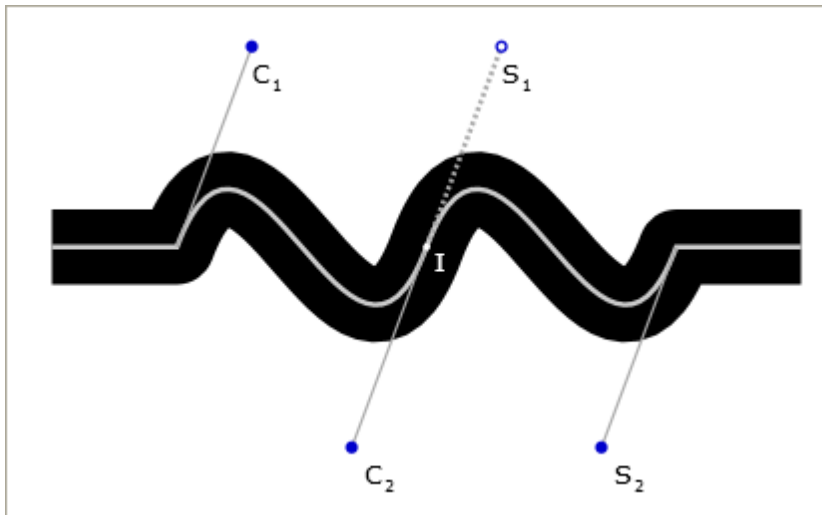
1 <Path Stroke="#000000" Data="M 100,100 L 300,100 L 200,300 z" />
 2 *end example]*

3 **11.2.3.1 Smooth Bézier Curve Abbreviated Syntax**

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

10 *Example 11-11. Smooth Bézier curve*

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



16

17 The above diagram is generated with the following markup:

```

18     <Canvas RenderTransform="1.25,0,0,1.25,-40,20" >
19       <!-- Main Path -->
20       <Path Stroke="#000000" StrokeThickness="30" StrokeLineJoin="Round"
21       Data="M50,80 L100,80 C130,0 170,160 200,80 S270,160 300,80 L350,80"/>
22       <Path Stroke="#CCCCCC" StrokeThickness="2"
23       Data="M50,80 L100,80 C130,0 170,160 200,80 S270,160 300,80 L350,80"/>
24       <!-- C1 -->
25       <Path Stroke="#AAAAAA" StrokeThickness="1" Data="M 100,80 L 130,0" />
26       <Path Stroke="#0000CC" StrokeThickness="5" StrokeStartLineCap="Round"
27       StrokeEndLineCap="Round" Data="M 130,0 L 130,0" />
28       <Glyphs Fill="#000000" UnicodeString="C" OriginX="130" OriginY="15"
29       FontUri="../Resources/Fonts/Verdana.ttf" FontRenderingEmSize="10"
30       />
31       <Glyphs Fill="#000000" UnicodeString="1" OriginX="138" OriginY="18"
32       FontUri="../Resources/Fonts/Verdana.ttf" FontRenderingEmSize="6"
33       />

```

```

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

```

42 *end example]*

43 11.2.3.2 Relative Commands and Curve Control Points

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

46 *Example 11–12. Relative commands and curves*

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

```

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

```

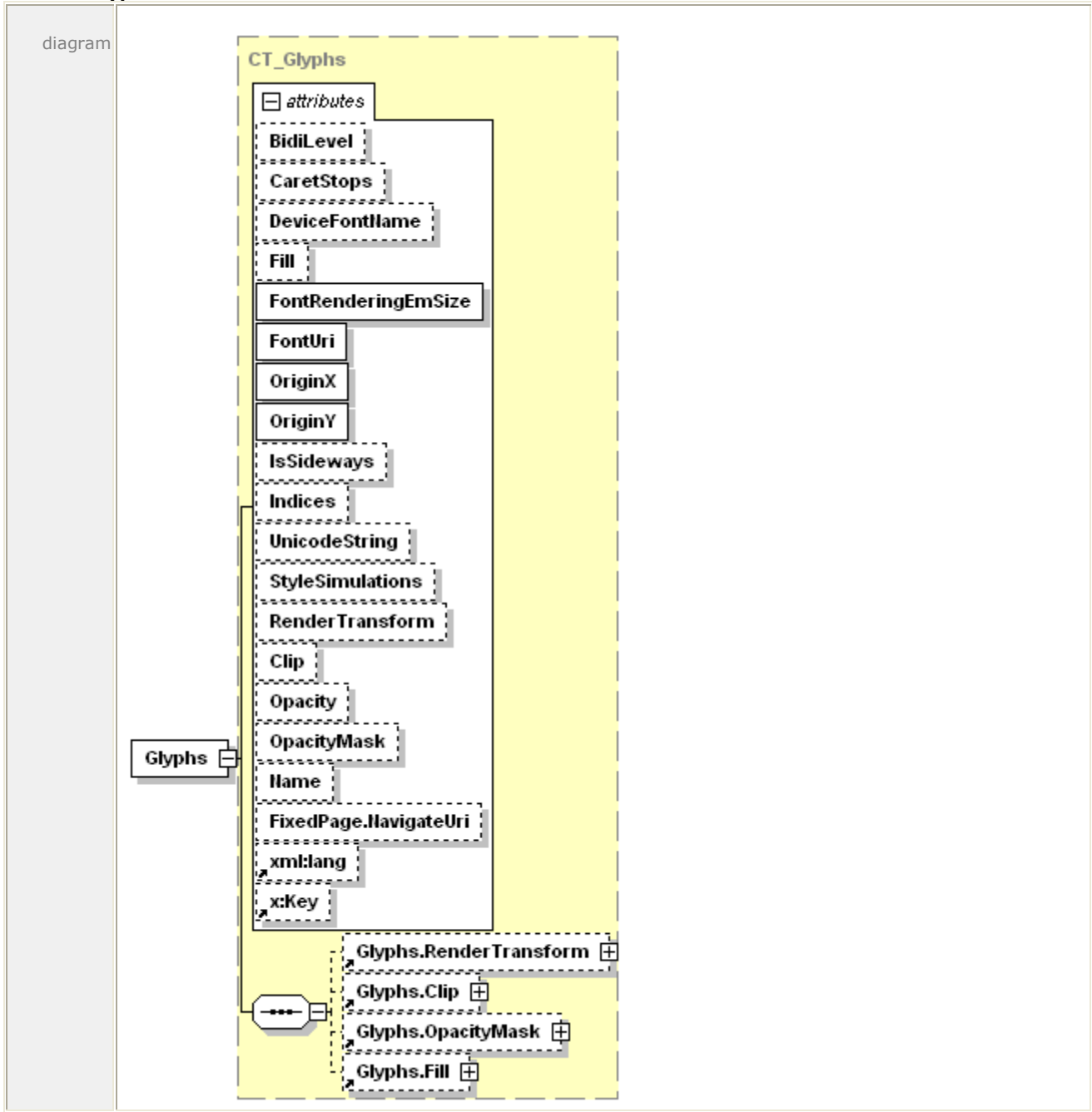
- 1 This markup describes the same shape, using relative commands:
- 2 <Path Stroke="#000000" Data="M 50,20 l 100,0 c 100,55 20,110 -30,80
- 3 c -50,-30 -30,10 10,60 q -130,-10 -80,-140" />
- 4 *end example]*

1 **12. Text**

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

1 **12.1 <Glyphs> Element**

2 element **Glyphs**



attributes	Name	Type	Use	Default	Fixed	Annotation
		BidiLevel			0	

					<p>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	ST_CaretStops				<p>Identifies the positions within the sequence of Unicode characters at which a text-selection tool can 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].</p> <p>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	ST_UnicodeString				<p>Uniquely identifies a specific device font. The identifier is typically defined by a hardware vendor or font vendor.</p>
Fill	ST_RscRefColor				<p>Describes the brush used to fill the shape of the rendered glyphs.</p>
FontRenderingEmSize	ST_GEZero	required			<p>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.</p>
FontUri	xs:anyURI	required			<p>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),</p>

					the fragment portion of the URI is a 0-based index indicating which font face of the TrueType Collection should be used.
OriginX	<u>ST_Double</u>	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	<u>ST_Double</u>	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	<u>ST_Boolean</u>		false		Indicates that a glyph is turned on its side, with the origin being defined as the top center of the unturned glyph.
Indices	<u>ST_Indices</u>				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 instantiate an error condition generate-an-error [M5.2].
UnicodeString	<u>ST_UnicodeString</u>				Contains the string of text rendered by the <Glyphs> element. The text is specified as Unicode code points.
StyleSimulations	<u>ST_StyleSimulations</u>		None		Specifies a style simulation. Valid values are None, ItalicSimulation, BoldSimulation, and BoldItalicSimulation.
RenderTransform	<u>ST_RscRefMatrix</u>				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	ST_RscRefAbbrGeomF				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	ST_ZeroOne		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	ST_RscRef				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	ST_Name				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

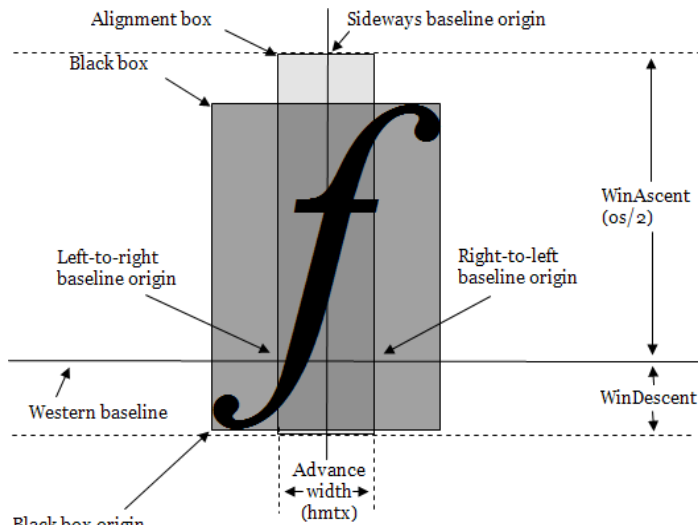
						dictionary [M5.3].
annotation	Represents a run of text from a single font.					

- 1 The <Glyphs> element represents a run of uniformly-formatted text from a single font. It
- 2 provides information necessary for accurate rendering and supports search and selection
- 3 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 markings
- 6 rendered to the page are determined by a combination of the property and all the like-named
- 7 properties of the <Glyphs> element’s parent and ancestor elements. For details, see §14.

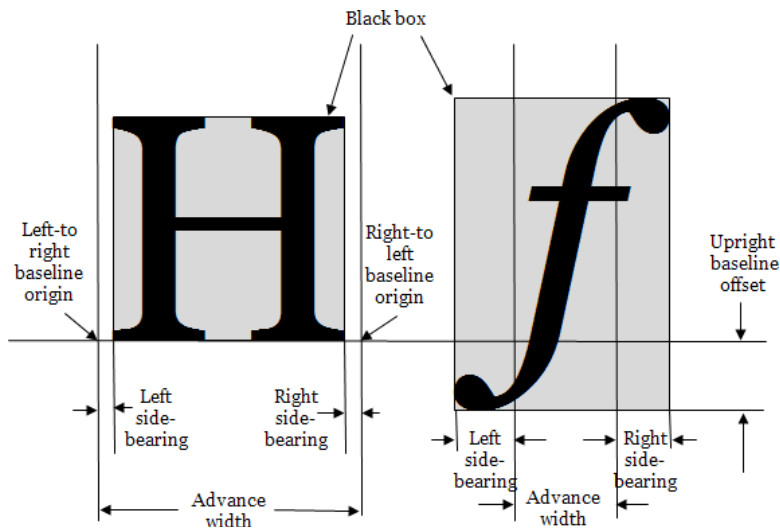
1 **12.1.1 Glyph Metrics**

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

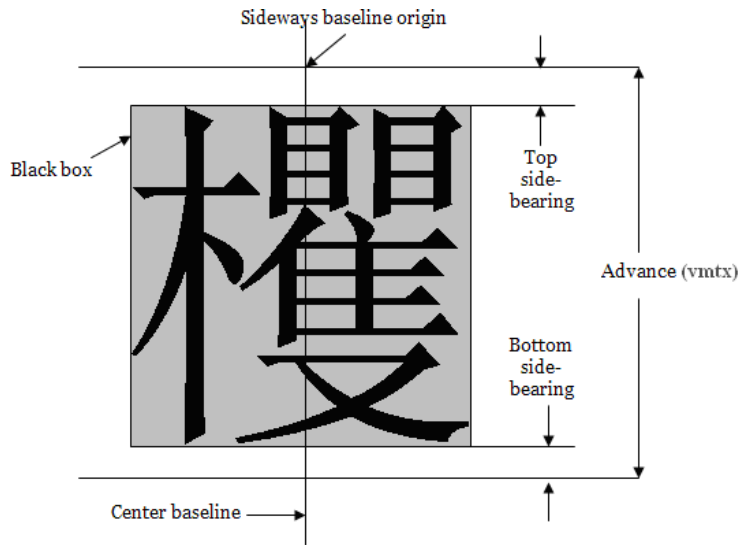
4 *Figure 12-1. Glyph metrics*[GG1]



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 can be attached
 4 to base glyphs. Base glyphs usually have an advance width that is non-zero, and a 0,0 [glyph](#)
 5 offset vector. Combining marks usually have a zero advance width. The [glyph](#) offset vector can
 6 be used to adjust the position of a combining mark and, therefore, can have a non-0,0 value for
 7 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 in the run,
 10 this is the origin of the run.
- 11 • *Advance Width*. The advance width for each glyph provides the origin of the next glyph
 12 relative to the origin of the current glyph. The advance vector is drawn in the direction
 13 of the run progression.
- 14 • *Glyph Offset (Base or Mark)*. The glyph offset vector ([as set by uOffset and vOffset in the](#)
 15 [Indices attribute; see §12.1.3](#)) adjusts the position of this glyph relative to its nominal
 16 origin. The orientation of the glyph offset vector is not affected by the value of the
 17 lsSideways attribute, but is affected 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 single UTF-16
 20 code unit and has a single corresponding glyph representation in the font. More complex
 21 mapping scenarios are common in non-Latin scripts: a single Unicode scalar value can map to
 22 two UTF-16 code units, multiple UTF-16 code units can map to a single glyph, single UTF-16
 23 code units can map to multiple glyphs based on context, and multiple UTF-16 code units can
 24 map indivisibly to multiple glyphs. In these cases, the clusters of UTF-16 code units are mapped
 25 using a cluster map.

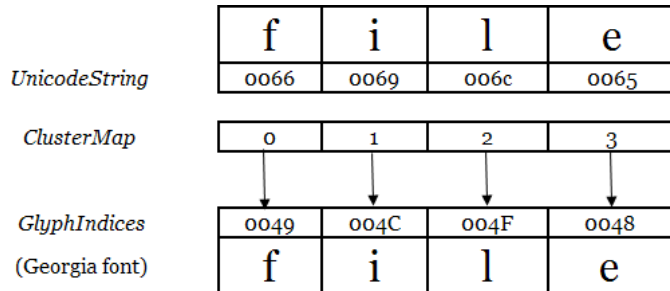
26 The cluster map contains one entry for each UTF-16 code unit in the UnicodeString attribute.
 27 Each entry specifies the offset of the first glyph that represents the cluster of UTF-16 code
 28 units.

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 entries are 0,
3 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.



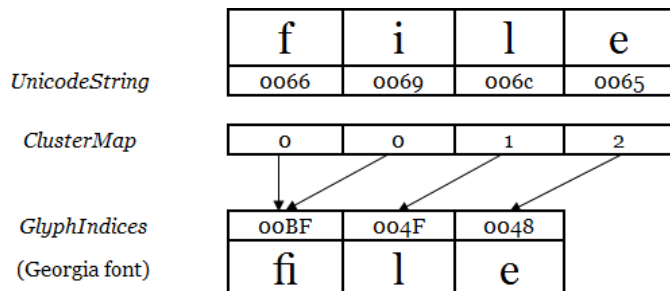
6
7 *end example]*

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 UTF-16 code
10 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.



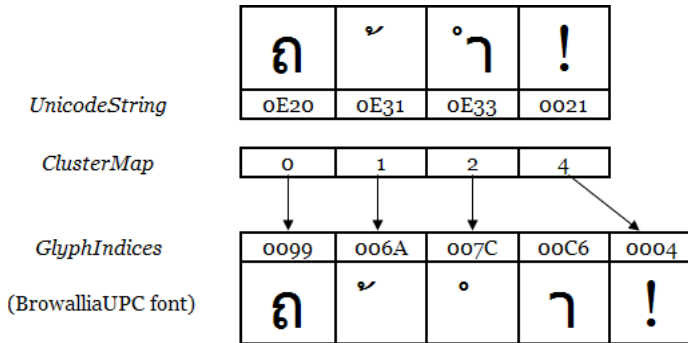
13
14 *end example]*

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 map for that
 3 UTF-16 code unit references the first glyph in the Indices attribute that represents that UTF-16
 4 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 character (the
 7 ring), and a part that sits to the right of the base character (the hook). When Thai text is
 8 micro-justified, the hook is spaced apart from the base character, while the ring remains on top
 9 of the base character. Many fonts encode the ring and the hook as separate glyphs.



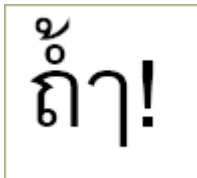
10

11 The markup appears as follows:

```

12 <Glyphs
13   FontUri="../Resources/Fonts/browau.ttf"
14   UnicodeString="&#xe20;&#xe31;&#xe33;&#x21;"
15   Indices="153;106,,16;(1:2)124;198;4"
16   OriginX="10" OriginY="60"
17   FontRenderingEmSize="70"
18   Fill="#000000"/>
    
```

19 The markup above is rendered as follows:



20

21 *end example]*

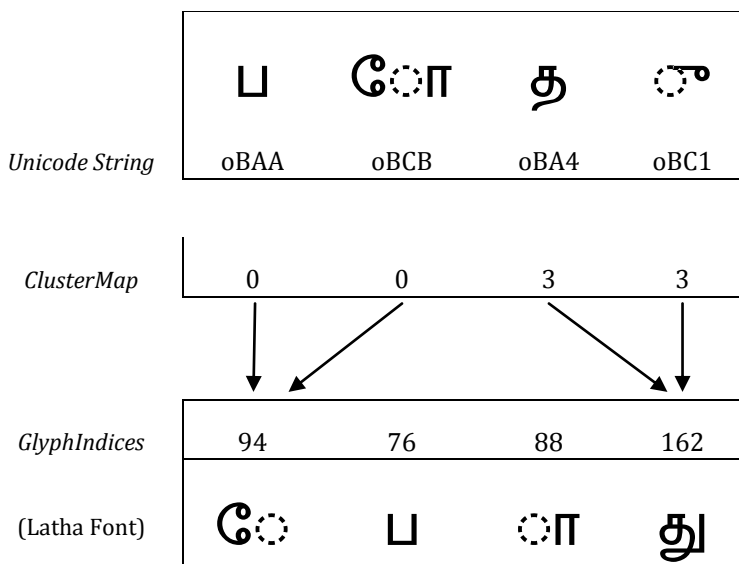
12.1.2.4 Many-to-Many Mappings

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

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

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

12



13

The markup appears as follows:

```

15 <Glyphs
16   FontUri="../Resources/Fonts/latha.ttf"
17   UnicodeString="#&#xbaa;&#x bcb;&#x ba4;&#x bc1;"
18   Indices="(2:3)94;76;88;(2:1)162"
19   OriginX="10" OriginY="120"
20   FontRenderingEmSize="40"
21   Fill="#000000"/>

```

The markup above is rendered as follows:

23



1 *end example]*

2 **12.1.3 Indices Attribute**

3 The <Glyphs> element MAY have an Indices attribute [O5.7M2-72]. The glyph specifications
4 within the Indices attribute are OPTIONAL [O5.8M2-72]. The GlyphIndex portion of the Indices
5 attribute MAY be used to specify a series of glyphs, complex character-to-glyph cluster
6 mappings, or a combination of both [O5.9M2-72]. The Indices attribute MAY also include glyph
7 placement information [O5.10M2-72].

8 Within the Indices attribute, each glyph specification is separated by a semicolon. The Indices
9 attribute MUST adhere to the glyph specification syntax as follows [M5.25M2-72]:

```
10   GlyphIndices   = *1GlyphMapping *( ";" *1GlyphMapping )
11   GlyphMapping  = *1([ClusterMapping] GlyphIndex) [GlyphMetrics]
12   ClusterMapping = "(" ClusterCodeUnitCount [":" ClusterGlyphCount] ")"
13   ClusterCodeUnitCount = 1*DIGIT
14   ClusterGlyphCount   = 1*DIGIT
15   GlyphIndex        = *DIGIT
16   GlyphMetrics      = "," *1AdvanceWidth ["," *1uOffset ["," vOffset]]
17   AdvanceWidth     = ["+"] RealNum
18   uOffset           = ["+" | "-"] RealNum
19   vOffset           = ["+" | "-"] RealNum
20   RealNum           = ((1*DIGIT [ "." 1*DIGIT ] | ( "." 1*DIGIT )) [Exponent]
21   Exponent          = *1( ("E"|"e") ("+"|" -") 1*DIGIT )
```

22 The sum of the code unit counts for all the GlyphMapping entries in the Indices attribute MUST
23 NOT exceed the number of UTF-16 code units in the UnicodeString attribute if the UnicodeString
24 attribute is specified and does not contain an empty value (" or "{}"). If a ClusterMapping is
25 not specified within a GlyphMapping entry, the code unit count is 1 [M5.4]. If the Indices
26 attribute specifies a GlyphIndex that does not exist in the font, the consumer MUST **generate**
27 **an error** [instantiate an error condition](#) [M5.24M5-4]. If the Indices attribute is specified, the
28 values provided MUST be used in preference to values determined from the UnicodeString
29 attribute alone [M5.23].

30 *Table 12-3. Glyph specifications*

Name	Description
GlyphIndex	Index of the glyph (16-bit) in the physical font. The entry MAY be empty [O5.11M2-72], 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]. In cases where character-to-glyph mappings are not one-to-one, a cluster mapping specification precedes the glyph index (further described below).
AdvanceWidth	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. 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 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 [M5.25M2.72]. The right-to-left writing direction can be specified using the `BidiLevel` 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 12–5. Using indices to specify advance width*

2 The following `Indices` attribute specifies that the seventh glyph in the Unicode string has an
3 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 the cluster
8 [O5.12M2.72].

9 Empty `Indices` attribute values indicate that the corresponding UTF-16 code unit within the
10 Unicode string has a one-to-one relationship with the glyph index as specified by the character
11 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 12–4. Portions of the cluster specification*

Name	Description
<code>ClusterCodeUnitCount</code>	Number of UTF-16 code units that combine to form this cluster. One or more code units can be specified. Default value is 1.

ClusterGlyphCount	Number of glyph indices that combine to form this cluster. One or more indices can be specified. Default value is 1.
-------------------	--

1 *Example 12-6. 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 units in the
3 Unicode string should be replaced by a single glyph having an index of 191:

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

5 *end example]*

6 **12.1.4 UnicodeString Attribute**

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

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

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

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

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

33 Producers MAY choose to generate UnicodeString attribute values that are not normalized by any
34 Unicode-defined algorithm [O5.2]. Because advance-widths, glyph indices, and caret-stops are
35 associated with the generated Unicode string, consumers MUST NOT normalize the UnicodeString
36 attribute value to produce an internal representation [M5.11]. See §9.1.7.5 for details and
37 exceptions.

38 **12.1.5 StyleSimulations Attribute**

39 Synthetic style simulations can be applied to the shape of the glyphs by using the
40 StyleSimulations attribute. Style simulations can be applied in addition to the designed style of a

1 font. The default value for the `StyleSimulations` attribute is `None`, in which case the shapes of
2 glyphs are not modified from their original design.

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

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

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

22 When `StyleSimulations` is specified as `BoldItalicSimulation`, both `BoldSimulation` and
23 `ItalicSimulation` are applied, in order.

24 **12.1.6 IsSideways Attribute**

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

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

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

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

1 **12.1.6.1 Calculating Sideways Text Origin and Advance Width**

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

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

8
$$\text{TopOriginX} = \text{hmtx.advanceWidth}[\text{GlyphIndex}] / 2$$

9 If the font is a CFF OpenType font, the y origin of the unturned glyph is determined from the
10 vertical origin (vorg) table for the font, which can be specified for a particular glyph index but
11 falls back to the default vertical origin if the glyph index is not present in the vertical origin
12 table. This formula is expressed as follows (in pseudocode):

13
$$\text{TopOriginY} = \text{vorg.vertOriginY}[\text{glyphIndex}]$$

14 or:

15
$$\text{TopOriginY} = \text{vorg.defaultVertOriginY}$$

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

20
$$\text{TopOriginY} = \text{glyf.yMax}[\text{glyphIndex}] + \text{vmtx.topSideBearing}[\text{glyphIndex}]$$

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

24 If the vertical metrics font table does not exist but the "OS/2" metrics table does exist and is at
25 least 78 bytes long, the "OS/2" table is consulted and the sTypoAscender and sTypoDescender
26 values are used, as follows (in pseudocode):

27
$$\text{TopOriginY} = \text{os/2.sTypoAscender}$$

28
$$\text{Descender} = \text{abs}(\text{os/2.typoDescender})$$

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

31
$$\text{TopOriginY} = \text{hhea.Ascender}$$

32
$$\text{Descender} = \text{abs}(\text{hhea.Descender})$$

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

35
$$\text{AdvanceWidth} = \text{TopOriginY} + \text{Descender}$$

36 **12.1.6.2 IsSideways and BidiLevel Effects on Glyph Positioning**

37 Right-to-left text (BidiLevel attribute [set to an odd value](#)~~value of 1~~) changes the direction of the
38 AdvanceWidth and uOffset (horizontal offset) values of the Indices attribute, as well as the
39 position of the glyph origin. Vertical text (IsSideways attribute set to true) changes the position
40 of the glyph origin.

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

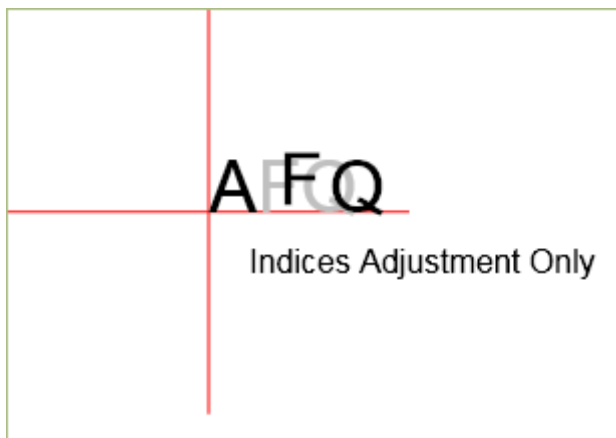
3 *Table 12-5. IsSideways and BidiLevel effects on origin placement*

IsSideways	BidiLevel	Glyph origin	Direction of advance width and positive uOffset
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>	

4 *Example 12-7. Text with positive uOffset and vOffset Indices values*

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

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



- 13
14 *end example]*

15 *Example 12-8. Right-to-left text (odd BidiLevel)*

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

```

1 <Glyphs Fill="#000000" FontRenderingEmSize="48"
2   OriginX="100" OriginY="100"
3   UnicodeString="AFQ"
4   Indices=";,100,30,10;"
5   BidilLevel="1"
6   FontUri=" ../Resources/Fonts/Arial.ttf" />

```



7
8 *end example]*

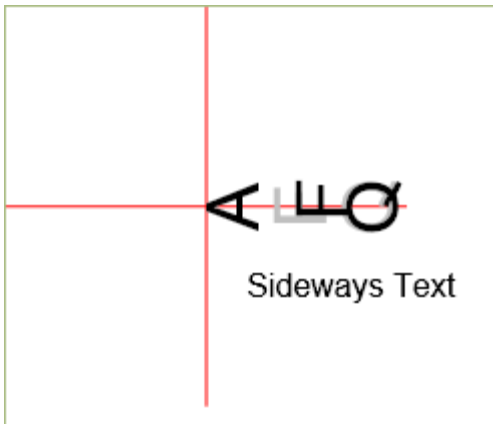
9 *Example 12-9. Sideways text (IsSideways set to true)*

10 This example shows the IsSideways attribute set to true. The BidilLevel MUST be even when the
 11 IsSideways attribute is set to true [M5.15]. Note that the origin has changed to be the top-
 12 center of the first glyph, with each glyph rotated 90° counter-clockwise. The interpretation of
 13 the advance direction and uOffset and vOffset values in the Indices attribute are otherwise
 14 unchanged.

```

15 <Glyphs Fill="#000000" FontRenderingEmSize="48"
16   OriginX="100" OriginY="100"
17   UnicodeString="AFQ"
18   Indices=";,100,30,10;"
19   IsSideways="true"
20   FontUri=" ../Resources/Fonts/Arial.ttf" />

```

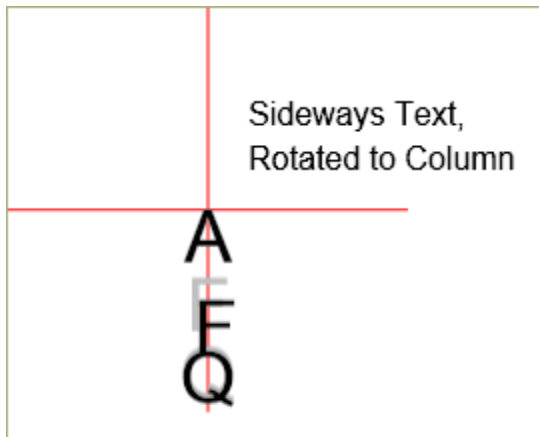


21
22 *end example]*

1 *Example 12-10. Vertical text*

2 The markup for this example matches the previous example, with the addition of a render
 3 transformation to rotate and position the element as vertical text. For more information on
 4 render transformations, see §14.4.

```
5 <Glyphs Fill="#000000" FontRenderingEmSize="48"
6   OriginX="100" OriginY="100"
7   UnicodeString="AFQ"
8   Indices=";100,30,10;"
9   IsSideways="true"
10  FontUri="../Resources/Fonts/Arial.ttf"
11  RenderTransform="0,1,-1,0,200,0" />
```



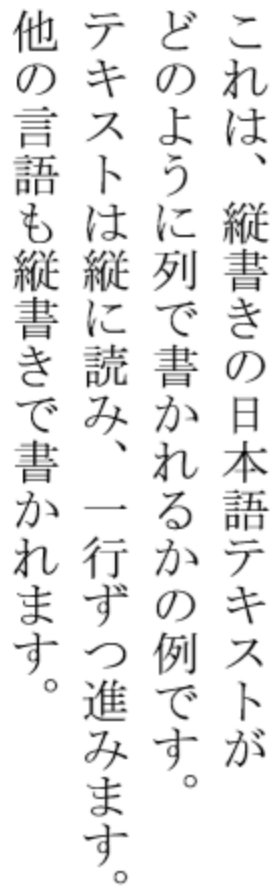
12

13 *end example]*14 *Example 12-11. Japanese vertical text*

15 This example demonstrates a real-world usage of vertical text. Japanese text is shown below
 16 where the text is read down each column, from right to left across the page. The `IsSideways`
 17 attribute is set to `true`, thus rotating the each glyph 90° counter-clockwise. Then, the
 18 `RenderTransform` attribute (see §14.4) rotates the overall block of text 90° clockwise to achieve
 19 the final result of columns of text.

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

- 1 This markup is rendered as follows:



2

3 *end example]*

4 **12.1.7 DeviceFontName Attribute**

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

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

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

17 For producers, a <Glyphs> element with a specified device font name MUST have exactly one
18 Indices glyph per code unit in the UnicodeString attribute. Its Indices attribute MUST NOT include
19 any cluster specifications. If the Indices attribute includes a cluster mapping, the consumer
20 MUST NOT use the device font and MUST render the embedded version of the font [M5.19].

1 This means that a device font cannot be used for characters outside the basic multilingual plane
2 (BMP).

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

6 **12.1.8 xml:lang Attribute**

7 XPS Document consumers might need to override the default language for a specific run of
8 glyphs, particularly in multilingual documents. The language defaults to the value specified for
9 the xml:lang attribute of the <FixedPage> element but MAY be overridden by an xml:lang
10 attribute on a <Glyphs> element [O5.13M2.72]. For larger blocks of text, the producer MAY
11 specify the xml:lang attribute on the <Canvas> element [M5.27M2.72].

12 The language specified does not affect rendering of <Glyphs> elements, but it can be used by
13 consumers for searching or selecting text. For more information, see §9.3.5.

14 **12.1.9 CaretStops Attribute**

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

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

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

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

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

```
32 UnicodeString = "This is an example string of text."  
33 CaretStops = "fffd"
```

34 *end example*]

35 **12.1.10 Optimizing Glyph Markup**

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

39 **12.1.10.1 Optimizing Glyph Indices Markup**

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

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

12.1.10.2 Optimizing Glyph Position Markup

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

- For glyphs that have not been algorithmically emboldened, the desired advance width is the value listed in the horizontal metrics font table (if the IsSideways attribute value is false) or as calculated in §12.1.6.1 (if the IsSideways attribute value is true).
- For algorithmically emboldened glyphs, the desired advance width is exactly 2% larger than the values in the horizontal metrics font table (if the IsSideways attribute value is false) or as calculated in §12.1.6.1 (if the IsSideways attribute value is true).

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

12.1.11 Glyph Markup Examples

Example 12–13. Basic italic font

```
<Canvas>
  <Glyphs
    FontUri="../Resources/Fonts/Timesi.ttf"
    FontRenderingEmSize="20"
    OriginX="35"
    OriginY="35"
    UnicodeString="Basic italic font..."
    Fill="#009900" />
  </Canvas>
```

This text is rendered as follows:



Basic italic font...

end example]

Example 12–14. Italic font using StyleSimulations attribute

```
<Canvas>
  <Glyphs
    FontUri="../Resources/Fonts/Times.ttf"
    FontRenderingEmSize="20"
    StyleSimulations="ItalicSimulation"
    OriginX="35"
    OriginY="35"
    UnicodeString="Simulated italic font..."
    Fill="#009900" />
  </Canvas>
```

1 This text is rendered as follows:



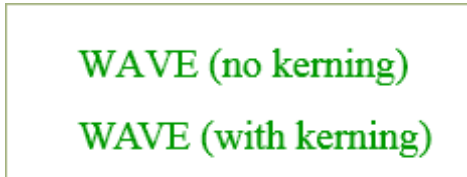
Simulated italic font...

2
3 *end example]*

4 *Example 12-15. Kerning*

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

29 This text is rendered as follows:



WAVE (no kerning)
WAVE (with kerning)

30
31 *end example]*

32 *Example 12-16. Ligatures*

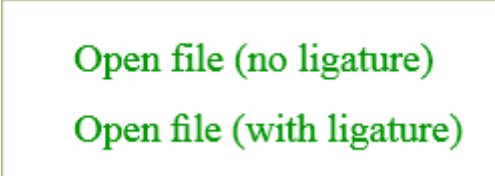
```
33     <Canvas>
34
35         <!-- "Open file" without "fi" ligature -->
36
37         <Glyphs
38            OriginX="35"
39            OriginY="35"
40            UnicodeString="Open file (no ligature)"
41            FontUri=" ../Resources/Fonts/Times.ttf"
```

```

1      FontRenderingEmSize="20"
2      Fill="#009900" />
3
4      <!-- "Open file" with "fi" ligature -->
5
6      <Glyphs
7          OriginX="35"
8          OriginY="70"
9          UnicodeString="Open file (with ligature)"
10         Indices=";;;(2:1)191"
11         FontUri="../Resources/Fonts/Times.ttf"
12         FontRenderingEmSize="20"
13         Fill="#009900" />
14
15     </Canvas>

```

16 This text is rendered as follows:



17

18 *end example]*

19 *Example 12-17. Cluster maps*

```

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

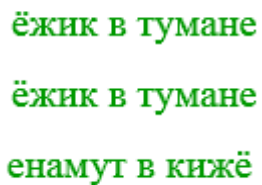
```

```

1
2     <Glyphs
3       OriginX="155"
4       OriginY="105"
5       BidiLevel="1"
6       xml:lang="ru-RU"
7       UnicodeString="ёжик в тумане"
8       Indices="(1:2)72;142,0,-40"
9       FontUri=" ../Resources/Fonts/Times.ttf"
10      FontRenderingEmSize="20"
11      Fill="#009900" />
12
13 </Canvas>

```

14 This text is rendered as follows:

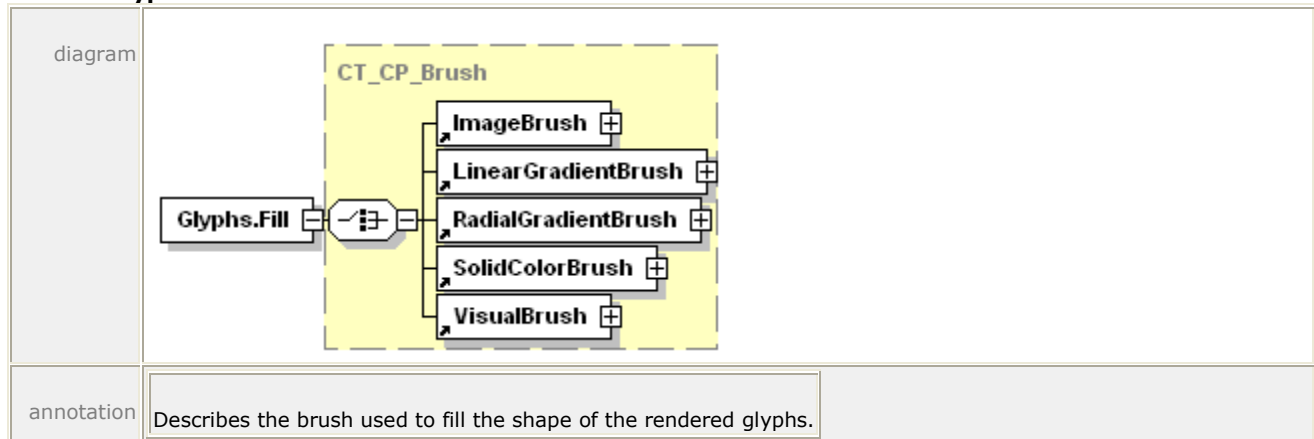


ёжик в тумане
ёжик в тумане
енамут в кижё

15
16 *end example]*

17 12.2 <Glyphs.Fill> Element

18 element **Glyphs.Fill**



19 The Fill property specifies the brush that fills a glyph. Any brush can be used.

20

1 13. Brushes

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

6 All brushes are defined relative to a coordinate space. Most brushes (including image brushes,
 7 visual brushes, linear gradient brushes, ~~Brush3D (Annex J)~~, and radial gradient brushes) can
 8 specify a coordinate-space transform, in which the Transform property is concatenated with the
 9 current effective coordinate space to yield an effective coordinate space local to the brush. For
 10 image brushes and visual brushes, the viewport is transformed using the local effective render
 11 transform. For linear gradient brushes, the start point and end point are transformed. For radial
 12 gradient brushes, the ellipse defined by the center, x radius, y radius, and gradient origin is
 13 transformed.

14 [Annex A defines an additional \(optional\) brush for the representation of 3D content.](#)

15 *Table 13-1. Brush types*

Name	Description
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
3D Content Brush	Fills a region with a 3D graphics model (Annex J)

1 **13.1 <SolidColorBrush> Element**

2 element **SolidColorBrush**

diagram	<p>The diagram shows a class named SolidColorBrush with three attributes: Opacity, x:Key, and Color. A dashed box labeled attributes encloses the Opacity and x:Key attributes. A yellow box labeled CT_SolidColorf highlights the Opacity and x:Key attributes. A line connects the SolidColorBrush class to the CT_SolidColorf box.</p>					
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 [M6.1].
	Color	<u>ST_Color</u>	required			Specifies the color for filled elements.
annotation	Fills defined geometric regions with a solid color.					

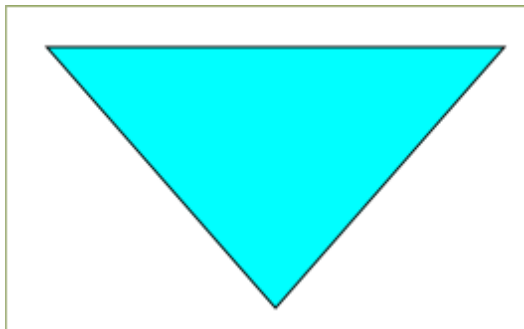
3 The <SolidColorBrush> element is used to fill defined geometric regions with a solid color. If
 4 there is an alpha component of the color, it is combined in a multiplicative way with the
 5 corresponding Opacity attribute.

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:

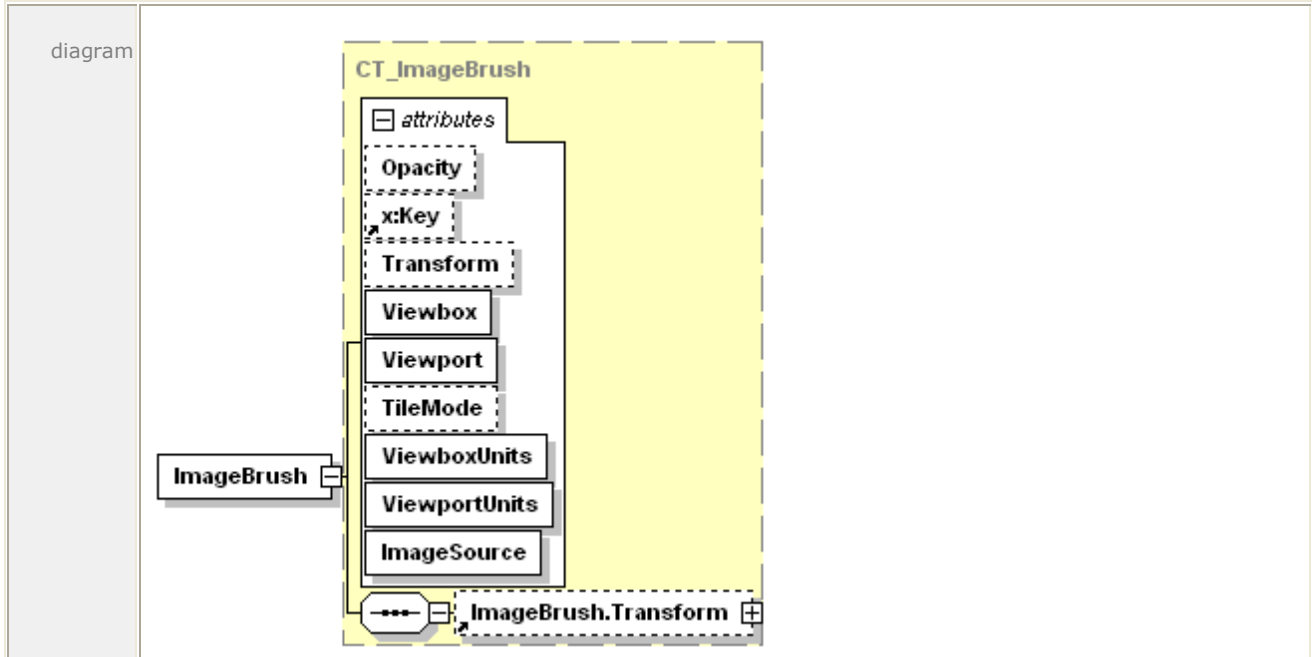


16

17 *end example]*

18 **13.2 <ImageBrush> Element**

19 element **ImageBrush**



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 [M6.2].
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 the local effective render transform.
Viewbox	ST_ViewBox	required			Specifies the position and dimensions of the brush's source content. Specifies four comma-separated real numbers (x, y, W width, H 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	ST_ViewBox	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, W width, H height), where width and height are non-negative. The alignment of the brush pattern is controlled by adjusting the x and y values.
TileMode	ST_TileMode		None		Specifies how contents will be tiled in the filled region. Valid values are None, Tile, FlipX, FlipY, and FlipXY.
ViewboxUnits	ST_ViewUnits	required		Absolute	Specifies the relationship of the viewbox coordinates to the containing coordinate space.
ViewportUnits	ST_ViewUnits	required		Absolute	Specifies the relationship of the viewport coordinates to the containing coordinate space.
ImageSource	ST_UriCtxBmp	required			Specifies the URI of an image resource or a combination of the URI of an image resource a color profile resource. See §15.3.8 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 defined in a
 2 coordinate space specified by the resolution of the image. The image MUST refer to a JPEG,
 3 PNG, TIFF, or Windows Media Photo image part within the XPS Document package [M6.3]. For
 4 more information, see §9.1.5. A URI part name for the image is specified using the ImageSource
 5 attribute.

6 Image brushes share a number of tile-related properties with visual brushes. For details,
 7 see §13.4.

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

9 The following markup describes an image on a canvas.

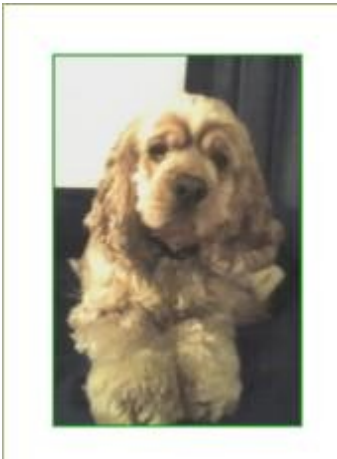
```

10 <Canvas>
11   <Path Stroke="#008000">
12     <Path.Fill>
13       <ImageBrush
14         ImageSource="dog.jpg"
15         TileMode="None"
16         Viewbox="0,0,270,423"

```

```
1         ViewboxUnits="Absolute"  
2         Viewport="25,25,125,185"  
3         ViewportUnits="Absolute" />  
4     </Path.Fill>  
5     <Path.Data>  
6         <PathGeometry>  
7             <PathFigure StartPoint="25,25" IsClosed="true">  
8                 <PolyLineSegment Points="150,25 150,210 25,210" />  
9             </PathFigure>  
10        </PathGeometry>  
11    </Path.Data>  
12 </Path>  
13 </Canvas>
```

14 This markup produces the following results:

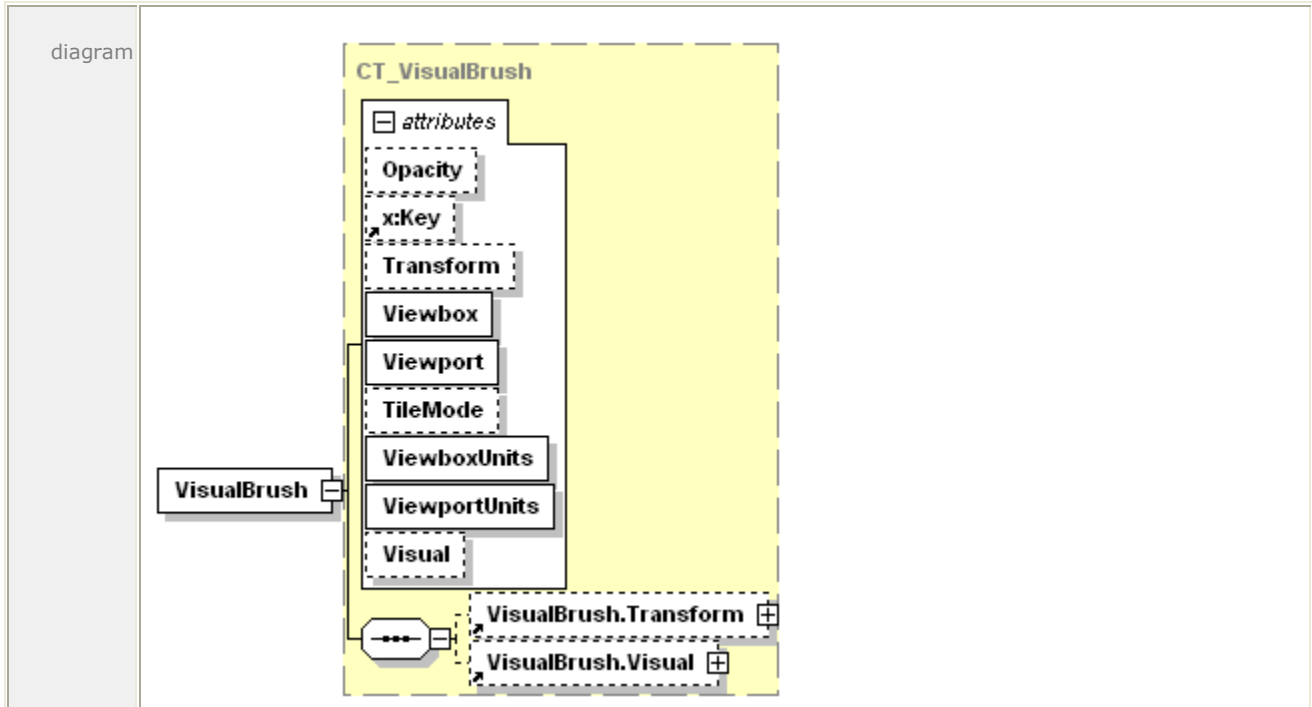


15

16 *end example]*

1 **13.3 <VisualBrush> Element**

2 element **VisualBrush**



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	ST_ViewBox	required			Specifies the position and dimensions of the brush's source content. Specifies four comma-separated real numbers (x, y, W width,

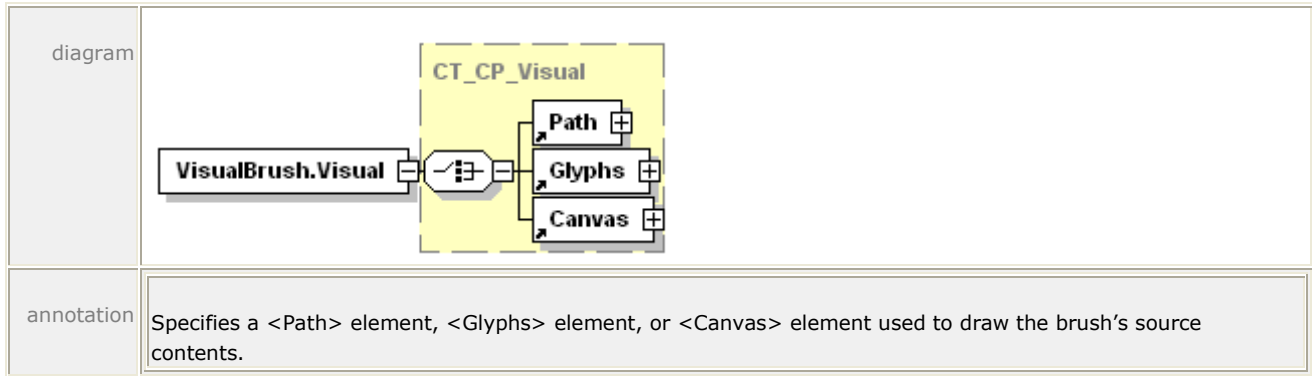
					<p>Hheight), where width and height are non-negative. The viewbox defines the default coordinate system for the element specified in the <VisualBrush.Visual> property element. The corners of the viewbox are mapped to the corners of the viewport, thereby providing the default clipping and transform for the brush's source content.</p>
	Viewport	ST_ViewBox	required		<p>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, Wwidth, Hheight), where width and height are non-negative. The alignment of the brush pattern is controlled by adjusting the x and y values.</p>
	TileMode	ST_TileMode		None	<p>Specifies how contents will be tiled in the filled region. Valid values are None, Tile, FlipX, FlipY, and FlipXY.</p>
	ViewboxUnits	ST_ViewUnits	required	Absolute	<p>Specifies the relationship of the viewbox coordinates to the containing coordinate space.</p>
	ViewportUnits	ST_ViewUnits	required	Absolute	<p>Specifies the relationship of the viewport coordinates to the containing coordinate space.</p>
	Visual	ST_RscRef			<p>Specifies resource reference to a <Path>, <Glyphs>, or <Canvas> element defined in a resource dictionary and used to draw the brush's source content.</p>
annotation	<p>Fills a region with a drawing. The drawing can 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.</p>				

1 The <VisualBrush> element is used to fill a region with a drawing. The drawing can be specified
 2 as either a <VisualBrush.Visual> property element or as a resource reference. Drawing content
 3 can include exactly one <Canvas>, <Path>, or <Glyphs> element and that element's child and
 4 descendant elements.

5 Visual brushes share a number of tile-related properties with image brushes. For details,
 6 see §13.4.

7 **13.3.1 <VisualBrush.Visual> Element**

8 element **VisualBrush.Visual**



1 The `<VisualBrush.Visual>` property element contains markup that defines the contents of a
 2 single visual brush tile. The tile can be used to fill the geometric region to which the visual
 3 brush is applied. The `<VisualBrush.Visual>` property element contains a single child element.
 4 For simple tiles, this can be a single `<Path>` or `<Glyphs>` element. More complex visuals
 5 containing multiple `<Path>` and `<Glyphs>` elements can be grouped within a `<Canvas>`
 6 child element.

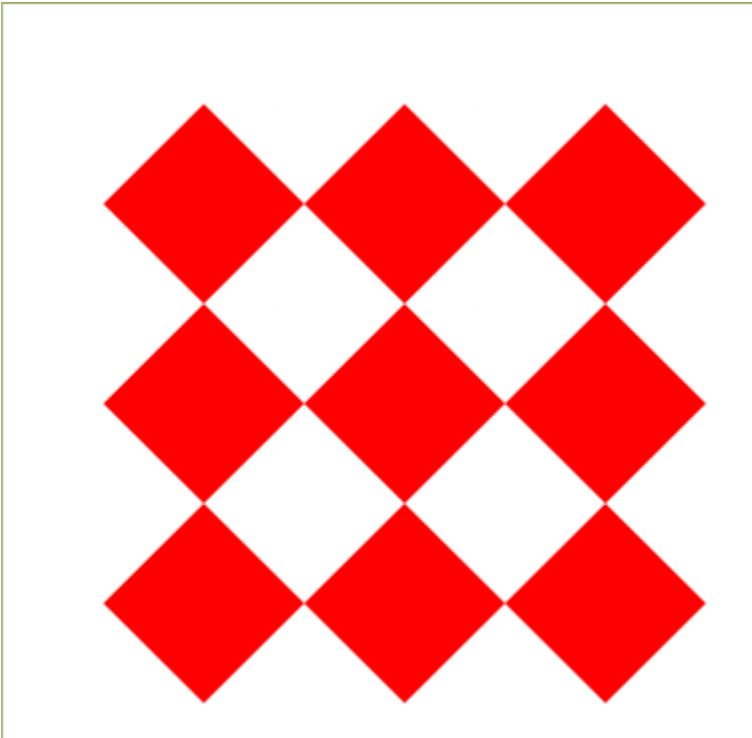
7 *Example 13–3. `<VisualBrush.Visual>` usage*

```

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

```

1 This markup produces the following result:



2

3 *end example]*

13.4 Common Attributes for Tiling Brushes

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

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" [M6.7M2-72].
ViewportUnits	Specifies the unit type for the Viewport attribute. MUST have the value "Absolute" [M6.8M2-72].
TileMode	Specifies how tiling is performed in the filled geometry. The value is optional, and defaults to "None" if no value is specified.

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

13.4.1 Viewbox, Viewport, ViewboxUnits, and ViewportUnits Attributes

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

For images, the dimensions specified by the viewbox are expressed in units of 1/96". The pixel coordinates in the source image are calculated as follows, [where HorizontalImageResolution and VerticalImageResolution are specified in dpi](#):

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

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

The viewbox can specify a region larger than the image itself, including negative 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 resolution of
11 50 dpi and measures 100 pixels horizontally and 50 pixels vertically, the physical dimensions of
12 the image are expressed (in units of 1/96") as $96 * 100 / 50 = 192$ horizontal and $96 * 50 / 50$
13 = 96 vertical.

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

19 **13.4.1.1 Viewbox and Viewport Examples**

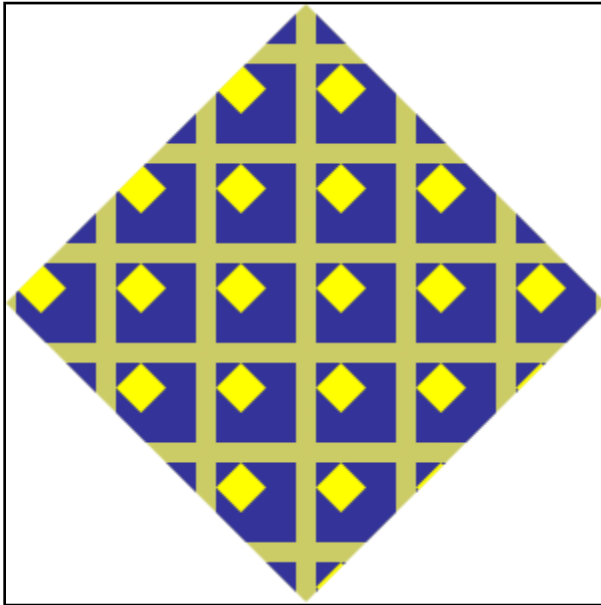
20 The following examples demonstrate how adjusting the viewbox and viewport can affect output.

21 *Example 13–5. Tiling brush base image and rendering*

22 The following markup describes a base image.

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

1 This markup is rendered as follows:



2

3 *end example]*

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

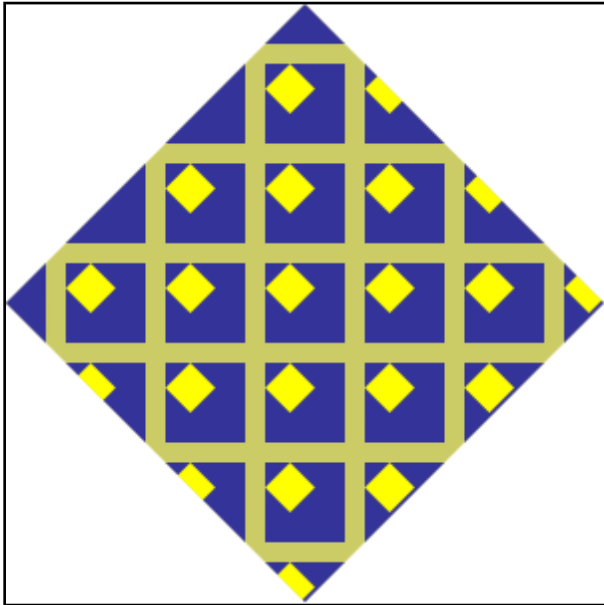
5 By adjusting the viewport, the position of the tiles within the image can be changed:

```

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*

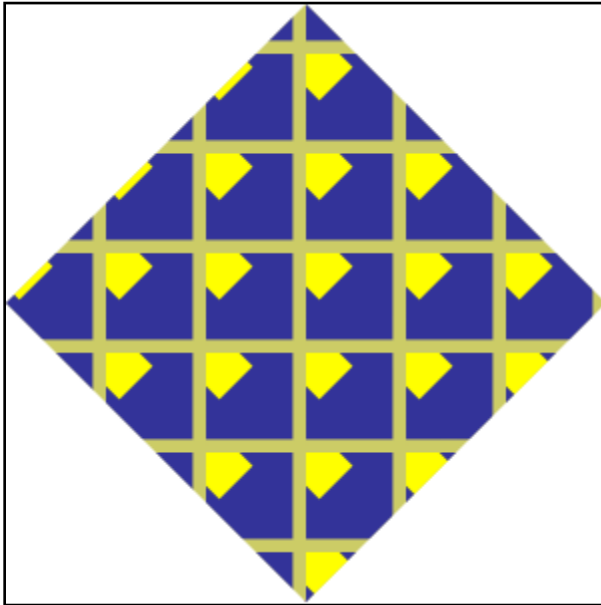
5 The following markup uses a smaller window on the viewbox to zoom in on each tile:

```

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*

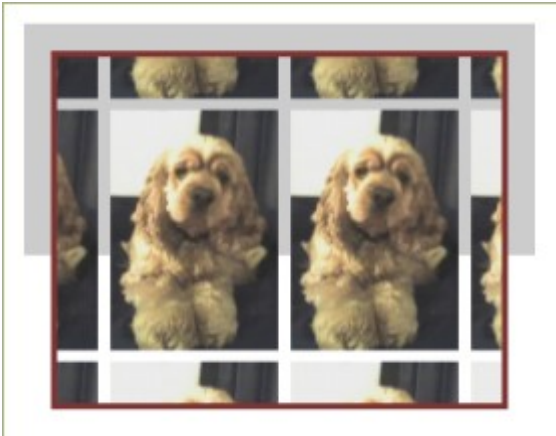
5 An image brush can specify a tile with the Viewbox attribute that exceeds the size of the image
6 it uses, including negative values, as shown below.

```

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>

```

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



- 3
 4 *end example]*

5 **13.4.2 TileMode Attribute**

- 6 Valid values for the TileMode attribute are None, Tile, FlipX, FlipY, and FlipXY.

7 **13.4.2.1 None**

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

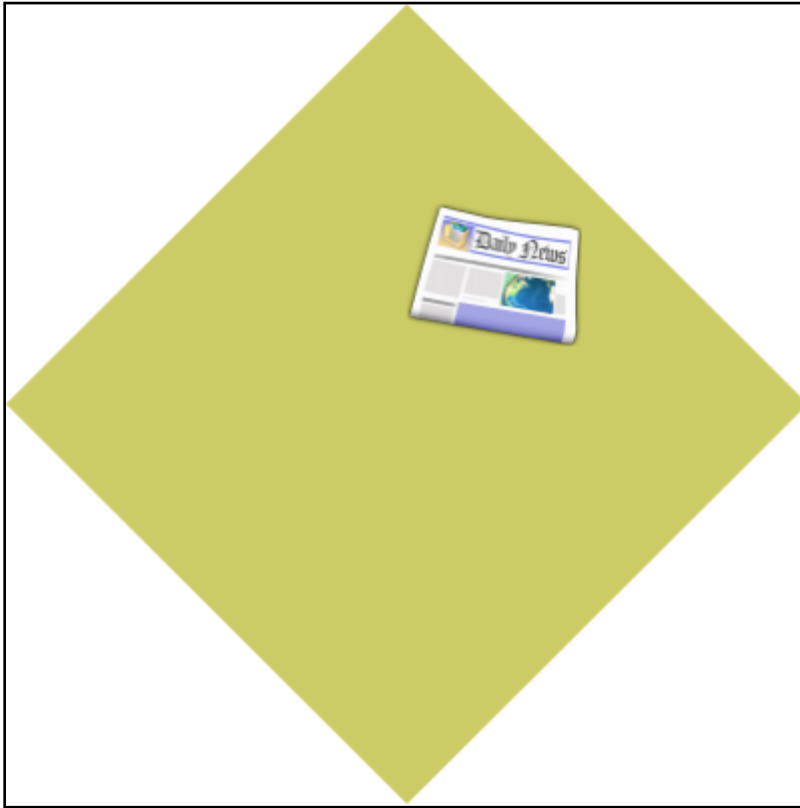
- 9 *Example 13-9. Image brush with TileMode value of None*

```

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

```

1 This markup is rendered as follows:



2

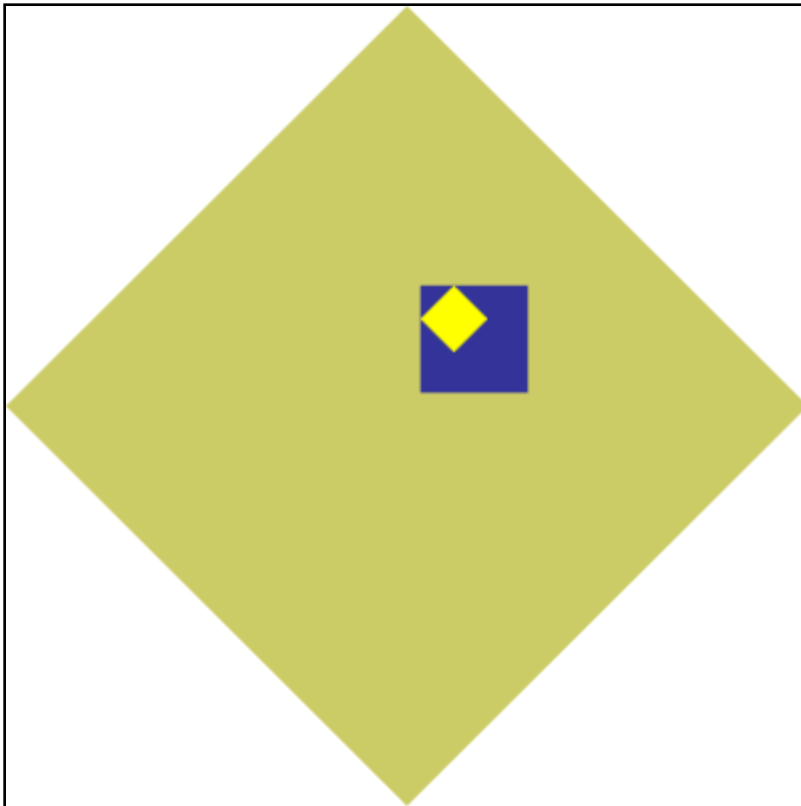
3 *end example]*

```

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 base tile
3 such that the right edge of each tile abuts the left edge of the next, and the bottom edge of
4 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]*

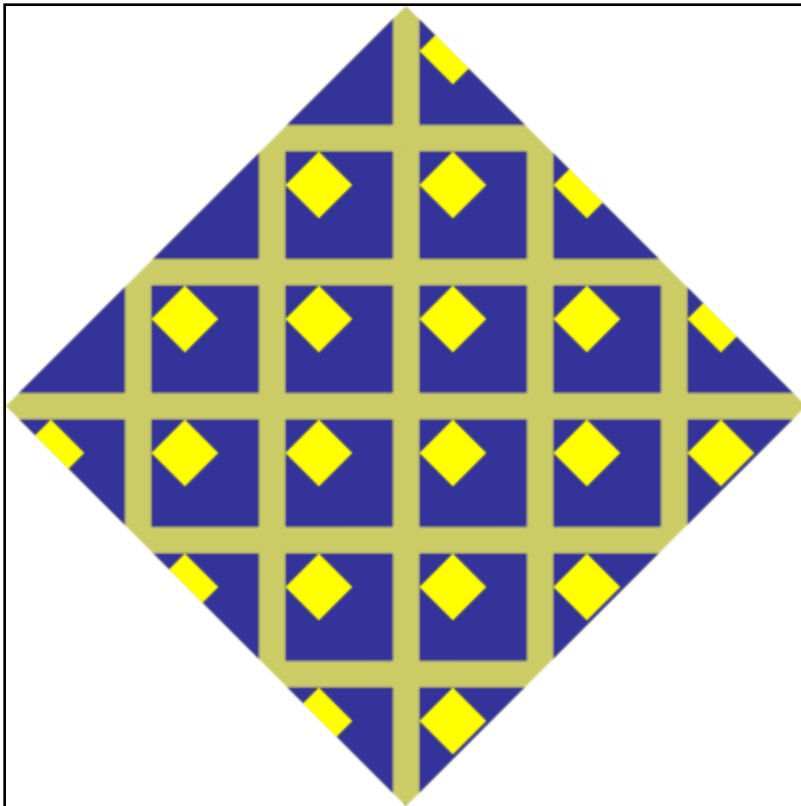
1 *Example 13-12. Visual brush with a TileMode value of Tile*

```

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 are flipped
3 horizontally. The base tile is positioned as specified by the viewport. Tiles in the columns to the
4 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]*

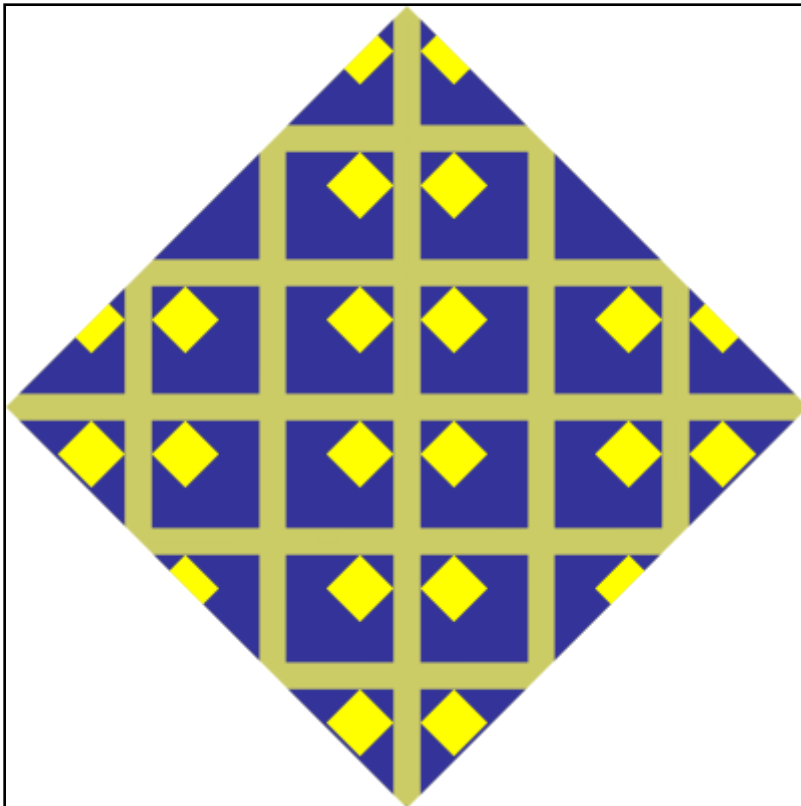
1 *Example 13-14. Visual brush with a TileMode value of FlipX*

```

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 flipped
3 vertically. The base tile is positioned as specified by the viewport. Rows above and below are
4 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]*

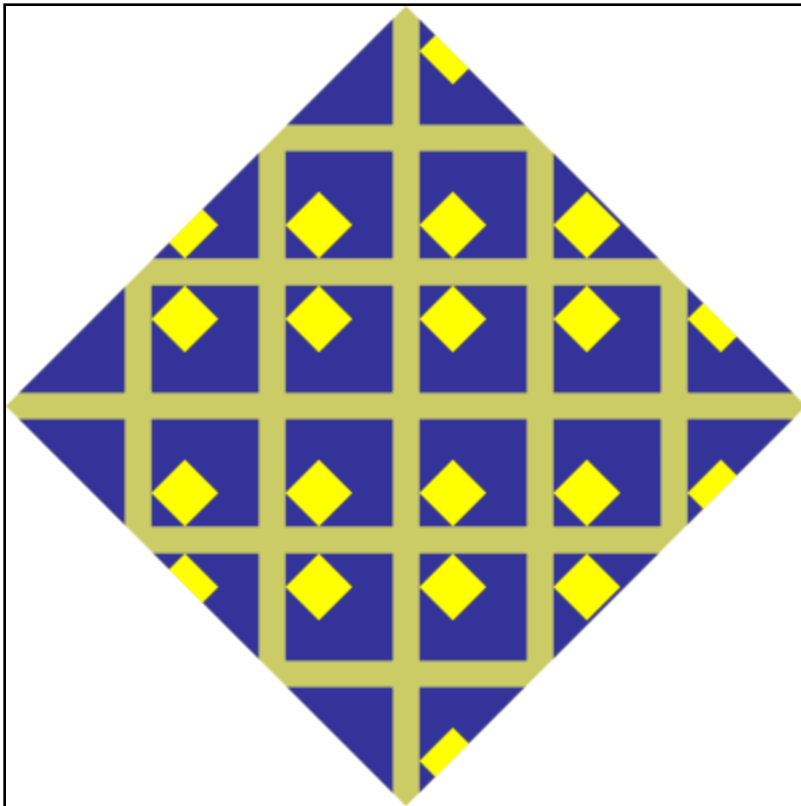
1 *Example 13-16. Visual Brush with a TileMode value of FlipY*

```

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]*

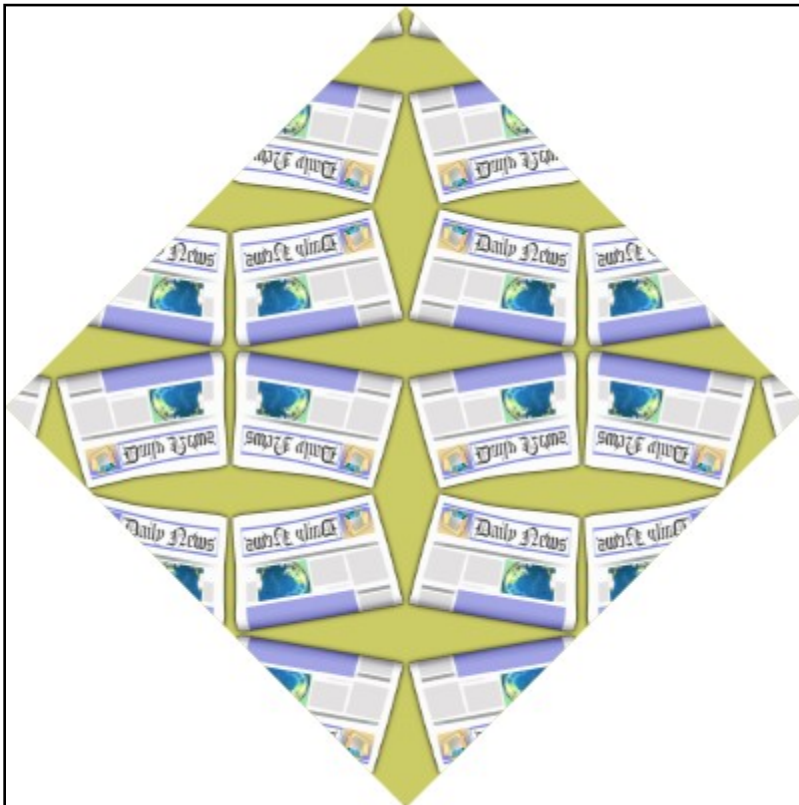
1 13.4.2.5 FlipXY

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

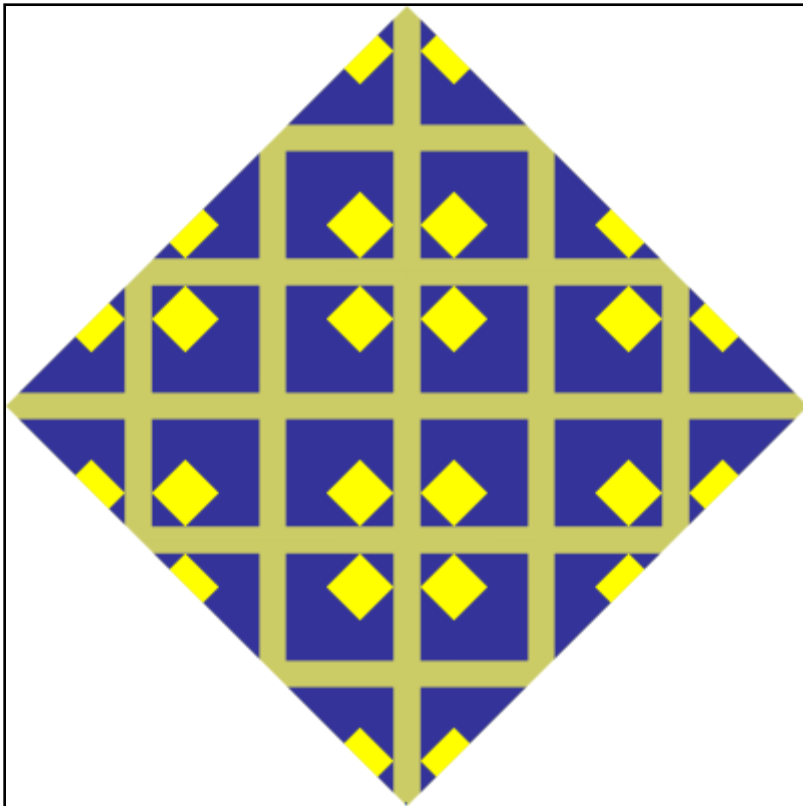
1 *Example 13-18. Visual brush with a TileMode value of FlipXY*

```

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:

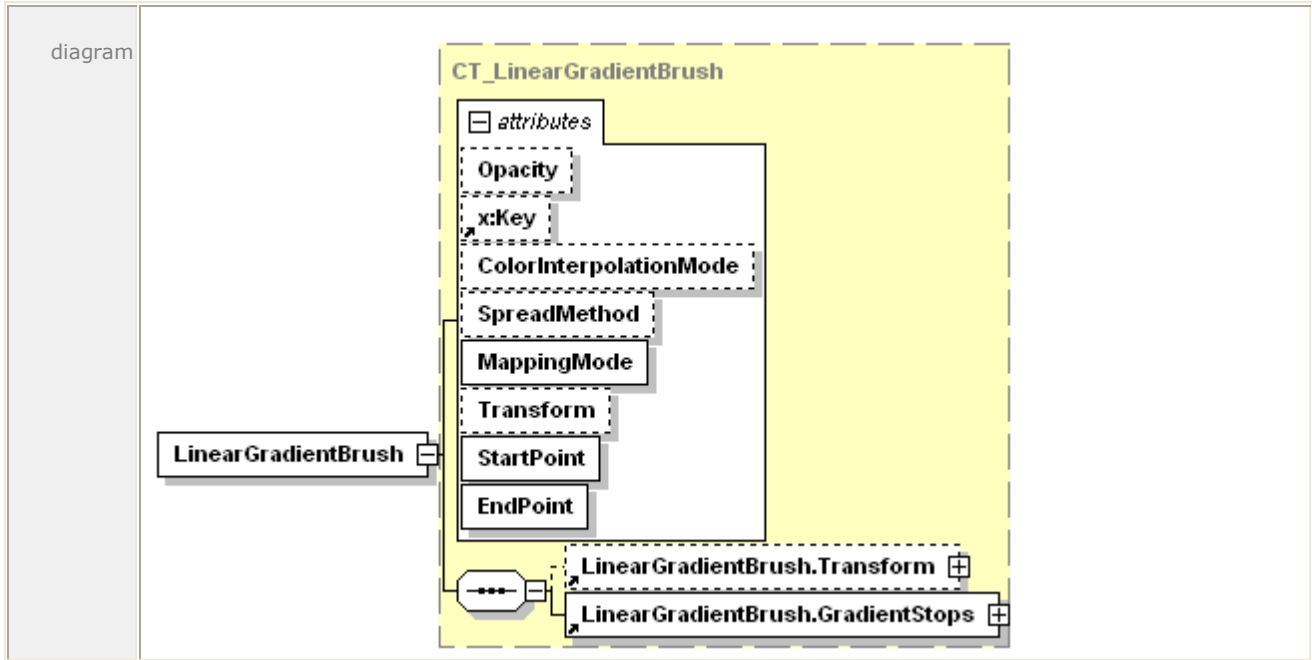


24

25 *end example]*

1 **13.5 <LinearGradientBrush> Element**

2 element **LinearGradientBrush**



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	ST_SpreadMethod		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	ST_MappingMode	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	ST_RscRefMatrix				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	ST_Point	required			Specifies the starting point of the linear gradient.
	EndPoint	ST_Point	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 The <LinearGradientBrush> element is used to specify a linear gradient brush along a vector.
- 2 For details about computing a linear gradient, see §18.3.

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

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

```

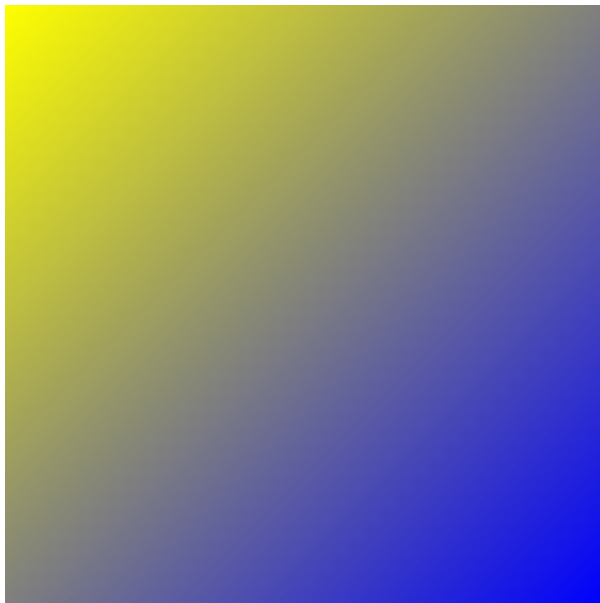
6     <Path>
7         <Path.Fill>
    
```

```

1      <LinearGradientBrush
2          MappingMode="Absolute"
3          StartPoint="0,0"
4          EndPoint="300,300">
5          <LinearGradientBrush.GradientStops>
6              <GradientStop Color="#FFFF00" Offset="0" />
7              <GradientStop Color="#0000FF" Offset="1" />
8          </LinearGradientBrush.GradientStops>
9      </LinearGradientBrush>
10     </Path.Fill>
11     <Path.Data>
12         <PathGeometry>
13             <PathFigure StartPoint="0,0">
14                 <PolyLineSegment Points="300,0 300,300 0,300" />
15             </PathFigure>
16         </PathGeometry>
17     </Path.Data>
18 </Path>

```

19 This markup is rendered as follows:



20
21 *end example]*

22 **13.5.1 SpreadMethod Attribute**

23 The SpreadMethod attribute describes the fill for areas beyond the start point and end point of
24 the linear gradient brush. Valid values are Pad, Reflect, and Repeat. [For details see §18.3.2.](#)

25 *Example 13–20. Linear gradient brush with a SpreadMethod value of Pad*

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

```

28     <Path Data="M 150,0 L 300,150 L 150,300 L 0,150 Z">
29         <Path.Fill>
30             <LinearGradientBrush

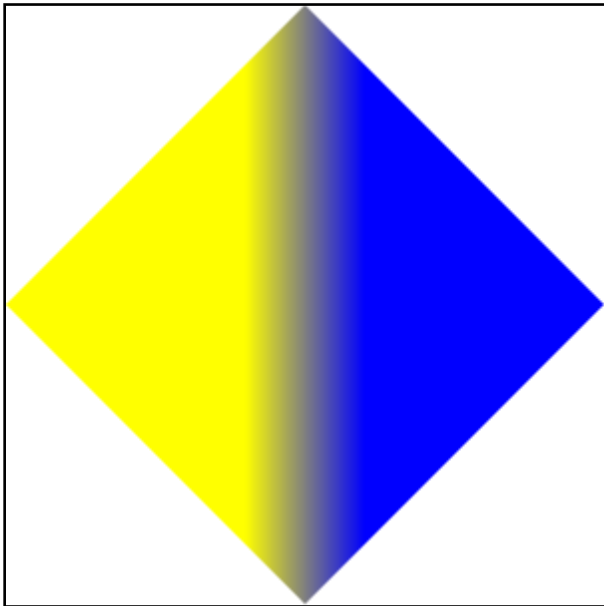
```

```

1      MappingMode="Absolute"
2      StartPoint="120,0"
3      EndPoint="180,0"
4      SpreadMethod="Pad">
5      <LinearGradientBrush.GradientStops>
6          <GradientStop Color="#FFFF00" Offset="0.0" />
7          <GradientStop Color="#0000FF" Offset="1.0" />
8      </LinearGradientBrush.GradientStops>
9      </LinearGradientBrush>
10     </Path.Fill>
11 </Path>

```

12 This markup is rendered as follows:



13

14 *end example]*

15 *Example 13–21. Linear gradient brush with a SpreadMethod value of Reflect*

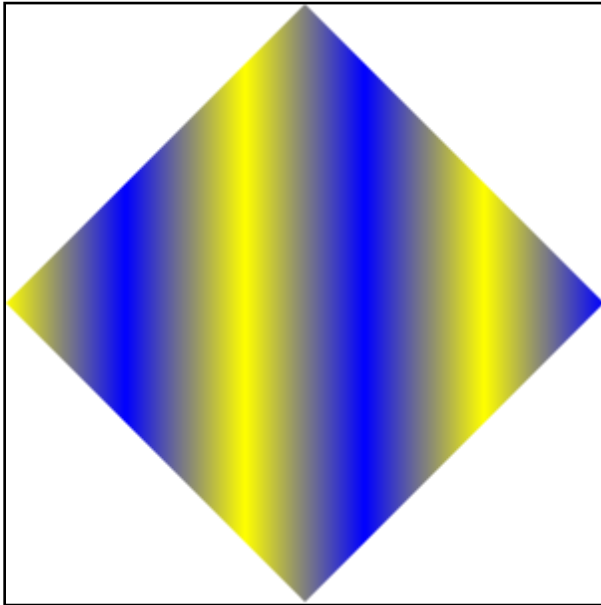
16 In this method, the gradient stops are replayed in reverse order repeatedly to cover the fill
17 area.

```

18     <Path Data="M 150,0 L 300,150 L 150,300 L 0,150 Z">
19         <Path.Fill>
20             <LinearGradientBrush
21                 MappingMode="Absolute"
22                 StartPoint="120,0"
23                 EndPoint="180,0"
24                 SpreadMethod="Reflect">
25                 <LinearGradientBrush.GradientStops>
26                     <GradientStop Color="#FFFF00" Offset="0.0" />
27                     <GradientStop Color="#0000FF" Offset="1.0" />
28                 </LinearGradientBrush.GradientStops>
29             </LinearGradientBrush>
30         </Path.Fill>
31 </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*

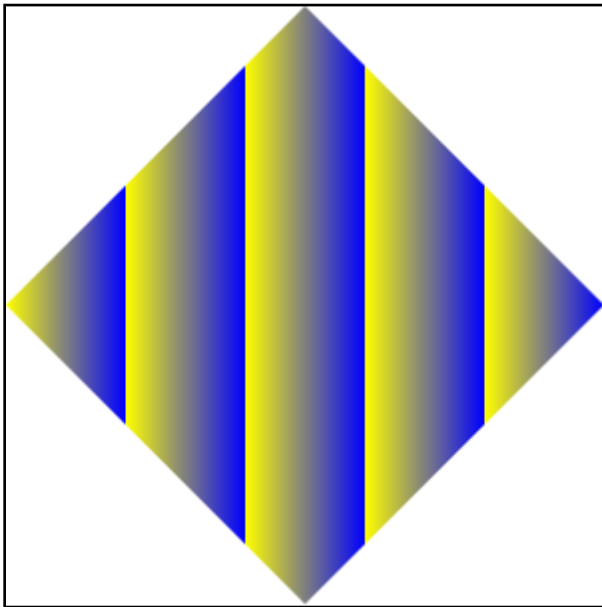
5 In this method, the gradient stops are repeated in order until the fill area is covered.

```

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:

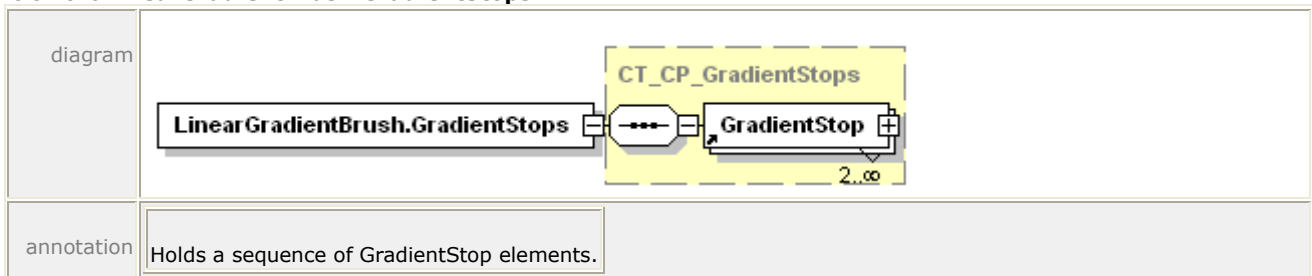


2

3 *end example]*

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

5 element **LinearGradientBrush.GradientStops**

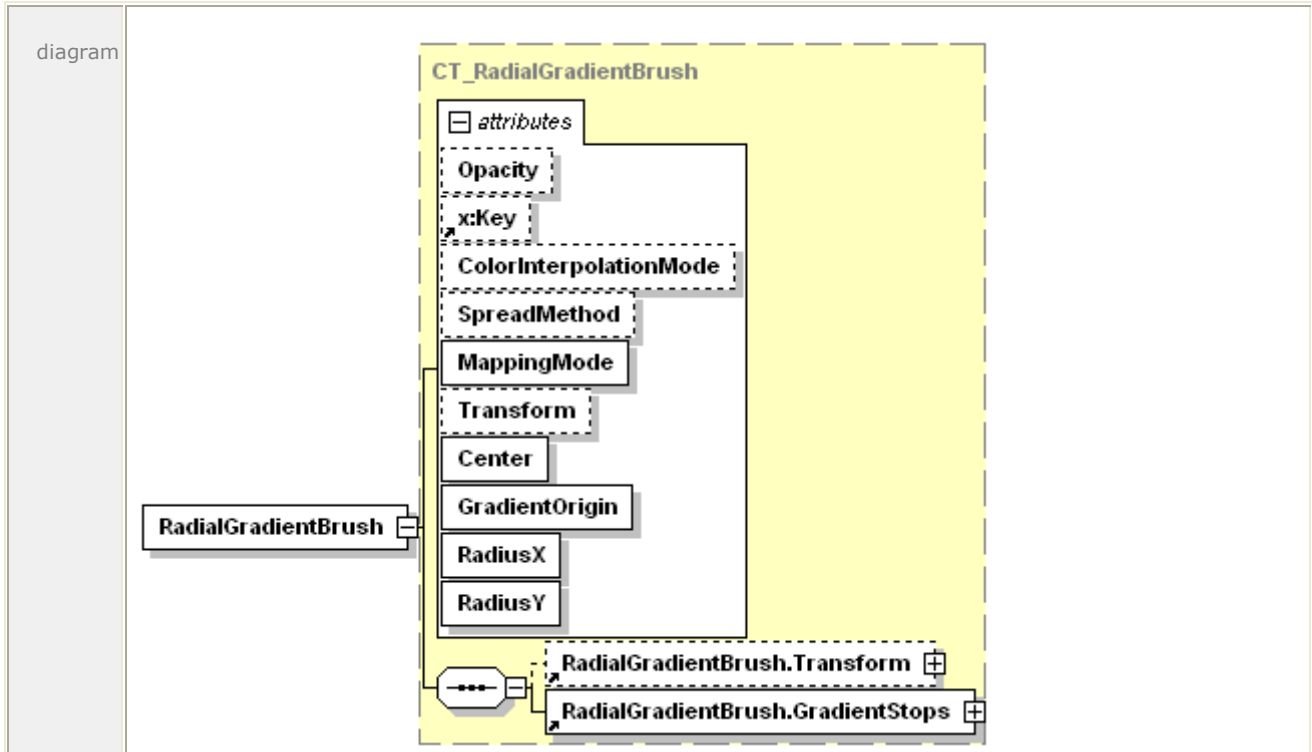


6

7 The <LinearGradientBrush.GradientStops> property element specifies a collection of gradient
 8 stops that comprise the linear gradient. For more information, see §13.7.

1 13.6 <RadialGradientBrush> Element

2 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	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. Specifies	

					<p>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.</p>
SpreadMethod	ST_SpreadMethod		Pad		<p>Describes how the brush should fill the content area outside of the primary, initial gradient area. Valid values are Pad, Reflect and Repeat.</p>
MappingMode	ST_MappingMode	required		Absolute	<p>Specifies that center, x radius, and y radius are defined in the effective coordinate space (includes the Transform attribute of the brush).</p>
Transform	ST_RscRefMatrix				<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 values is transformed using the local effective render transform.</p>
Center	ST_Point	required			<p>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.</p>
GradientOrigin	ST_Point	required			<p>Specifies the origin point of the radial gradient.</p>
RadiusX	ST_GEZero	required			<p>Specifies the radius in the x dimension of the ellipse which defines the radial gradient.</p>
RadiusY	ST_GEZero	required			<p>Specifies the radius in the y dimension of the ellipse which</p>

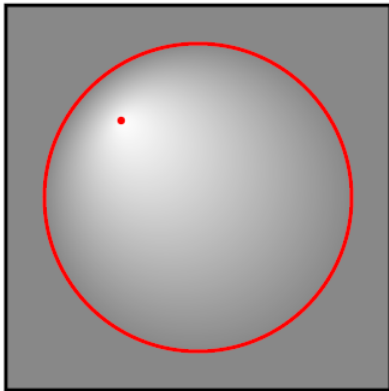
						defines the radial gradient.
annotation	Fills a region with a radial gradient.					

1 Radial gradient brushes are similar to linear gradient brushes. However, whereas a linear
 2 gradient brush has a start point and end point to define the gradient vector, a radial gradient
 3 brush has an ellipse (defined by the center, x radius, and y radius) and a gradient origin. The
 4 ellipse defines the end point of the gradient. In other words, a gradient stop with an offset
 5 at 1.0 defines the color at the circumference of the ellipse. ~~The gradient origin defines the~~
 6 ~~center of the gradient.~~ A gradient stop with an offset at 0.0 defines the color at the gradient
 7 origin.

8 For details about computing a radial gradient, see §18.3.3.

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

10 The following figure is a radial gradient that transitions from white to gray. The outside ellipse
 11 represents the gradient ellipse while the dot denotes the gradient origin. This gradient has a
 12 SpreadMethod value of Pad. ~~end example]~~



13

14 *end example]*

15 *Example 13-24. RadialGradientBrush usage*

16 The following markup describes a page with a rectangular path that is filled with a radial
 17 gradient:

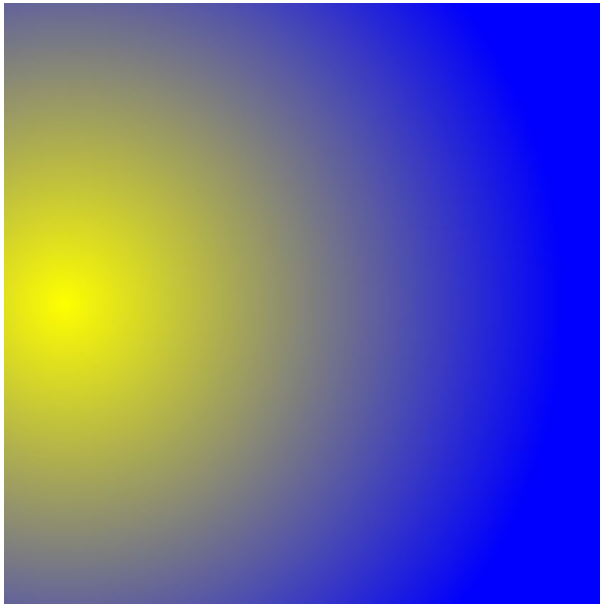
```
18     <Path>
19         <Path.Fill>
20             <RadialGradientBrush
21                 MappingMode="Absolute"
22                 Center="30,150"
23                 GradientOrigin="30,150"
24                 RadiusX="250"
25                 RadiusY="250">
26                 <RadialGradientBrush.GradientStops>
27                     <GradientStop Color="#FFFF00" Offset="0" />
28                     <GradientStop Color="#0000FF" Offset="1" />
```

```

1         </RadialGradientBrush.GradientStops>
2     </RadialGradientBrush>
3 </Path.Fill>
4 <Path.Data>
5     <PathGeometry>
6         <PathFigure StartPoint="0,0" IsClosed="true">
7             <PolyLineSegment Points="300,0 300,300 0,300" />
8         </PathFigure>
9     </PathGeometry>
10 </Path.Data>
11 </Path>

```

12 This markup is rendered as follows:



13

14 *end example]*

15 13.6.1 SpreadMethod Attribute

16 The SpreadMethod attribute describes the fill of areas beyond the ellipse described by the center,
 17 x radius, and y radius of the radial gradient brush. Valid values are Pad, Reflect, and Repeat.

18 [For details see §18.3.3.](#)

19 *Example 13–25. Radial gradient brush with a SpreadMethod value of Pad*

20 In the following markup, the last color is used to cover the fill area outside the ellipse.

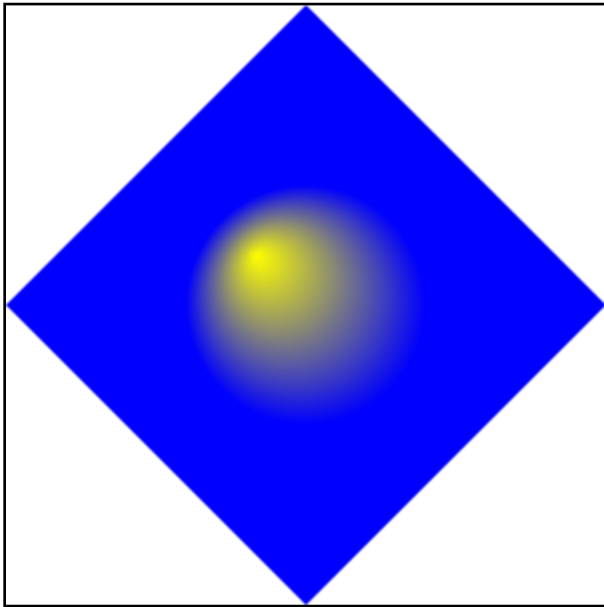
```

21 <Path Data="M 150,0 L 300,150 L 150,300 L 0,150 Z">
22 <Path.Fill>
23     <RadialGradientBrush
24         MappingMode="Absolute"
25         Center="150,150"
26         GradientOrigin="125,125"
27         RadiusX="60"
28         RadiusY="60"
29         SpreadMethod="Pad">
30 </RadialGradientBrush.GradientStops>

```

```
1         <GradientStop Color="#FFFF00" Offset="0.0" />
2         <GradientStop Color="#0000FF" Offset="1.0" />
3     </RadialGradientBrush.GradientStops>
4 </RadialGradientBrush>
5 </Path.Fill>
6 </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*

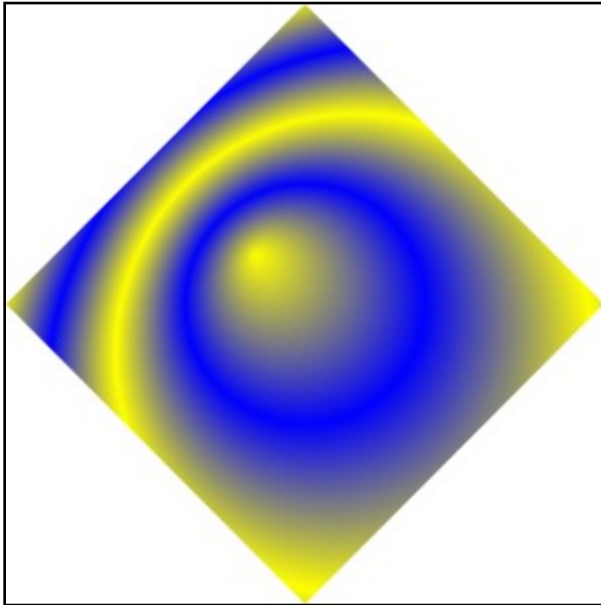
5 In the following markup, the gradient stops are replayed in reverse order repeatedly to cover
6 the fill area.

```

7     <Path Data="M 150,0 L 300,150 L 150,300 L 0,150 Z">
8         <Path.Fill>
9             <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*

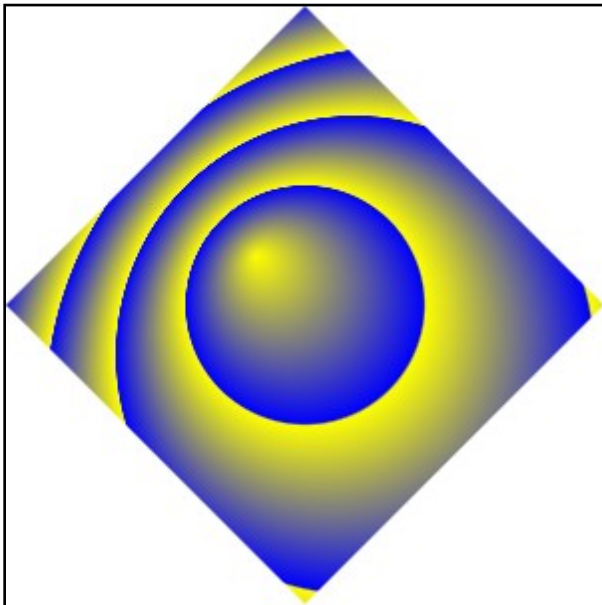
5 In the following markup, the gradient stops are repeated in order until the fill area is covered.

```

6     <Path Data="M 150,0 L 300,150 L 150,300 L 0,150 Z">
7         <Path.Fill>
8             <RadialGradientBrush
9                 MappingMode="Absolute"
10                Center="150,150"
11                GradientOrigin="125,125"
12                RadiusX="60"
13                RadiusY="60"
14                SpreadMethod="Repeat">
15                <RadialGradientBrush.GradientStops>
16                    <GradientStop Color="#FFFF00" Offset="0.0" />
17                    <GradientStop Color="#0000FF" Offset="1.0" />
18                </RadialGradientBrush.GradientStops>
19            </RadialGradientBrush>
20        </Path.Fill>
21    </Path>

```

1 This markup is rendered as follows:

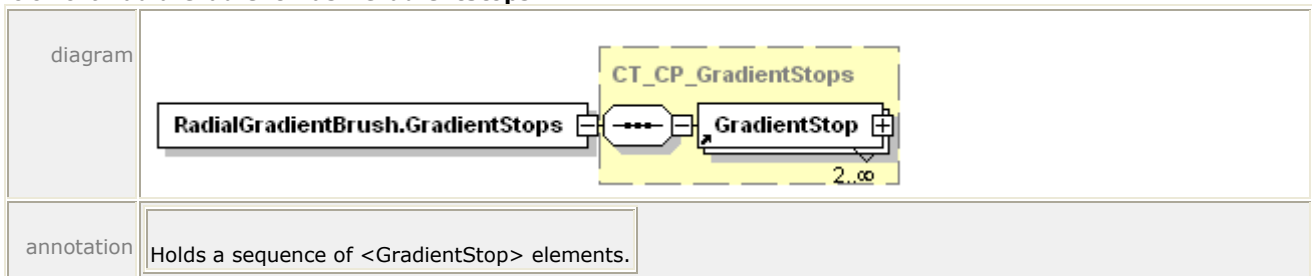


2

3 *end example]*

4 **13.6.2 <RadialGradientBrush.GradientStops> Element**

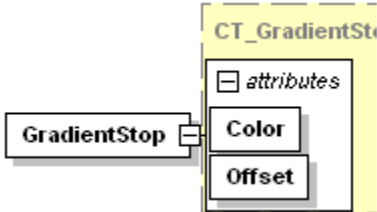
5 element **RadialGradientBrush.GradientStops**



6 The <RadialGradientBrush.GradientStops> property element specifies a collection of gradient
 7 stops that comprise the radial gradient. For more information, see §13.7.

1 13.7 <GradientStop> Element

2 element **GradientStop**

diagram						
attributes	Name	Type	Use	Default	Fixed	Annotation
	Color	<u>ST_Color</u>	required			Specifies the gradient stop color.
	Offset	<u>ST_Double</u>	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 progression for
 5 rendering a gradient.

6 For linear gradient brushes, the offset value of 0.0 is mapped to the start point of the gradient,
 7 and the offset value of 1.0 is mapped to the end point. Intermediate offset values are
 8 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 location.
 10 The offset value of 1.0 is mapped to the circumference of the ellipse as determined by the
 11 center, x radius, and y radius. Offsets between 0.0 and 1.0 are positioned at a location
 12 interpolated between these points.

13 For full details of rendering of gradient brushes, including handling of offsets, please see §18.3.

1 13.8 Using a Brush as an Opacity Mask

2 Each pixel carries an alpha value ranging from 0.0 (fully transparent) to 1.0 (fully opaque). The
3 alpha value is used when blending elements to achieve the visual effect of transparency. Each
4 element can have an Opacity attribute by which the alpha value of each pixel is multiplied
5 uniformly.

6 The OpacityMask property also allows the specification of per-pixel opacity, which controls how
7 rendered content is blended with its destination. The opacity specified by the opacity mask is
8 combined multiplicatively with any opacity that can already be present in the alpha channel of
9 the contents. The per-pixel opacity specified by the opacity mask is determined by the alpha
10 channel of each pixel in the mask. The color data is ignored.

11 The alpha value of the area not marked by the brush is 0.0. The required computations for
12 transparently blending two elements when rendering, also known as *alpha blending*, are
13 described in §18.4.

14 An opacity mask always has a brush [as the](#) child element ([see §14.5](#)).

15 *Example 13–28. Opacity mask with linear gradient*

16 The following markup illustrates how an opacity mask is used to create a fade effect on a glyph.
17 The opacity mask is a linear gradient that fades from opaque black to transparent black.

```
18 <FixedPage Height="1056" Width="816" xml:lang="en-US">
19   <Glyphs
20     OriginX="25"
21     OriginY="50"
22     UnicodeString="This is a fading text example."
23     FontUri=" ../Resources/Fonts/ ../Resources/Fonts/Times.TTF"
24     FontRenderingEmSize="32">
25     <Glyphs.OpacityMask>
26       <LinearGradientBrush
27         StartPoint="25,0"
28         EndPoint="450,0"
29         MappingMode="Absolute">
30         <LinearGradientBrush.GradientStops>
31           <GradientStop Color="#FF000000" Offset="0" />
32           <GradientStop Color="#00000000" Offset="1" />
33         </LinearGradientBrush.GradientStops>
34       </LinearGradientBrush>
35     </Glyphs.OpacityMask>
36     <Glyphs.Fill>
37       <SolidColorBrush Color="#000000" />
38     </Glyphs.Fill>
39   </Glyphs>
40 </FixedPage>
```


1 This markup is rendered as follows:



2
3 *end example]*

4 *Example 13–29. Opacity mask with radial gradient*

5 In the following markup, the opacity mask is a radial gradient:

```

6     <FixedPage Width="816" Height="1056" xml:lang="en-US">
7         <Path>
8             <Path.OpacityMask>
9                 <RadialGradientBrush
10                    MappingMode="Absolute"
11                    Center="200,300"
12                    GradientOrigin="200,300"
13                    RadiusX="200"
14                    RadiusY="300">
15                    <RadialGradientBrush.GradientStops>
16                        <GradientStop Color="#FF000000" Offset="0" />
17                        <GradientStop Color="#20000000" Offset="1" />
18                    </RadialGradientBrush.GradientStops>
19                </RadialGradientBrush>
20            </Path.OpacityMask>
21            <Path.Fill>
22                <ImageBrush
23                    Viewbox="0,0,400,600"
24                    ViewboxUnits="Absolute"
25                    Viewport="0,0,400,600"
26                    ViewportUnits="Absolute"
27                    TileMode="None"
28                    ImageSource="images/jpeg3.jpg" />
29            </Path.Fill>
30            <Path.Data>
31                <PathGeometry>
32                    <PathFigure StartPoint="0,0" IsClosed="true">
33                        <PolyLineSegment Points="400,0 400,600 0,600" />
34                    </PathFigure>
35                </PathGeometry>
36            </Path.Data>
37        </Path>
38    </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 summarized in
 3 Table 14–1 and Table 14–2 and detailed in the following sections. Other than the Name,
 4 FixedPage.NavigateUri, and xml:lang attributes, these properties compose their results from
 5 parent to child, as described in §18.5~~18.4.1~~.

6 *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.
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.
FixedPage.NavigateUri	<Canvas> <Glyphs> <Path>	Defines a hyperlink source. For more information, see §16.2.

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 can 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.
<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 with
4 previously specified elements, also known as alpha blending. The opacity value MUST fall within
5 the 0 (fully transparent) to 1 (fully opaque) range, inclusive [[M7.12](#)~~M2.72~~].

6 For more information, see §18.4.

7 **14.2 Resources and Resource References**

8 Fixed page markup supports the concept of resources. A *resource* is a reusable property value
9 that is expressed in markup, identified by a key, and stored in a *resource dictionary*. In general,
10 any property value that can be expressed using property element syntax can be held in a
11 resource dictionary.

12 Each resource in a resource dictionary has a key. Any property that specifies its value by
13 referencing a resource key in a resource dictionary is called a *resource reference*.

1 The <Canvas> and <FixedPage> elements can carry a resource dictionary. A resource
 2 dictionary is expressed in markup by the <FixedPage.Resources> or <Canvas.Resources>
 3 property element. Individual resource values MUST be specified within a resource dictionary
 4 [M7.1].

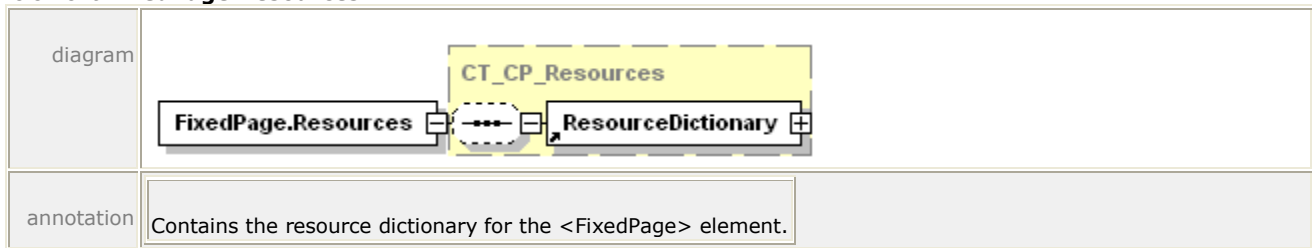
5 The <Canvas.Resources> or <FixedPage.Resources> property elements MUST precede any
 6 property elements of the <Canvas> or <FixedPage> elements [M2.72]. Likewise, they MUST
 7 precede any path, glyphs, or canvas children of the <Canvas> or <FixedPage> elements
 8 [M7.14M2.72].

9 Alternatively, resource dictionaries MAY be specified in separate parts and referenced from
 10 within the <FixedPage.Resources> or <Canvas.Resources> property element [O7.1]. Such a
 11 *remote resource dictionary* can be shared across multiple pages. [*Example: By defining a brush*
 12 *in a remote resource dictionary, graphical elements that are common to multiple pages can be*
 13 *reused. end example*]

14 The <Path>, <Glyphs>, and <Canvas> elements can appear as a resource definition solely for
 15 the purpose of using these elements in the Visual attribute of a <VisualBrush> element.
 16 Brushes and geometries appear in resource dictionaries far more frequently.

17 14.2.1 <FixedPage.Resources> Element

18 element **FixedPage.Resources**



19 *Example 14-1. <FixedPage.Resources> usage*

```

20 <FixedPage Width="816" Height="1056" xml:lang="en-US"
21   xmlns="http://schemas.microsoft.com/xps/2005/06"
22   xmlns:x=
23   "http://schemas.microsoft.com/xps/2005/06/resourcedictionary-key">
24   <FixedPage.Resources>
25     <ResourceDictionary>
26       <PathGeometry x:Key="Rectangle">
27         <PathFigure StartPoint="20,20" IsClosed="true">
28           <PolyLineSegment Points="120,20 120,70 20,70" />
29         </PathFigure>
30       </PathGeometry>
31     </ResourceDictionary>
32   </FixedPage.Resources>
33   <Path Stroke="#000000"
34     StrokeThickness="1"
35     Data="{StaticResource Rectangle}" />
36 </FixedPage>

```

1 This markup is rendered as follows:

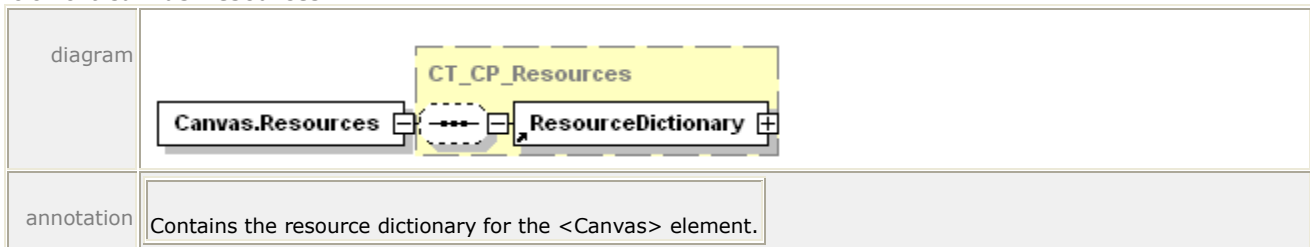


2

3 *end example]*

4 **14.2.2 <Canvas.Resources> Element**

5 element **Canvas.Resources**



6

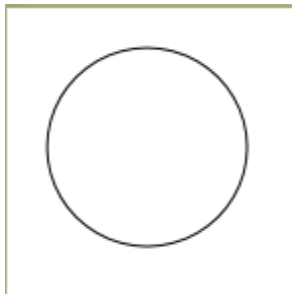
7 *Example 14–2. <Canvas.Resources> usage*

```

8 <Canvas
9   xmlns:x=
10  "http://schemas.microsoft.com/xps/2005/06/resourcedictionary-key">
11  <Canvas.Resources>
12    <ResourceDictionary>
13      <PathGeometry x:Key="Circle">
14        <PathFigure StartPoint="20,70">
15          <ArcSegment
16            Point="120,70"
17            Size="50,50"
18            RotationAngle="0"
19            IsLargeArc="true"
20            SweepDirection="Clockwise" />
21          <ArcSegment
22            Point="20,70"
23            Size="50,50"
24            RotationAngle="0"
25            IsLargeArc="true"
26            SweepDirection="Clockwise" />
27        </PathFigure>
28      </PathGeometry>
29    </ResourceDictionary>
30  </Canvas.Resources>
31  <Path Stroke="#000000"
32    StrokeThickness="1"
33    Data="{StaticResource Circle}" />
34 </Canvas>

```

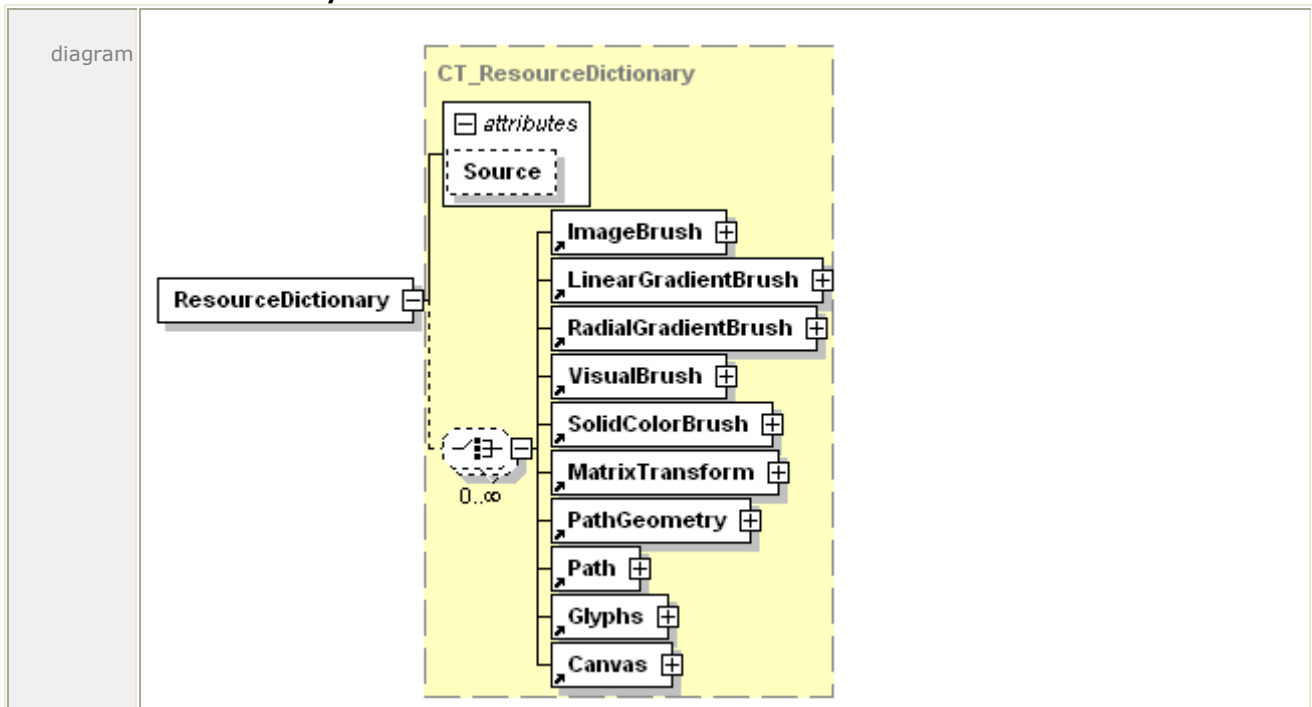
1 This markup is rendered as follows:



2
3 *end example]*

4 **14.2.3 <ResourceDictionary> Element**

5 element **ResourceDictionary**



attributes	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 Defines a set of reusable resource definitions that can be used as property values in the fixed page markup.

6 The <FixedPage.Resources> and <Canvas.Resources> property elements contain exactly one
 7 <ResourceDictionary> element. A resource dictionary contains *resource definition* element
 8 entries. Each resource definition has a key specified in the x:Key attribute that is unique within
 9 the scope of the resource dictionary. The x:Key attribute is included in the Resource Dictionary
 10 namespace specified in §A.

1 Resource dictionaries can be declared inline inside a <FixedPage.Resources> or
 2 <Canvas.Resources> element, or they MAY be defined in a separate part and referenced by a
 3 <ResourceDictionary> element inside a <FixedPage.Resources> or <Canvas.Resources>
 4 element [O7.1]. This allows resource dictionaries to be shared across parts. [*Example: A single*
 5 *resource dictionary can be used by every fixed page in the XPS Document. end example*]
 6 See §14.2.3.1 for more details.

7 A resource definition MAY reference another resource defined previously in the same resource
 8 dictionary [O7.2]. If the resource dictionary does not appear in a separate part, a resource
 9 definition MAY reference a previously defined resource in a resource dictionary of a parent or
 10 ancestor <Canvas> or <FixedPage> element [O7.3].

11 Namespace prefixes in resource definitions MUST apply in the context of the definition, rather
 12 than in the context of the resource reference [M7.2]. An xml:lang attribute within a resource
 13 definition MUST be interpreted in the context of the resource reference, not the resource
 14 definition [M7.3].

15 *Example 14–3. Resource dictionary markup*

16 The following markup defines two geometries, one for a rectangle, and the other for a circle:

```

17 <Canvas
18     xmlns:x=
19     "http://schemas.microsoft.com/xps/2005/06/resourcedictionary-key">
20     <Canvas.Resources>
21         <ResourceDictionary>
22             <PathGeometry x:Key="Rectangle">
23                 <PathFigure StartPoint="20,20" IsClosed="true">
24                     <PolyLineSegment Points="120,20 120,70 20,70" />
25                 </PathFigure>
26             </PathGeometry>
27             <PathGeometry x:Key="Circle">
28                 <PathFigure StartPoint="20,70">
29                     <ArcSegment
30                         Point="120,70"
31                         Size="50,50"
32                         RotationAngle="0"
33                         IsLargeArc="true"
34                         SweepDirection="Clockwise" />
35                     <ArcSegment
36                         Point="20,70"
37                         Size="50,50"
38                         RotationAngle="0"
39                         IsLargeArc="true"
40                         SweepDirection="Clockwise" />
41                 </PathFigure>
42             </PathGeometry>
43         </ResourceDictionary>
44     </Canvas.Resources>
45     <Path Data="{StaticResource Rectangle}">
46         <Path.Fill>
47             <SolidColorBrush Color="#FF0000" />
48         </Path.Fill>
49     </Path>
50 </Canvas>

```


1 *end example]*

2 **14.2.3.1 Remote Resource Dictionaries**

3 A resource dictionary MAY be defined in a separate part [O7.1]. This is referred to as a *remote*
4 *resource dictionary*. A remote resource dictionary MUST follow the requirements above that
5 apply to all resource dictionaries [M7.4]. A remote resource dictionary MUST NOT contain any
6 resource definition children that reference another remote resource dictionary [M7.5].

7 The <FixedPage.Resources> and <Canvas.Resources> property elements include a remote
8 resource dictionary via reference, using the Source attribute of the <ResourceDictionary>
9 element.

10 A <ResourceDictionary> element that specifies a remote resource dictionary in its Source
11 attribute MUST NOT contain any resource definition children [M7.6]. <FixedPage.Resources>
12 and <Canvas.Resources> elements that include a remote resource dictionary MUST include
13 exactly one <ResourceDictionary> element [M2.72].

14 A remote Resource Dictionary part MUST be added as a Required Resource relationship from
15 the FixedPage part that references it [M2.10]. In addition, producers MUST add each resource
16 such as fonts or images referenced in the Resource Dictionary part as a Required Resource
17 relationship from the FixedPage part (*not* the Resource Dictionary part) to the indirectly
18 required resource, even if the particular fixed page does not reference the resource [M2.10].
19 For more information, see §D.3.

20 Inline references to fonts or images in remote resource dictionary entries MUST be interpreted
21 with the same base URI as the Remote Resource Dictionary part, not from the base URI of the
22 part referring to the particular remote resource dictionary entry [M7.7].

23 *Example 14-4. A remote resource dictionary and reference*

24 The following markup defines a resource dictionary that contains two geometries, one for a
25 rectangle and the other for a circle:

```
26 <!-- Contents of /resource.xaml -->
27 <ResourceDictionary xmlns="http://schemas.microsoft.com/xps/2005/06"
28     xmlns:x=
29     "http://schemas.microsoft.com/xps/2005/06/resourcedictionary-key">
30     <PathGeometry x:Key="Rectangle">
31         <PathFigure StartPoint="20,20" IsClosed="true">
32             <PolyLineSegment Points="120,20 120,70 20,70" />
33         </PathFigure>
34     </PathGeometry>
35     <PathGeometry x:Key="Circle">
36         <PathFigure StartPoint="20,70">
37             <ArcSegment
38                 Point="120,70"
39                 Size="50,50"
40                 RotationAngle="0"
41                 IsLargeArc="true"
42                 SweepDirection="Clockwise" />
43             <ArcSegment
44                 Point="20,70"
45                 Size="50,50"
46                 RotationAngle="0"
47                 IsLargeArc="true"
```

```

1         SweepDirection="Clockwise" />
2     </PathFigure>
3 </PathGeometry>
4 </ResourceDictionary>

```

5 The following markup references the previously defined resource dictionary:

```

6 <Canvas
7     xmlns:x=
8     "http://schemas.microsoft.com/xps/2005/06/resourcedictionary-key">
9     <Canvas.Resources>
10        <ResourceDictionary Source="/resource.xaml"/>
11    </Canvas.Resources>
12    <Path Data="{StaticResource Rectangle}">
13        <Path.Fill>
14            <SolidColorBrush Color="#FF0000" />
15        </Path.Fill>
16    </Path>
17 </Canvas>

```

18 *end example]*

19 **14.2.4 Resource References**

20 To set a property value to a defined resource, use the form:

```
21 {StaticResource key}
```

22 Where *key* is the same string specified with *x:Key* in the resource definition.

23 The context of the resource reference determines how defined resources are rendered (such as
24 the transformation matrix to be applied). Specifically, the effective coordinate space for
25 rendering the referenced resource is a composition of the effective coordinate space of the
26 referring element plus any Transform or RenderTransform properties included in the resource
27 definition itself.

28 It is considered an error [condition](#) if a static resource reference cannot be resolved, or if it *can*
29 be resolved but the resource type does not match the usage at the location of reference.

30 *Example 14-5. Using a resource reference to fill a brush*

31 In the following markup, the rectangular region defined by the geometry specified in the
32 dictionary is filled by a solid color brush:

```

33 <Canvas
34     xmlns:x=
35     "http://schemas.microsoft.com/xps/2005/06/resourcedictionary-key">
36     <Canvas.Resources>
37         <ResourceDictionary>
38             <PathGeometry x:Key="Rectangle">
39                 <PathFigure StartPoint="20,20" IsClosed="true">
40                     <PolyLineSegment Points="120,20 120,70 20,70" />
41                 </PathFigure>
42             </PathGeometry>
43         </ResourceDictionary>
44     </Canvas.Resources>
45     <Path Data="{StaticResource Rectangle}">

```

```

1      <Path.Fill>
2          <SolidColorBrush Color="#FF0000" />
3      </Path.Fill>
4  </Path>
5  </Canvas>

```

6 *end example]*

7 **14.2.5 Scoping Rules for Resolving Resource References**

8 The value of the x:Key attribute MUST be unique within the resource dictionary [M2.72].
9 However, the resource dictionary of a <Canvas> element MAY re-use an x:Key value defined in
10 the resource dictionary of a parent or ancestor <Canvas> or <FixedPage> element [O7.5].
11 Resource references are resolved from the innermost to the outermost resource dictionary.

12 A resource definition MAY reference a previously defined resource with the same name that is
13 defined in an ancestor resource dictionary [O7.6]; the reference MUST be resolved before the
14 redefined resource is added to the dictionary [M7.8].

15 A resource definition MAY reference another resource defined prior to the point of reference,
16 including a resource previously defined within the same resource dictionary [O7.2]. If a
17 resource definition references another resource, the reference MUST be resolved in the context
18 of the resource definition, not in the context of the resource use [M7.9].

19 To find a resource, the nearest parent or ancestor canvas or fixed page is searched. If the
20 desired name is not defined in the initially searched resource dictionary, then the next-nearest
21 parent or ancestor canvas or fixed page is searched. An error [condition](#) occurs if the search has
22 continued to the root <FixedPage> element and a specified resource has not been found. This
23 search occurs only within the containing FixedPage part.

24 *Example 14–6. Using scoping rules*

```

25  <FixedPage
26      xmlns="http://schemas.microsoft.com/xps/2005/06"
27      xmlns:x=
28      "http://schemas.microsoft.com/xps/2005/06/resourcedictionary-key"
29      Height="1056" Width="816" xml:lang="en-US">
30      <FixedPage.Resources>
31          <ResourceDictionary>
32              <SolidColorBrush x:Key="FavoriteColorFill" Color="#808080" />
33          </ResourceDictionary>
34      </FixedPage.Resources>
35      <Canvas>
36          <Canvas.Resources>
37              <ResourceDictionary>
38                  <SolidColorBrush x:Key="FavoriteColorFill"
39                      Color="#000000" />
40              </ResourceDictionary>
41          </Canvas.Resources>
42          <!-- The following path is filled with color #000000 -->
43          <Path Fill="{StaticResource FavoriteColorFill}">
44              <Path.Data>
45                  ...
46              </Path.Data>
47          </Path>
48      </Canvas>

```

```

1      <!-- The following path is filled with color #000000 -->
2      <Path Fill="{StaticResource FavoriteColorFill}">
3          <Path.Data>
4              ...
5          </Path.Data>
6      </Path>
7  </Canvas>
8 </Canvas>
9  <!-- The following path is filled with color #808080 -->
10 <Path Fill="{StaticResource FavoriteColorFill}">
11     <Path.Data>
12         ...
13     </Path.Data>
14 </Path>
15 </FixedPage>

```

16 *end example]*

17 **14.2.6 Support for Markup Compatibility**

18 If a resource dictionary contains Markup Compatibility and Extensibility elements and
 19 attributes, the processing of the Markup Compatibility and Extensibility markup MUST occur in
 20 the context of the definition of the resource dictionary, not in the context of resource references
 21 [M2.10].

22 **14.3 Clipping**

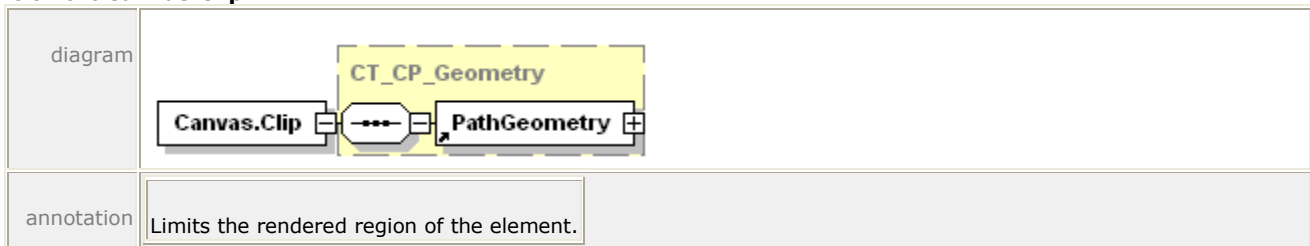
23 The Clip property specifies a geometric area that restricts the rendered region of an element.

24 The geometry is specified by a child <PathGeometry> element as detailed in §11.2, or by
 25 abbreviated geometry syntax, described in §11.2.3.

26 The default fill rule for geometries that do not specify a value is EvenOdd.

27 **14.3.1 <Canvas.Clip> Element**

28 element **Canvas.Clip**



29 The <Canvas.Clip> property element applies to all child and descendant elements of the
 30 canvas.

31 *Example 14–7. Canvas clip markup and rendering*

```

32 <Canvas>
33     <Canvas.Clip>
34         <PathGeometry>
35             <PathFigure StartPoint="25,25" IsClosed="true">

```

```

1         <PolyLineSegment Points="60,25 70,60 80,25 115,25
2           115,115 80,115 70,80 60,115 25,115" />
3       </PathFigure>
4     </PathGeometry>
5 </Canvas.Clip>
6 <Path Fill="#9999CC">
7   <Path.Data>
8     <PathGeometry>
9       <PathFigure StartPoint="20,70">
10        <ArcSegment
11          Point="120,70"
12          Size="50,50"
13          RotationAngle="0"
14          IsLargeArc="true"
15          SweepDirection="Clockwise" />
16        <ArcSegment
17          Point="20,70"
18          Size="50,50"
19          RotationAngle="0"
20          IsLargeArc="true"
21          SweepDirection="Clockwise" />
22      </PathFigure>
23    </PathGeometry>
24  </Path.Data>
25 </Path>
26 </Canvas>

```

27 This markup is rendered as follows:

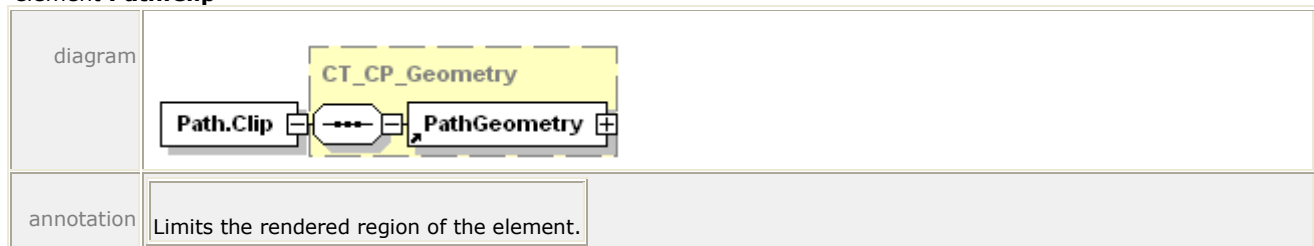


28

29 *end example]*

30 14.3.2 <Path.Clip> Element

31 element **Path.Clip**



32 A clipping region can also be applied to a specific path.

1 *Example 14-8. <Path.Clip> usage*

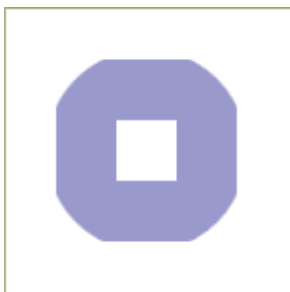
2 The following markup describes a complex clipping behavior:

```

3     <Path Fill="#9999CC">
4         <Path.Clip>
5             <PathGeometry>
6                 <PathFigure StartPoint="25,25" IsClosed="true">
7                     <PolyLineSegment Points="115,25 115,115 25,115" />
8                 </PathFigure>
9                 <PathFigure StartPoint="55,55" IsClosed="true">
10                    <PolyLineSegment Points="85,55 85,85 55,85" />
11                </PathFigure>
12            </PathGeometry>
13        </Path.Clip>
14        <Path.Data>
15            <PathGeometry>
16                <PathFigure StartPoint="20,70">
17                    <ArcSegment
18                        Point="120,70"
19                        Size="50,50"
20                        RotationAngle="0"
21                        IsLargeArc="true"
22                        SweepDirection="Clockwise" />
23                    <ArcSegment
24                        Point="20,70"
25                        Size="50,50"
26                        RotationAngle="0"
27                        IsLargeArc="true"
28                        SweepDirection="Clockwise" />
29                </PathFigure>
30            </PathGeometry>
31        </Path.Data>
32    </Path>

```

33 This markup is rendered as follows:

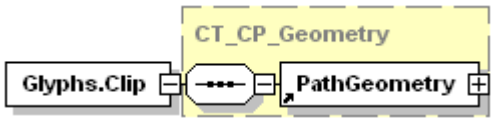


34

35 *end example]*

36 **14.3.3 <Glyphs.Clip> Element**

37 element **Glyphs.Clip**

diagram	
annotation	<p>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.</p>

1 *Example 14-9. <Glyphs.Clip> usage*

2 The following markup uses abbreviated geometry syntax to define the clipping region:

```

3 <Glyphs
4   Fill="#000000"
5   Clip="M 0,0 L 180,0 L 180,140 L 0,140 Z M 20,60 L 140,60 L 140,80
6     L 20,80 Z"
7   OriginX="20"
8   OriginY="130"
9   UnicodeString="N"
10  FontRenderingEmSize="170"
11  FontUri="../Resources/Fonts/Timesbd.ttf" />

```

12 This markup is rendered as follows:



13

14 *end example]*

15 **14.4 Positioning Content**

16 Content is positioned according to the properties specified for the fixed page or canvas, the
 17 properties specified for elements within the fixed page or canvas, and the compositional rules
 18 defined for the fixed payload namespace.

19 Elements are positioned relative to the current origin (0,0) of the coordinate space. The current
 20 origin can be moved by setting the RenderTransform property of a canvas, path, or glyph. The
 21 render transformation establishes a new coordinate frame for all children of the parent element.

22 Geometries and brushes can be manipulated in a similar way by setting the Transform
 23 property. The transform results are concatenated with the current render transformation to
 24 create an effective render transformation for the local element.

25 The RenderTransform and Transform properties both specify an affine matrix transformation to
 26 the local coordinate space, using the <MatrixTransform> element as their value. An

1 abbreviated matrix transformation syntax MAY be used to specify a RenderTransform or Transform
 2 attribute value [O7.7M2.72].

3 **14.4.1 <MatrixTransform> Element**

4 element **MatrixTransform**

diagram						
attributes	Name	Type	Use	Default	Fixed	Annotation
	Matrix	<u>ST_Matrix</u>	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.					

5 The <MatrixTransform> element defines an arbitrary affine matrix transformation used to
 6 manipulate the coordinate systems of elements. A 3x3 matrix is used for transformations in an
 7 x,y plane. Affine transformation matrices can be multiplied to form any number of linear
 8 transformations, such as rotation and skew (shear), followed by translation. An affine
 9 transformation matrix has its final column equal to 0,0,1, so only the members in the first two
 10 columns are specified.

$$\begin{bmatrix} M11 & M12 & 0 \\ M21 & M22 & 0 \\ \text{OffsetX} & \text{OffsetY} & 1 \end{bmatrix}$$

11 This structure is specified by the Matrix attribute of the <MatrixTransform> element as the six
 12 numbers in the first two columns. [Example: "M11,M12,M21,M22,OffsetX,OffsetY". end
 13 example]

14 A matrix transform can also be specified as a RenderTransform or Transform property attribute
 15 using the following abbreviated matrix transformation syntax:

16 M11,M12,M21,M22,OffsetX,OffsetY

17 The values M11, M12, M21, and M22 control linear transformations such as rotation and skew,
 18 while OffsetX and OffsetY provide positional translation. Some typical affine matrix
 19 transformation examples follow.

1 *Example 14-10. Matrix scaling*

$$\begin{bmatrix} \text{X scale-factor} & 0 & 0 \\ 0 & \text{Y scale-factor} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

2 *end example]*

3 *Example 14-11. Matrix reversing the x axis*

$$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

4 *end example]*

5 *Example 14-12. Matrix reversing the y axis*

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

6 *end example]*

7 *Example 14-13. Matrix skewing*

$$\begin{bmatrix} 1 & \text{Y skew-factor} & 0 \\ \text{X skew factor} & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

8 *end example]*

9 *Example 14-14. Matrix Rotating*

$$\begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

10 *end example]*

11 *Example 14-15. Matrix positioning*

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \text{OffsetX} & \text{OffsetY} & 1 \end{bmatrix}$$

12 *end example]*

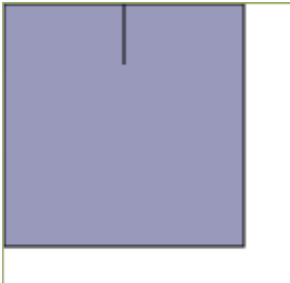
1 *Example 14-16. <MatrixTransform> usage*

2 The following markup describes a box (with the top edge marked) that is rotated 90° and
3 shifted 50 units down and to the right:

```
4 <Path  
5   Stroke="#000000"  
6   Fill="#9999BB"  
7   Data="M 0,0 L 60,0 L 60,25 L 60,0 L 120,0 L 120,120 L 0,120 Z">  
8   <Path.RenderTransform>  
9     <MatrixTransform Matrix="0,1,-1,0,170,50" />  
10  </Path.RenderTransform>  
11 </Path>
```

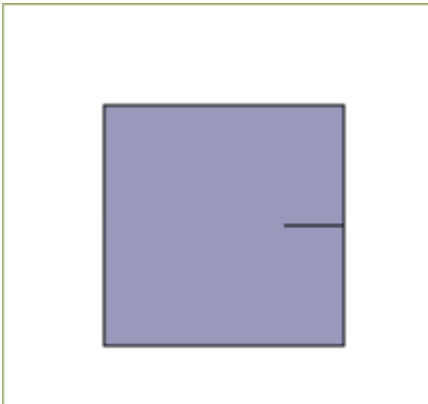
12 Since the x origin has been shifted, the overall box must be additionally shifted the width of the
13 box to achieve the desired visual effect.

14 Before the render transformation, the box appears like this:



15

16 After the render transformation, the box appears like this:



17

18 *end example]*

1 *Example 14–17. Using abbreviated matrix transformation syntax*

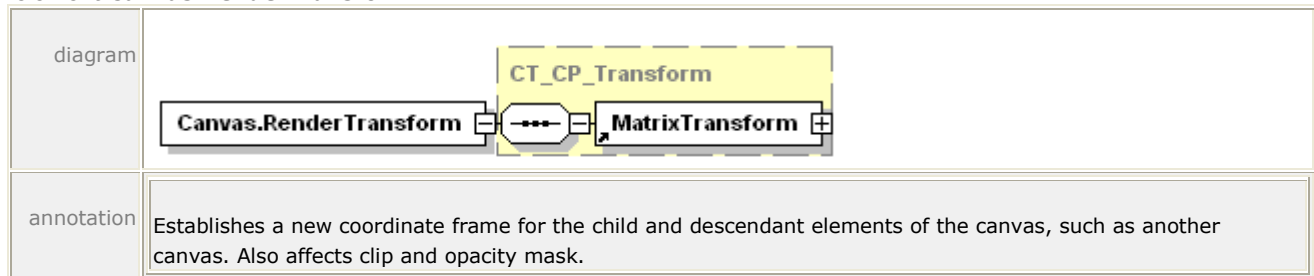
2 The following markup uses abbreviated syntax to produce the above image:

```
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:

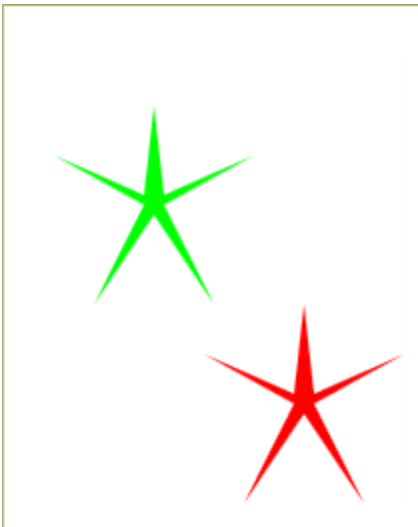
```
14 <Canvas>
15   <Canvas.Resources>
16     <ResourceDictionary>
17       <PathGeometry x:Key="StarFish">
18         <PathFigure StartPoint="50,0" IsClosed="true">
19           <PolyLineSegment Points="55,45 100,25 55,50 80,100 50,55
20             20,100 45,50 0,25 45,45" />
21         </PathFigure>
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>
```

```

1      <Canvas.RenderTransform>
2          <MatrixTransform Matrix="1,0,0,1,100,150" />
3      </Canvas.RenderTransform>
4      <Path Data="{StaticResource StarFish}">
5          <Path.Fill>
6              <SolidColorBrush Color="#FF0000" />
7          </Path.Fill>
8      </Path>
9  </Canvas>
10 </Canvas>

```

11 This markup is rendered as follows:



12

13 *end example]*

14 14.4.3 <Path.RenderTransform> Element

15 element **Path.RenderTransform**

diagram	
annotation	<p>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.</p>

16 *Example 14-19. <Path.RenderTransform> usage*

17 The following markup describes a y-skew transformation applied to a circular path. (Before the
18 render transformation, the middle of the right edge of the circle was marked with a horizontal
19 line.)

```

20 <Path
21     Fill="#999999"
22     Stroke="#000000"
23     Data="M 20,70 A 50,50 0 1 1 120,70 L 100,70 L 120,70 A 50,50 0 1 1
24         20,70 Z" >

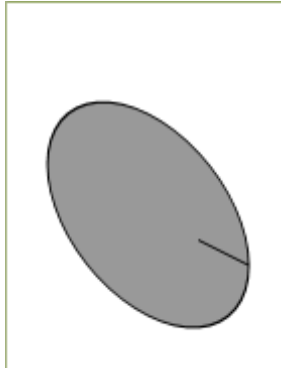
```

```

1      <Path.RenderTransform>
2          <MatrixTransform Matrix="1,0.5,0,1,0,0" />
3      </Path.RenderTransform>
4  </Path>

```

5 This markup is rendered as follows:



6
7 *end example]*

8 **14.4.4 <Glyphs.RenderTransform> Element**

9 element **Glyphs.RenderTransform**

diagram	
annotation	<p>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.</p>

10 *Example 14-20. <Glyphs.RenderTransform> usage*

11 The following markup describes the letter J, flipped vertically and repositioned.

```

12  <Glyphs
13      Fill="#000000"
14      OriginX="20"
15      OriginY="130"
16      UnicodeString="J"
17      FontRenderingEmSize="170"
18      FontUri="../Resources/Fonts/Timesbd.ttf" >
19      <Glyphs.RenderTransform>
20          <MatrixTransform Matrix="1,0,0,-1,0,150" />
21      </Glyphs.RenderTransform>
22  </Glyphs>

```

1 This markup is rendered as follows:



2

3 *end example]*

4 **14.4.5 <PathGeometry.Transform> Element**

5 element **PathGeometry.Transform**

diagram	
annotation	Specifies the local matrix transformation that is applied to all child and descendant elements of the path geometry before it is used for filling, clipping, or stroking.

6 *Example 14–21. <PathGeometry.Transform> usage*

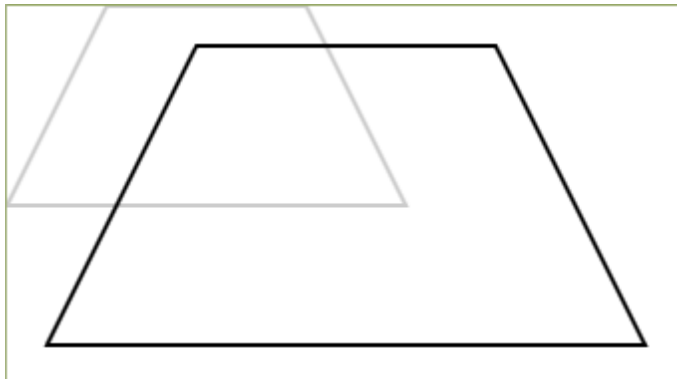
7 The following markup demonstrates a simple 150% zoom and positional transformation:

```

8     <Path StrokeThickness="2" Stroke="#000000">
9         <Path.Data>
10            <PathGeometry>
11                <PathGeometry.Transform>
12                    <MatrixTransform Matrix="1.5,0,0,1.5,20,20" />
13                </PathGeometry.Transform>
14                <PathFigure StartPoint="50,0" IsClosed="true">
15                    <PolyLineSegment Points="150,0 200,100 0,100" />
16                </PathFigure>
17            </PathGeometry>
18        </Path.Data>
19    </Path>

```

- 1 This markup is rendered as follows. The pre-transform path is indicated in light gray. Note that
 2 the stroke thickness did not change. If this transformation had been applied to the entire Path,
 3 the stroke thickness would also have increased by 150%.



4
 5 *end example]*

6 **14.4.6 <ImageBrush.Transform> Element**

7 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>

- 8 The Transform property can result in a non-rectangular (that is, skewed) viewport that defines
 9 the tile shape. In this circumstance, tile mode operations (FlipX, FlipY, and FlipXY) are treated
 10 as if the tile was rectangular, a larger tile was constructed from a 2-by-2 arrangement of
 11 regular tiles, the skew transform was applied afterward, and the new non-rectangular tile was
 12 tiled with adjacent edges and without flipping.

13 *Example 14-22. <ImageBrush.Transform> usage*

14 The following markup describes an image rotated 20° and repositioned within a path. The path
 15 itself remains untransformed; the viewport of the image brush is transformed instead.

```

16 <Path
17   StrokeThickness="5"
18   Stroke="#996666"
19   StrokeLineJoin="Round"
20   Data="M 25,25 L 350,25 L 355,250 L 25,250 Z">
21   <Path.Fill>
22     <ImageBrush
23       ImageSource="dog.jpg"
24       TileMode="Tile"
25       Viewbox="0,0,270,423"
26       ViewboxUnits="Absolute"
27       Viewport="75,75,90,125"

```

```

1      ViewportUnits="Absolute" >
2      <ImageBrush.Transform>
3          <MatrixTransform Matrix=".939,.342,-.342,.939,0,-80" />
4      </ImageBrush.Transform>
5  </ImageBrush>
6  </Path.Fill>
7  </Path>

```

8 This markup is rendered as follows:

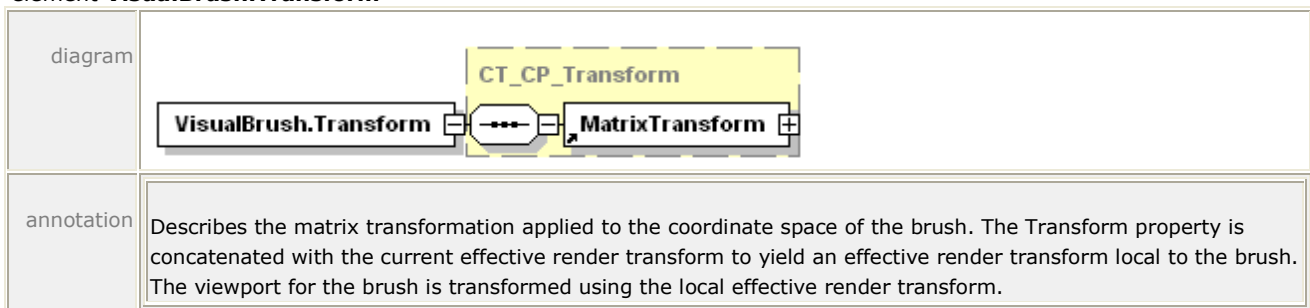


9

10 *end example]*

11 14.4.7 <VisualBrush.Transform> Element

12 element **VisualBrush.Transform**



13 The Transform property can result in a non-rectangular (that is, skewed) viewport that defines
14 the tile shape. In this circumstance, tile mode operations (FlipX, FlipY, and FlipXY) are treated
15 as if the tile was rectangular, a larger tile was constructed from a 2-by-2 arrangement of
16 regular tiles, the skew transform was applied afterward, and the new non-rectangular tile was
17 tiled with adjacent edges and without flipping.

18 *Example 14-23. <VisualBrush.Transform> usage*

19 The following markup describes a solid background and vertical pinstripe rotated 45° to fill a
20 frame:


```

1    <Path
2      StrokeThickness="5"
3      Stroke="#336666"
4      StrokeLineJoin="Round"
5      Data="M 25,25 L 365,25 L 365,250 L 25,250 Z M 70,70 L 320,70
6          L 320,205 L 70,205 Z">
7      <Path.Fill>
8        <VisualBrush
9          TileMode="Tile"
10         Viewbox="0,0,60,100"
11         ViewboxUnits="Absolute"
12         Viewport="25,25,50,50"
13         ViewportUnits="Absolute">
14         <VisualBrush.Transform>
15           <MatrixTransform Matrix=".707,.707,-.707,.707,0,0" />
16         </VisualBrush.Transform>
17         <VisualBrush.Visual>
18           <Canvas>
19             <Path
20               Fill="#99CCCC"
21               Data="M 0,0 L 60,0 L 60,100 L 0,100 Z" />
22             <Path
23               Stroke="#336666"
24               Data="M 0,0 L 0,100 M 20,0 L 20,100 M 40,0 L 40,100
25                 M 60,0 L 60,100 M 80,0 L 80,100" />
26             </Canvas>
27           </VisualBrush.Visual>
28         </VisualBrush>
29       </Path.Fill>
30     </Path>

```

31 This markup is rendered as follows:



32

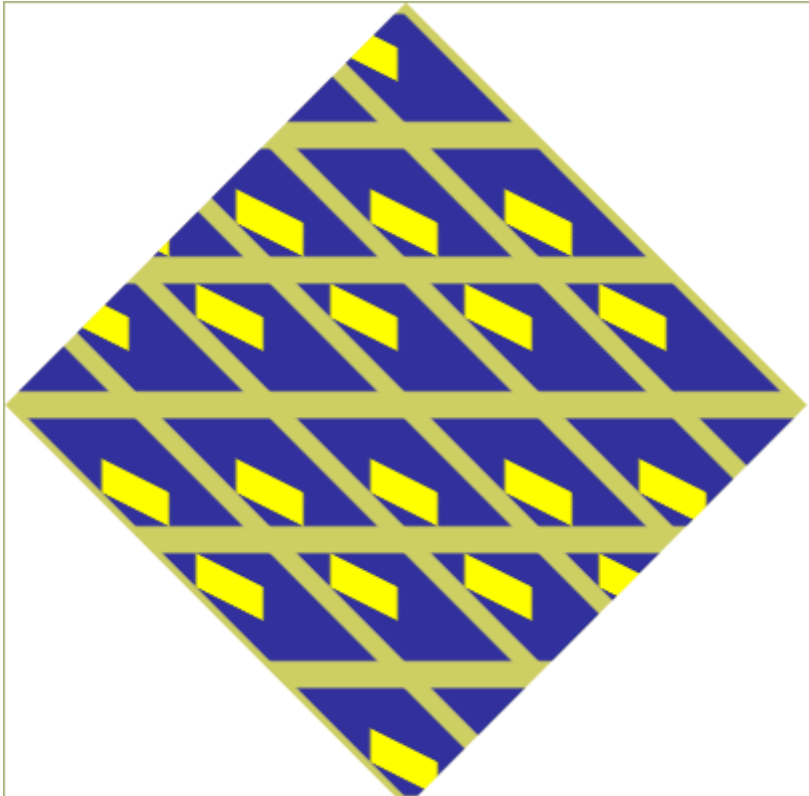
33 *end example]*

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      TileMode="FlipY">
13       <VisualBrush.Transform>
14         <MatrixTransform Matrix="1,0,1,1,0,0" />
15       </VisualBrush.Transform>
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>
```

27 This markup is rendered as follows:



1

2 *end example]*3 **14.4.8 <LinearGradientBrush.Transform> Element**4 element **LinearGradientBrush.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 start point and end point are transformed using the local effective render transform.</p>

5 *Example 14–25. <LinearGradientBrush.Transform> usage*

6 The following markup demonstrates a transform applied to the brush directly:

```

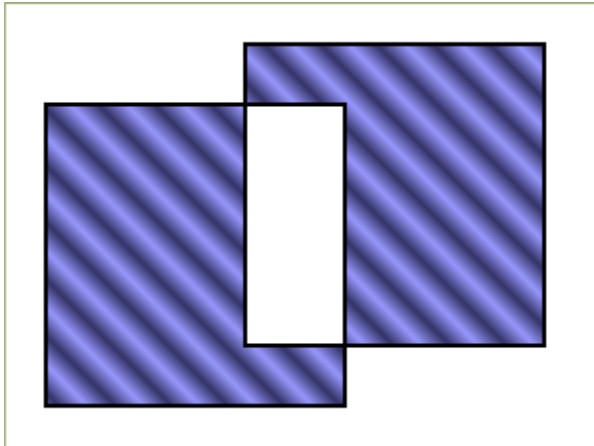
7     <Path Stroke="#000000" StrokeThickness="2" Data="M 20,50 L 170,50 L 170,200 L
8     20,200 Z M 120,20 L 270,20 L 270,170 120,170 Z">
9         <Path.Fill>
10            <LinearGradientBrush
11                MappingMode="Absolute"
12                StartPoint="0,0"
13                EndPoint="0,10"
14                SpreadMethod="Reflect">
15                <LinearGradientBrush.Transform>

```

```

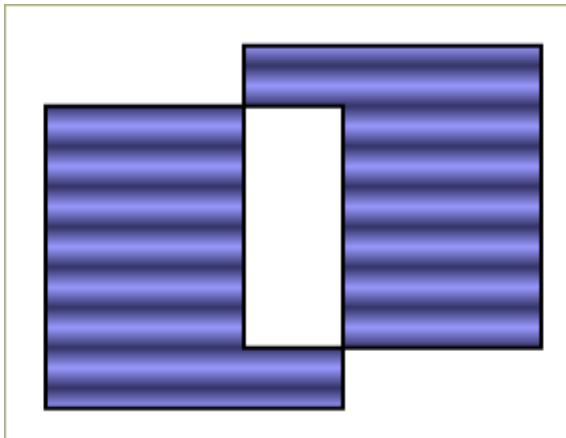
1         <MatrixTransform Matrix=".707,.707,-.707,.707,150,-30" />
2     </LinearGradientBrush.Transform>
3     <LinearGradientBrush.GradientStops>
4         <GradientStop Color="#9999FF" Offset="0.0"/>
5         <GradientStop Color="#333366" Offset="1.0"/>
6     </LinearGradientBrush.GradientStops>
7 </LinearGradientBrush>
8 </Path.Fill>
9 </Path>
    
```

10 This markup is rendered as follows:



11

12 Without the Transform property, this markup would be rendered as follows:

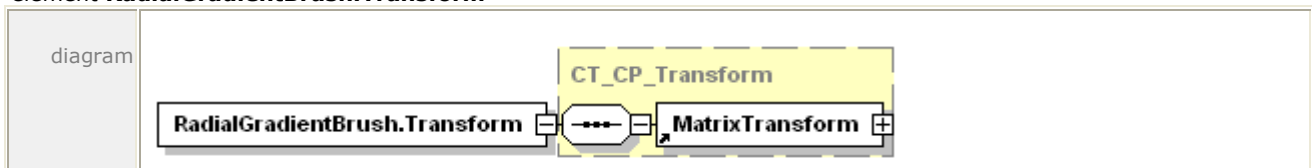


13

14 *end example]*

15 **14.4.9 <RadialGradientBrush.Transform> Element**

16 element **RadialGradientBrush.Transform**



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 ellipse defined by the center, gradient origin, x radius, and y radius vaules is transformed using the local effective render transform.

1 *Example 14–26. <RadialGradientBrush.Transform> usage*

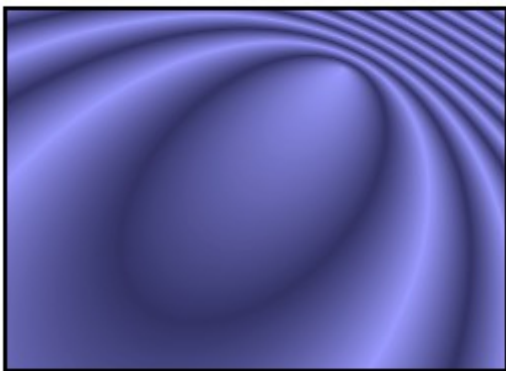
2 The following markup describes a rotation and reposition transform on a radial gradient:

```

3     <Path
4         Stroke="#000000"
5         StrokeThickness="2"
6         Data="M 20,20 L 270,20 L 270,200 L 20,200 Z">
7         <Path.Fill>
8             <RadialGradientBrush
9                 MappingMode="Absolute"
10                Center="80,90"
11                RadiusX="50"
12                RadiusY="80"
13                GradientOrigin="70,15"
14                SpreadMethod="Reflect">
15                <RadialGradientBrush.Transform>
16                    <MatrixTransform Matrix=".707,.707,-.707,.707,150,-10" />
17                </RadialGradientBrush.Transform>
18                <RadialGradientBrush.GradientStops>
19                    <GradientStop Color="#9999FF" Offset="0.0" />
20                    <GradientStop Color="#333366" Offset="1.0" />
21                </RadialGradientBrush.GradientStops>
22            </RadialGradientBrush>
23        </Path.Fill>
24    </Path>

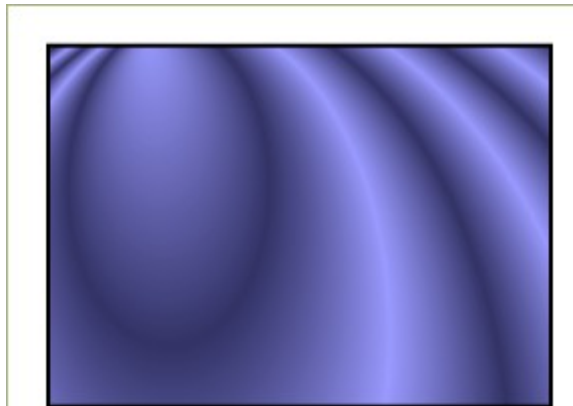
```

25 This markup is rendered as follows:



26

27 Without the Transform property, this markup is rendered as follows:



1

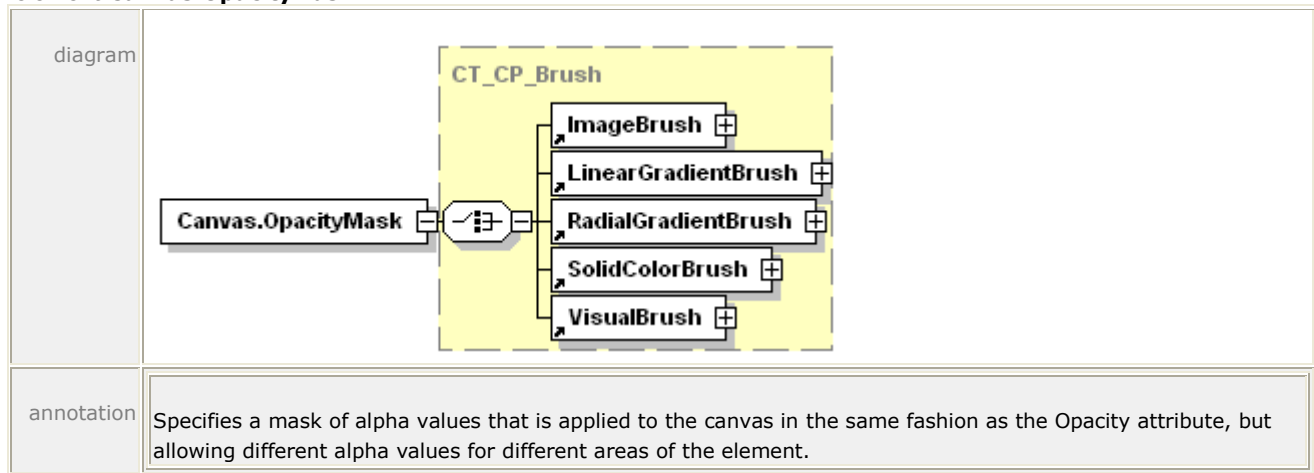
2 *end example]*

1 14.5 OpacityMask

2 The OpacityMask property defines a variable alpha mask for the parent element. The alpha for
3 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 other, but
8 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"

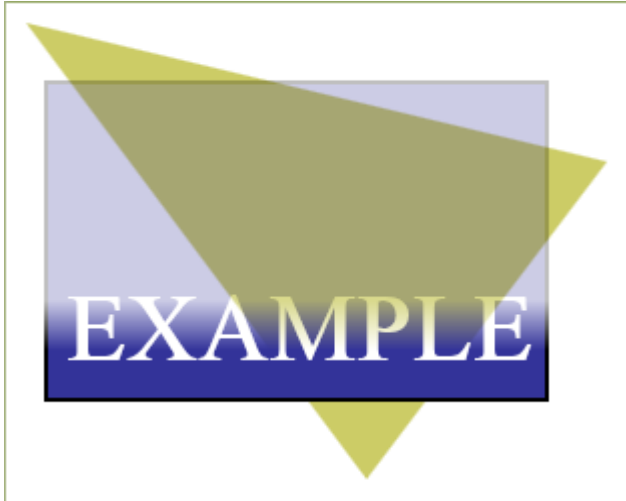
```

```

1      FontRenderingEmSize="48"
2      Fill="#FFFFFF" />
3  </Canvas>

```

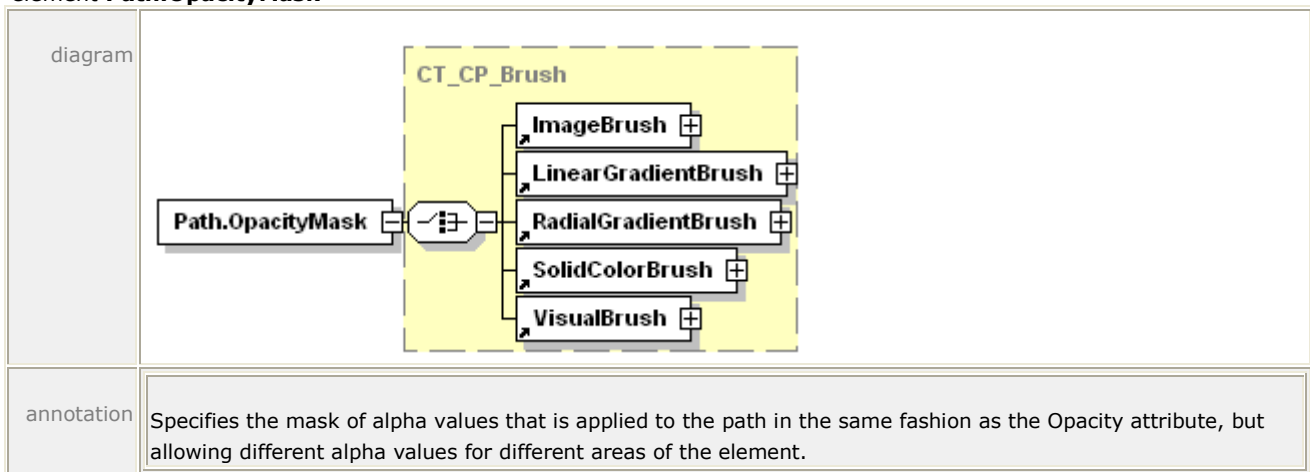
4 This markup is rendered as follows:



5
6 *end example]*

7 **14.5.2 <Path.OpacityMask> Element**

8 element **Path.OpacityMask**



9 *Example 14–28. <Path.OpacityMask> usage*

10 The following markup describes a path that has a linear gradient for the opacity mask and a
11 solid color brush for the fill:

```

12 <Path
13   Stroke="#000000"
14   StrokeThickness="2"
15   Fill="#CCCC66"
16   Data="M 135,10 L 270,250 L 20,250 Z" />
17 <Path
18   Stroke="#000000"

```

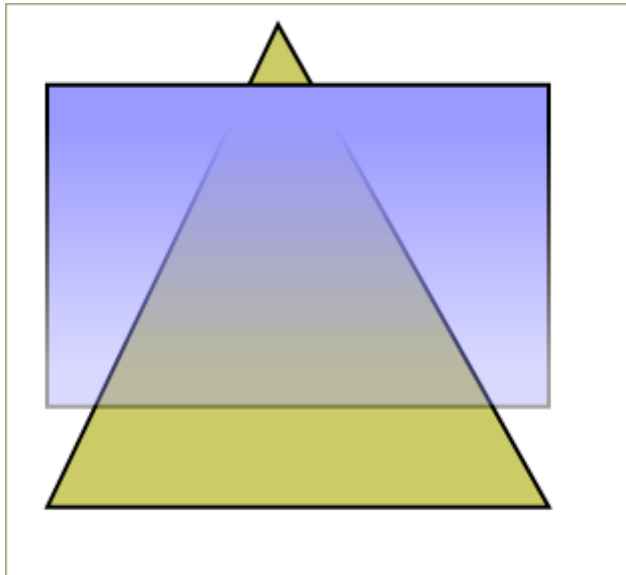


```

1      StrokeThickness="2"
2      Data="M 20,40 L 270,40 L 270,200 L 20,200 Z">
3      <Path.OpacityMask>
4          <LinearGradientBrush
5              MappingMode="Absolute"
6              StartPoint="0,60"
7              EndPoint="0,180"
8              SpreadMethod="Pad">
9              <LinearGradientBrush.GradientStops>
10                 <GradientStop Color="#FF000000" Offset="0.0" />
11                 <GradientStop Color="#60000000" Offset="1.0" />
12             </LinearGradientBrush.GradientStops>
13         </LinearGradientBrush>
14     </Path.OpacityMask>
15     <Path.Fill>
16         <SolidColorBrush Color="#9999FF" />
17     </Path.Fill>
18 </Path>

```

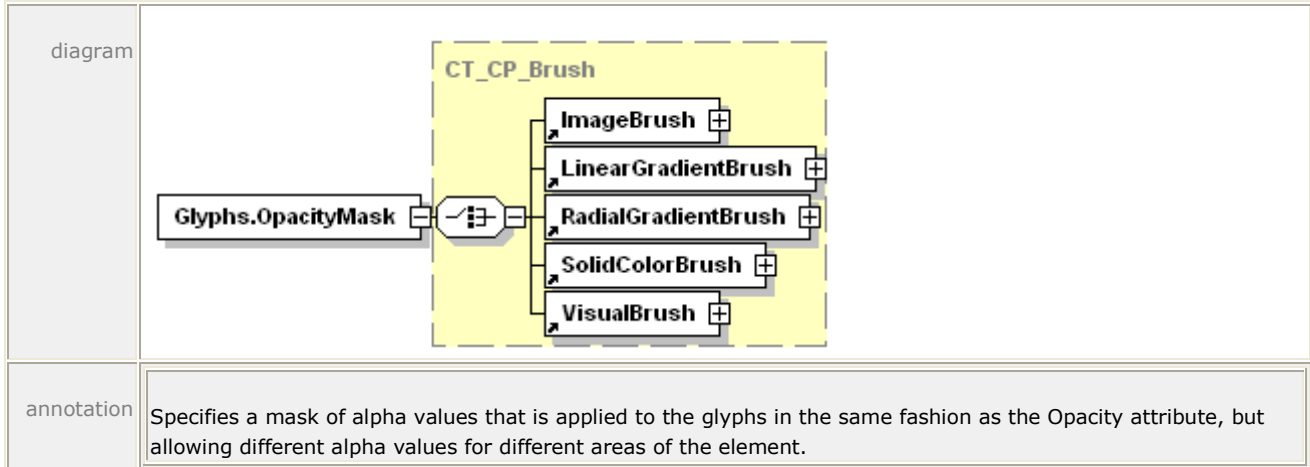
19 This markup is rendered as follows:



20
21 *end example]*

22 **14.5.3 <Glyphs.OpacityMask> Element**

23 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 effect:

```

3 <Path Fill="#CCCC66" Data="M 40,40 L 480,40 L 260,120 Z" />
4 <Glyphs
5   OriginX="20"
6   OriginY="95"
7   UnicodeString="EXAMPLE"
8   FontUri="../Resources/Fonts/Timesbd.ttf"
9   FontRenderingEmSize="100"
10  Fill="#000080">
11  <Glyphs.OpacityMask>
12    <VisualBrush
13      Viewbox="0,0,2,2"
14      ViewboxUnits="Absolute"
15      Viewport="0,0,6,6"
16      ViewportUnits="Absolute"
17      TileMode="Tile">
18      <VisualBrush.Visual>
19        <Path
20          Fill="#CC000000"
21          Data="M 0,0 L 1.5,0 L 1.5,1.5 L 0,1.5 Z" />
22        </VisualBrush.Visual>
23      </VisualBrush>
24    </Glyphs.OpacityMask>
25  </Glyphs>

```

26 This markup is rendered as follows:



27
28 *end example]*

15. Color

The mechanisms described in this clause for storing advanced color information in XPS Documents apply to both vector graphics (including text) and raster images. Color producers such as digital cameras and consumers such as printers can store and render significantly more color information than many display devices can render (typically 8 bits per channel). Storing the advanced color information in an XPS Document and passing it through to printing consumers enables greater end-to-end color fidelity.

15.1 Color Support

XPS Documents support sRGB and other color spaces, including scRGB, CMYK, N-Channel, and named colors. Consumers MUST support the following color features:

- [Grayscale colors \(single channel\) in vector data, with and without alpha \[M8.56\]](#)
- [Grayscale colors in image data, using the JPEG, PNG, TIFF, or HDPhoto image formats \[M8.57\]](#)
- sRGB colors (8 bit-per-channel) in vector data, with and without alpha [M8.1]
- sRGB colors in image data, using the JPEG, PNG, TIFF, or Windows Media Photo image formats [M8.2]
- scRGB color specification in vector data, with and without alpha [M8.3]
- scRGB colors in image data, using the Windows Media Photo image format [M8.4]
- CMYK colors in vector data [M8.5]
- CMYK colors in image data, using the TIFF or Windows Media Photo image formats [M8.6]
- N-Channel colors in vector data [M8.7]
- N-Channel colors in image data, using the Windows Media Photo image format [M8.8]

[Producers and consumers MAY support the following color features:](#)

- [N-Channel colors in image data, using the TIFF image format \[O8.19\]](#).

When non-sRGB color information is used, color value specifications are expressed using markup from the XPS Document schema.

Consumers are not required to handle all color spaces natively through every processing stage, but, rather, MAY convert data specified in a color space other than sRGB to sRGB at an early stage [O8.1]. Consumers that do not handle natively colors other than sRGB can experience reduced fidelity.

The requirements and recommendations of this subclause and its subclauses pertain equally to raster and vector color content.

15.1.1 sRGB Color Space

The XPS Document format supports colors in the sRGB color space for both vector and raster graphics.

15.1.2 scRGB Color Space

The XPS Document format supports colors in the scRGB color space for both vector and raster graphics. Such scRGB colors are typically used without an ICC profile. The encoding of scRGB as specified in IEC 61966-2-2 does not include a gamut boundary definition, therefore the scRGB gamut boundary to be used for scRGB colors is implementation-defined.

~~The XPS Document format supports colors in the scRGB color space for both vector and raster graphics. See §D for the scRGB gamut definition. [Lex2]~~

15.1.3 Gray Color Space

Gray colors for vector elements can be specified as sRGB or scRGB colors with the red, blue and green components set to the same value. [Alternatively, grayscale colors for vector elements can be specified as single channel monochrome with an associated ICC profile.](#) Gray colors for raster images can be specified using any image format.

15.1.4 CMYK Color Space

CMYK color is supported through the use of color management transformations from an ICC profile.

15.1.5 N-Channel Color Spaces

N-channel color is supported through the use of color management transformations from an ICC profile.

15.1.6 Named Color for Spot Colors and N-tone Images

Named colors are supported through the use of color management transformations from an ICC profile.

15.1.7 [Identifying Output-Ready Color Spaces Using ICC Profiles](#)~~Device Color Spaces~~

[An ICC profile, corresponding to a color space suitable for a particular device or device type, can be identified ~~at the page level~~ using a PrintTicket setting, as described in §9.1.9.](#)

~~Standard~~[XPS markup or profile embedding is used to identify such an ICC profile with the elements intended to be rendered in the native color space of the device or device-type.](#)

[If a consumer recognizes that a profile given in the syntax for a page element matches the ~~page level~~PrintTicket output-ready ICC profile and that the ~~page level~~PrintTicket output-ready ICC profile is suitable for the output device conditions, then the consumer SHOULD elect to treat the element colors as output-ready colors and not color-manage them, unless forced to do so for transparency effects or gradient blending \[S8.21\].](#)

[\[Note: Elements using named colors in a workflow in which the consumer is expected to use the encoded name of a named color to lookup a device-specific color value are not affected by the use of the PrintTicket setting described here. end note\]](#)

~~To specify colors in the native color space (usually CMYK or N-Channel) for a device, use the standard markup with an ICC profile that approximates the device color space. Include the same ICC device profile in the PageDeviceColorSpaceProfileURI PrintTicket setting. Determination of the color management result for such native color space colors is determined according to mechanisms described in §15.6, . It is the responsibility of the consumer to identify the ICC profile as one that correctly approximates the native colors of the device. When~~

~~1 colors in an XPS file are encoded in the native colors of the device, then the colors MUST NOT
2 be color managed according to the included profile unless forced to do so for transparency
3 effects or gradient blending. [M8.11].~~

4 **15.1.8 ICC Profiles**

5 XPS Documents MAY include associated ICC profile parts [O2.3]. XPS producers MAY include
6 ICC profiles embedded in any image format (according to the restrictions of the image file
7 format) with any color space [O8.15]. For color spaces other than sRGB and scRGB ~~[S8.1],~~
8 XPS producers MUST provide color management using associated or embedded (for raster
9 images) ICC profiles conforming to the requirements of the ICC Color Profile specification,
10 ICC.1:2001-04 [M8.12]. XPS producers MAY include ICC profiles for sRGB and scRGB color
11 spaces [O8.16]. XPS consumers MUST use associated and embedded ICC profiles, according to
12 the precedence order of §15.3.8 for raster images and according to §15.2 for vector content
13 [M8.53]. Optionally, XPS producers and consumers MAY provide color management using ICC
14 profiles conforming to the requirements of ISO 15076-1, "Image technology colour
15 management — Architecture, profile format, and data structure — Part 1: Based on
16 ICC.1:2004-10" [O8.9]. Producers ~~MUST SHOULD~~ restrict associated ICC profiles ~~[Lex3]~~ to
17 conform to the requirements of the older ICC Color Profile specification, ICC.1:2001-04, when
18 consumer support of the newer ISO version cannot be ascertained [M8.58S8.15]. If a Producer
19 includes an image with an embedded profile conforming to the requirements of ISO 15076-1,
20 then the Producer MUST associate an ICC profile conforming to the requirements of the older
21 ICC Color Profile specification, ICC.1:2001-04, to have precedence over such an embedded
22 profile, when consumer support of the newer ISO version cannot be ascertained [M8.59].

23 All ICC profiles used in XPS Documents MUST be one of the following [M8.13]:

- 24 • Input
- 25 • Output
- 26 • Monitor (RGB)
- 27 • ColorSpace Conversion
- 28 • Named Color

~~29 Consumers MUST support color profiles as specified in the ICC specification [M8.9].~~ Supported
30 profiles include Monochrome Input Profiles, Monochrome Display Profiles, Monochrome Output
31 Profiles, Three-component Matrix-based Input Profiles, and RGB Display Profiles. The set of
32 usable N-component LUT-based profiles is limited to 2-, 3-, 4-, 5-, 6-, 7-, or 8-color channels.
33 The set of usable Named Color profiles is limited to 1-, 2-, 3-, 4-, 5-, 6-, 7-, or 8-colors.

~~34 One tag is explicitly supported as specified in ICC Version 4.0.0 profiles: the colorant table for
35 named colors. A consumer incapable of supporting named colors SHOULD treat this tag as a
36 user-defined custom tag, and therefore ignore it. It SHOULD instead use the color tables as
37 provided in the profile to convert the specified colors to the Profile Connection Space (PCS)
38 [S8.14].~~

~~39 ICC profiles SHOULD be used when embedded in any image format with any color space,
40 except the scRGB color space, for which the gamut boundary described in G is assumed [S8.1].~~

~~41 [Note: Some consumers do not correctly apply ICC profiles to grayscale images. end note]~~ If
42 consistency of appearance of grayscale images is important, the producer SHOULD adjust the
43 gray tone response curve of such the images before adding it to the XPS Document [S8.2].

44 [Note: Some consumers do not correctly apply ICC profiles to grayscale images. end note]

- 1 An ICC profile MAY contain private tags [O8.17]. Implementations MAY act on private tags
 2 | [O8.18] and MUST ignore and preserve private tags that they do not understand [[M8.55](#)].

3 **15.2 Vector Color Syntax**

4 | [This subclause describes specific considerations for including vector colors in XPS Documents.](#)

5 Vector colors can 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 with the
 12 specified color.

13 *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"	Color="ContextColor /namedtintprofile.icc 1.0, 1.0"
RGB with alpha	Color="ContextColor ProfileURI AlphaFloat, Chan0Float, Chan1Float, Chan2Float"	Color="ContextColor /RGBprofile.icc 1.0, 1.0, 1.0, 1.0"
Grayscale with alpha	Color="ContextColor ProfileURI AlphaFloat, Chan0Float"	Color="ContextColor /grayprofile.icc 1.0, 1.0"

- 14 Real numbers specified for color channel values of scRGB and ContextColor colors MUST NOT
 15 use exponent forms of numbers [M8.14].

1 Profiles associated as described in Table 15–1, and determined to be usable, MUST be used by
2 consumers [M8.44].

3 It is the responsibility of consumers to determine profile usability. A profile associated as in
4 Table 15–1 SHOULD be considered unusable by a consumer if

- 5 • The profile is not compatible with the context color syntax
- 6 • The profile contains optional tags that ambiguate XPS use
- 7 • The profile contains invalid tag type signatures that invalidate XPS use [S8.18].

8 In general, the presence of one or more optional tags in an ICC profile does not make the
9 profile unusable. A consumer incapable of supporting a particular ICC profile tag that is optional
10 in both ICC and XPS MAY treat this tag as a user-defined custom tag, and therefore ignore it
11 [O8.13].

12 If no usable profile is present in a context color syntax, then a consumer MUST apply a color
13 rule based on the context color syntax [M8.45]. The context color value(s) are interpreted to be
14 the encoding of a particular color space as follows:

- 15 • Single component integer default for vector data MUST be grayscale with the sRGB non-
16 linearity, black point, and white point [M8.46].
- 17 • Three component integer default for vector data MUST be sRGB [M8.47].
- 18 • Three component float default for vector data MUST be scRGB [M8.48].
- 19 • The specific CMYK to be used as the four component data default for vector data MUST
20 be determined by the consumer [M8.49].
- 21 • N-Channel data with $N \leq 3$ and any named color data: the data of the first channel
22 MUST be interpreted independently as grayscale [M8.50]. Other channels are
23 disregarded.
- 24 • N-Channel with $N > 4$ MUST be treated as four component data using the four
25 component data default for vector data determined by the consumer [M8.51].

26 When no usable profile is present a consumer MAY choose to instantiate an error condition
27 [O8.14].

28 A producer MUST associate or embed a usable color profile if the color rules above do not
29 guarantee appropriate color interpretation for the vector color content [M8.52].

30 **15.2.1 sRGB Color Syntax**

31 The sRGB color syntax is the same as that used in HTML, with the red, green, and blue
32 channels represented by two hexadecimal digits. XPS Documents can specify an sRGB color
33 either with or without an alpha channel value, which is also expressed as two hexadecimal
34 digits.

35 The syntax is as follows (without alpha):

36 #RRGGBB

37 or (with alpha):

38 #AARRGGBB

39 When an sRGB color is specified without an alpha value, an alpha of "FF" is implied.

1 15.2.2 scRGB Color Syntax

- 2 The scRGB color syntax allows XPS Document producers to specify a color using the full scRGB
3 color space, which is much larger than the sRGB color space and can represent the entire range
4 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 implied. When
6 an scRGB color is specified with four numeric values, the first value is the alpha channel.
7 Although alpha values smaller than 0.0 and larger than 1.0 can be specified, they MUST be
8 clamped to the valid range from 0.0 to 1.0 before any further processing [M8.15].

9 **15.2.3 Grayscale syntax**

10 [XPS Document producers specify grayscale colors using the context color syntax, which allows
11 specification of a monochrome 'GRAY' ICC profile and an individual color channel value as a real
12 number. The context color MUST specify the matching number of channel float values \[M8.17\].](#)

13 [The syntax is as follows:](#)

14 `ContextColor ProfileURI AlphaFloat, Chan0Float`

15 [ProfileURI specifies a part containing the binary data of the color profile. The profile URI MUST
16 be added as a Required Resource relationship to the FixedPage part \[M2.10\].](#)

17 [Although alpha values smaller than 0.0 and larger than 1.0 can be specified, they MUST be
18 clamped to the valid range from 0.0 to 1.0 before any further processing \[M8.16\]. Channel float
19 values MUST also be clamped to the valid range from 0.0 to 1.0 before further processing.
20 Before the value is used as input for an ICC profile color transformation, it MUST be linearly
21 scaled \[with specified rounding/clipping\] to the range from 0 to 255 or from 0 to 65535,
22 depending on whether the profile uses 8-bit or 16-bit input tables \[M8.31\].](#)

23 ~~15.2.3~~ **15.2.4 CMYK Color Syntax**

24 XPS Document producers specify CMYK colors using the context color syntax, which allows
25 specification of an ICC profile and the individual color channel values as real numbers.

26 The syntax is as follows:

27 `ContextColor ProfileURI AlphaFloat, Chan0Float, Chan1Float, Chan2Float,
28 Chan3Float`

29 [ProfileURI specifies a part containing the binary data of the color profile. The profile URI MUST
30 be added as a Required Resource relationship to the FixedPage part \[M2.10\].](#)

31 Although alpha values smaller than 0.0 and larger than 1.0 can be specified, they MUST be
32 clamped to the valid range from 0.0 to 1.0 before any further processing [M8.16]. Channel float
33 values MUST also be clamped to the valid range from 0.0 to 1.0 before further processing.
34 Before the value is used as input for an ICC profile color transformation, it MUST be linearly
35 scaled (with specified rounding/clipping) to the range from 0 to 255 or from 0 to 65535,
36 depending on whether the profile uses 8-bit or 16-bit input tables [M8.31].

~~15.2.4~~ 15.2.5 N-Channel Color Syntax

XPS Document producers specify N-channel colors using the context color syntax, which allows specification of an ICC profile and the individual color channel values as real numbers. The syntax is expressed as follows:

```
ContextColor ProfileURI AlphaFloat, Chan0Float, ..., ChanN-1Float
```

`ProfileURI` specifies a part containing the binary data of the color profile. The profile URI MUST be added as a Required Resource relationship to the FixedPage part [M2.10].

The profile can be a 2-, 3-, 4-, 5-, 6-, 7- or 8-channel N-clr profile (indicated by using one of the {'2CLR' ... '8CLR'} values in the profile header color space signature field). The context color MUST specify the matching number of channel float values [M8.17].

Although alpha values smaller than 0.0 and larger than 1.0 can be specified, they MUST be clamped to the valid range from 0.0 to 1.0 before any further processing [M8.18]. Channel float values MUST also be clamped to the valid range from 0.0 to 1.0 before further processing. Before the value is used as input for an ICC profile color transformation, it MUST be linearly scaled (with specified rounding/clipping) 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 [M8.31] ~~before input to the profile.~~

[*Example:* For duotone 2-clr content (with color-managed color mixing) the syntax is:

~~[*Example:*~~

```
ContextColor ProfileURI AlphaFloat, Chan0Float, Chan1Float
```

end example]

For 1-channel color, i.e., monochrome, use a monochrome input ~~(or output)~~ profile ([profile header color space signature is 'GRAY'](#)). The profile MUST include the ICC-optional AToB1Tag (relative colorimetric intent) if the single color is chromatic (not neutral) [M8.32]. [*Example:*

```
ContextColor ProfileURI AlphaFloat, Chan0Float
```

end example]

If the XPS system environment allows the use of ICC ISO 15076-1 profiles, the optional `colorantTableTag` SHOULD be included in such ISO 15076-1 profiles to indicate the names and corresponding PCS values of the individual [N-color](#) colorants [S8.16]. (See §15.1.8 for the appropriate use of ISO 15076-1 profiles.)

15.2.6 Named Color Syntax

A *named color* is an industry-defined color specification that identifies a particular color in a well-defined color [system schema](#), usually for the purpose of printing. [There are currently several named color schema systems](#). In XPS, a named color is expressed as a combination of an ink name and transform information stored in an ICC profile and a tint level (percentage ink dilution) given in the XPS context color syntax. The XPS context color syntax allows specification of one or more named color tint values and association of an ICC profile. The syntax is expressed as follows:

```
ContextColor ProfileURI AlphaFloat,Tint0Float,...,TintN-1Float
```

Two ICC profile approaches are available for named colors, one using ICC monochrome profiles that each include a tint LUT for a single named color, and the other using ICC Named Color type profiles that each [can](#) include 100% color values for 1, 2, 3, 4, 5, 6, 7, or 8 named colors. In both cases, the XPS context color syntax MUST specify the matching number of tint float values [\[M8.55M8.17\]](#).

A named color with an associated tint LUT MUST be implemented in an XPS Document using an associated ICC monochrome profile [\[M8.33\]](#). In this case, the ICC profile MUST contain the tint LUT for a single named color [\[M8.34\]](#). The ICC profile MUST be an ICC monochrome input or output profile [\[M8.35\]](#). [The profile header color space signature MUST be 'GRAY' \[M8.37\]. The profile MUST include an AtoB1Tag \(relative colorimetric rendering intent\), mapping the named color tint values to valid PCS values \[M8.19\], in addition to the ICC-required grayTRCTag \(not used for XPS named colors\).](#) The ASCII prefix-root-suffix name of the named color MUST be encoded into the profileDescriptionTag of the ICC profile [\[M8.36\]](#); [so that a consumer MAY use the profile to obtain the encoded name of the named color \[O8.20\]. A consumer MAY use the encoded name of a named color to lookup a device-specific color value for the named color \[O8.21\].](#) ~~The profile header color space signature MUST be 'GRAY' [M8.37]. The profile MUST include an AtoB1Tag (relative colorimetric rendering intent), mapping the named color tint values to valid PCS values [M8.19], in addition to the ICC-required grayTRCTag (not used for XPS named colors).~~

The context color syntax for referencing a [single](#) named color is as follows:

```
ContextColor ProfileURI AlphaFloat,TintFloat
```

A single named color MAY be implemented in an XPS Document using an associated ICC Named Color type profile [\[O8.11\]](#). Two or more named colors implemented in an XPS Document using a single associated profile MUST use an ICC Named Color type profile [\[M8.38\]](#). An ICC Named Color type profile MUST contain the namedColor2Tag including the ASCII prefix-root-suffix name for each named color [\[M8.39\]](#); [so that a consumer MAY use the profile to obtain the encoded name of the named color \[O8.22\].](#) The namedColor2Tag MUST be populated with the ICC PCS color value for each named color [\[M8.40\]](#) and MAY be populated with specific device color values for each named color [\[O8.12\]](#). A named color duotone, tritone, etc., can be implemented in this way. [A consumer MAY use the encoded name of a named color to lookup a device-specific color value for the named color \[O8.23\].](#)

[\[Example:](#) For duotone named color content (with NO color managed color mixing) the syntax is: ~~[Example:~~

1 ContextColor ProfileURI AlphaFloat,Tint0Float,Tint1Float

2 *end example]*

3 ProfileURI specifies a part containing the binary data of the color profile. The profile URI MUST
4 be added as a Required Resource relationship to the FixedPage part [M2.10]. AlphaFloat
5 specifies the alpha to be applied to the named color. TintFloat specifies how diluted with
6 | respect to the color [systemschema](#)'s white color point the named color is, with 1.0 being the
7 pure named color and 0.0 being fully diluted.

8 Although alpha values smaller than 0.0 and larger than 1.0 can be specified, they MUST be
9 clamped to the valid range from 0.0 to 1.0 before any further processing [M8.20]. The tint float
10 value MUST also be clamped to the valid range from 0.0 to 1.0 before further processing.
11 | **IfBefore** the value is used as input for an ICC profile color transformation, it MUST be linearly
12 scaled (with specified rounding/clipping) to the range from 0 to 255 or from 0 to 65535,
13 | depending on whether the profile uses 8-bit or 16-bit input tables ~~before input to the profile~~
14 [M8.31].

15 Consumers MAY use the ASCII name in the ICC profile or MAY compute a color approximation
16 using the specified color value in the ICC profile. When a named color is used in a gradient
17 brush or with transparency, the results of these two methods MAY differ significantly [O8.3].

18 **15.3 Colors in Raster Images**

19 This subclause describes specific considerations for including color raster images in XPS
20 Documents.

21 **15.3.1 sRGB Raster Images**

22 XPS Documents support sRGB raster images in the following formats:

- 23 • JPEG
- 24 • PNG
- 25 • TIFF
- 26 • Windows Media Photo

27 The following Windows Media Photo pixel formats are supported:

- 28 • WICPixelFormat24bppRGB
- 29 • WICPixelFormat24bppBGR
- 30 • WICPixelFormat32bppBGR
- 31 • WICPixelFormat32bppBGRA
- 32 • WICPixelFormat32bppPBGRA
- 33 • WICPixelFormat48bppRGB
- 34 • WICPixelFormat64bppRGBA
- 35 • WICPixelFormat64bppPRGBA

36 Pixel formats WICPixelFormat32bppPBGRA and WICPixelFormat64bppPRGBA are pre-multiplied
37 alpha formats. See §18.4.1 for details.

1 The following Windows Media Photo packed pixel formats are supported:

- 2 • WICPixelFormat16bppBGR555
- 3 • WICPixelFormat16bppBGR565
- 4 • WICPixelFormat32bppBGR101010

5 See §9.1.5 for more details.

6 **15.3.2 scRGB Raster Images**

7 XPS Documents support scRGB raster images only in the Windows Media Photo image format.
8 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 §18.4.1
23 for details.

24 **15.3.3 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 **15.3.4 CMYK Raster Images**

4 CMYK images are stored in TIFF or Windows Media Photo format.

5 **15.3.4.1 TIFF CMYK Raster Images**

6 CMYK TIFF image tags are described in §9.1.5.3.

7 ICC profiles can be associated with CMYK raster images by using an ICC profile embedded in
8 the TIFF file (tag 34675) or associated using the mechanism described in §15.3.8.

9 **15.3.4.2 Windows Media Photo CMYK Raster Images**

10 The Windows Media Photo CMYK format is described in the Windows Media Photo specification.
11 The following formats are supported:

- 12 • WICPixelFormat32bppCMYK
- 13 • WICPixelFormat40bppCMYKAlpha
- 14 • WICPixelFormat64bppCMYK
- 15 • WICPixelFormat80bppCMYKAlpha

16 **15.3.4.3 JPEG CMYK Raster Images**

17 Support for JPEG CMYK images varies by implementation and SHOULD NOT be used in XPS
18 Documents [S2.7]. See §9.1.5.1 for more details.

19 **15.3.5 N-channel Raster Images**

20 N-channel images are stored in the Windows Media Photo image file format using an ICC
21 profile. The following formats are supported:

- 22 • WICPixelFormat24bpp3Channels, WICPixelFormat48bpp3Channels
- 23 • WICPixelFormat32bpp4Channels, WICPixelFormat64bpp4Channels
- 24 • WICPixelFormat40bpp5Channels, WICPixelFormat80bpp5Channels
- 25 • WICPixelFormat48bpp6Channels, WICPixelFormat96bpp6Channels
- 26 • WICPixelFormat56bpp7Channels, WICPixelFormat112bpp7Channels
- 27 • WICPixelFormat64bpp8Channels, WICPixelFormat128bpp8Channels
- 28 • WICPixelFormat32bpp3ChannelsAlpha, WICPixelFormat64bpp3ChannelsAlpha
- 29 • WICPixelFormat40bpp4ChannelsAlpha, WICPixelFormat80bpp4ChannelsAlpha
- 30 • WICPixelFormat48bpp5ChannelsAlpha, WICPixelFormat96bpp5ChannelsAlpha
- 31 • WICPixelFormat56bpp6ChannelsAlpha, WICPixelFormat112bpp6ChannelsAlpha
- 32 • WICPixelFormat64bpp7ChannelsAlpha, WICPixelFormat128bpp7ChannelsAlpha
- 33 • WICPixelFormat72bpp8ChannelsAlpha, WICPixelFormat144bpp8ChannelsAlpha

34 [The profile can be a 2-, 3-, 4-, 5-, 6-, 7- or 8-channel N-clr profile \(indicated by using one of](#)
35 [the {'2CLR' ... '8CLR'} values in the profile header color space signature field\).](#)

1 [For 1-channel color, i.e., monochrome, use a monochrome input \(or output\) profile. The profile](#)
2 [MUST include the ICC-optional AToB1Tag \(relative colorimetric intent\) if the single color is](#)
3 [chromatic \(not neutral\) \[M8.32\].](#)

4 [If the XPS system environment allows the use of ICC ISO 15076-1 profiles, the optional](#)
5 [colorantTableTag SHOULD be included in such ISO 15076-1 profiles to indicate the names and](#)
6 [corresponding PCS values of the individual N-color colorants \[S8.16\]. \(See §15.1.8 for the](#)
7 [appropriate use of ISO 15076-1 profiles.\)](#)

8 **15.3.6 Named Color Raster Images**

9 [A *named color* is an industry-defined color specification that identifies a particular color in a](#)
10 [well-defined color system *schema*, usually for the purpose of printing. There are currently](#)
11 [several named color system *schemas*. In XPS, a named color is expressed as a combination of](#)
12 [an ink name and transform information stored in an ICC profile.](#)

13 Named color (~~N-tone~~) raster images are stored in the Windows Media Photo image file format
14 using an ICC profile that maps the tint channel combinations to valid PCS values. See §15.3.5
15 for pixel format definitions.

16 Consumers unaware of named colors can then compute color approximations using the PCS
17 values computed from the profile.

18 [Two ICC profile approaches are available for named colors, one using ICC monochrome profiles](#)
19 [that each include a tint LUT for a single named color, and the other using ICC Named Color type](#)
20 [profiles that each can include 100% color values for 1, 2, 3, 4, 5, 6, 7, or 8 named colors.](#)

21 [A monochrome named color raster image can have a tint LUT encoded in an ICC monochrome](#)
22 [input or output profile. The profile header color space signature is 'GRAY'. The profile includes](#)
23 [an AtoB1Tag \(relative colorimetric rendering intent\), for the tint LUT mapping the named color](#)
24 [tint values to valid PCS values. The ASCII prefix-root-suffix name of the named color is](#)
25 [encoded into the profileDescriptionTag of the ICC profile. ~~The profile header color space~~](#)
26 [signature is 'GRAY'. ~~The profile includes an AtoB1Tag \(relative colorimetric rendering intent\), for~~](#)
27 [the tint LUT mapping the named color tint values to valid PCS values; so that a consumer MAY](#)
28 [use the profile to obtain the encoded name of the named color \[O8.22\]. A consumer MAY use](#)
29 [the encoded name of a named color to lookup a device-specific color value for the named color](#)
30 [\[O8.23\].](#)

31 [A multi-tone named color raster image can have an ICC Named Color type profile. An ICC](#)
32 [Named Color type profile MUST contain the namedColor2Tag including the ASCII prefix-root-](#)
33 [suffix name for each named color \[M8.39\] so that a consumer MAY use the profile to obtain the](#)
34 [encoded name of the named color \[O8.22\]. The namedColor2Tag MUST be populated with the](#)
35 [ICC PCS color value for each named color \[M8.40\] and MAY be populated with specific device](#)
36 [color values for each named color \[O8.12\]. A named color duotone, tritone, etc., can be](#)
37 [implemented in this way. A consumer MAY use the encoded name of a named color to lookup a](#)
38 [device-specific color value for the named color \[O8.23\].](#)

39 [Consumers MAY use the ASCII name in the ICC profile or MAY compute a color approximation](#)
40 [using a specified color value in the ICC profile; the results of these two methods MAY differ](#)
41 [significantly \[O8.25\].](#)

42 **15.3.7 Device Color Raster Images**

43 Device color (N-channel) raster images are stored in the Windows Media Photo image file
44 format in the same manner as a named color raster image. See §15.1.7 for more details. RGB

1 and CMYK raster images can also be stored in the TIFF image file format. JPEG CMYK images
2 SHOULD NOT be used [S2.7].

3 **15.3.8 Images and Color Profile Association**

4 Images can depend on color profiles using one of two methods:

- 5 • Associated: Color profile contained in a separate part associated with the image, ~~using~~
6 ~~the following markup:~~

7 ~~• `<ImageBrush ImageSource="{ColorConvertedBitmap image.tif profile.icc}" ... />`~~

8 ~~The profile URI MUST be added as a Required Resource relationship to the FixedPage part~~
9 ~~[M2.10].~~

- 10 • Embedded: Color profile embedded in an image using the image format specific mechanism.

11 When associating a profile with an image the syntax for the ImageSource attribute is as
12 follows:

13 `{ColorConvertedBitmap ImageSourceURI ProfileURI}`

14 ImageSourceURI Specifies the URI of an image resource. The image URI MUST be added as a
15 Required Resource relationship to the FixedPage part [M2.10].

16 ProfileURI specifies a part containing the binary data of the color profile. The profile URI MUST
17 be added as a Required Resource relationship to the FixedPage part [M2.10].

18 [Example:

19 `<ImageBrush ImageSource="{ColorConvertedBitmap ../Resources/Images/image.tif`
20 `../Metadata/profile.icc}" ... />`

21 end example]

22 It is the responsibility of consumers to determine the usability of embedded or associated
23 profiles. A profile associated or embedded with an image SHOULD be considered unusable by a
24 consumer if

- 25 • The profile is not compatible with the pixel format of the image
- 26 • The profile contains optional tags that ambiguate XPS use
- 27 • The profile contains invalid tag type signatures that invalidate XPS use [S8.17].

28 In general, the presence of one or more optional tags in an ICC profile does not make the
29 profile unusable. A consumer incapable of supporting a particular ICC profile tag that is optional
30 in both ICC and XPS MAY treat this tag as a user-defined custom tag, and therefore ignore it
31 [O8.13].

32 If present and usable, an associated profile MUST be used by consumers [M8.41]. A usable
33 associated color profile overrides an embedded color profile and is processed instead of any
34 embedded color profile.

35 If present and usable, a color profile embedded in an image file MUST be used by consumers
36 when no usable associated profile is present with the image [M8.42].

37 If no usable profile is present with an image, then a consumer MUST apply a color rule based
38 on the pixel format. Each pixel format is interpreted to be the encoding of a particular color
39 space as shown in Table 15-1 ~~Table 15-4~~ [M8.30].

1 When no usable profile is present a consumer MAY choose to instantiate an error condition
2 [O8.14].

3 A producer MUST associate or embed a usable color profile if the color rules of Table 15-1
4 ~~Table 15-4~~ do not guarantee appropriate color interpretation for an image [M8.43].

5 *Table 15-2. Color Space Pixel Format Defaults*

Pixel Formats	Color Space
Integer 1-Channel	Grayscale using non-linearity, black point, and white point from sRGB
Fixed Point 1-Channel	
Half-Float 1-Channel	
Floating Point 1-Channel	
Integer 3-Channel	sRGB
Floating Point 3-Channel	scRGB
Half-Float 3-Channel	
Fixed-Point 3-Channel	
Integer 4-Channel	CMYK
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)	

6 The sRGB non-linearity, white point, and black point can be applied to single channel grayscale
7 data using the equations of IEC 61966-2-1, §5.2, by setting R=G=B equal to the grayscale
8 value.

9 The specific CMYK to be used as the four-component raster data default, and the N-Channel
10 (N=>4) default, is implementation-defined. In the absence of specific requirements the use of
11 CGATS/SWOP TR003 2007 CMYK is ~~recommended~~ **RECOMMENDED** [S8.19 ~~O8.10~~].
12 Alternatively, a consumer MAY choose to instantiate an error condition [O8.14 ~~O8.10~~ ~~O8.5~~].

13 [*Note: A profile for CGATS/SWOP TR003 2007 CMYK is available from the ICC Profile Registry,
14 specifically SWOP2006_Coated3v2.icc. end note*]

15 **15.4 Registration Marks for Color Separations**

16 [Producers MAY elect to generate content that provides registration marks for consumers that
17 perform color separation \[O8.5\].](#)

18 [The named color syntax can be used for registration marks that are intended to be rendered on
19 every separation. A document registration named color can be identified at the document level
20 using a PrintTicket setting \(see §9.1.9\).](#)

21 [A document registration named color identified in a PrintTicket MAY occur in an XPS Document
22 using the single named color and monochrome profile with tint LUT syntax \(see §15.2.6\)](#)

~~[O8.26Ox.x]. The name of the document registration named color is given in the profile's profileDescriptionTag according to §15.2.6. Such a document registration named color SHOULD be unique for that use in the XPS Document instance[S8.22].~~

~~For consumers that do not perform separation, the document registration named color ICC profile is used to compute output colorant values corresponding to the document registration named color. For consumers that do perform separation, the occurrence of the document registration named color in a color syntax is *only* an indicator that the tint level supplied in the syntax SHOULD be used when drawing the registration marking in each colorant separation [S8.7]. Producers SHOULD create the profile for the document registration named color in such a way that it does not lay down excessive ink when printed on a device that does not perform separation [S8.8].~~

~~Consumers MAY perform color separation, if desired [O8.5].~~

~~A named color used for markings that are intended to be rendered on every layer of the separation can be specified with the DocumentImpositionColor PrintTicket setting.~~

~~The color name specified by the DocumentImpositionColor PrintTicket setting MUST be matched only to profiles containing exactly one non-zero-length colorant name in the profile's colorantTable [M8.22]. 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 [M8.23]. 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 [M8.24].~~

~~The imposition named color is used *only* to compute XYZ values for consumers that do not perform separation. For consumers that do perform separation, it is an indicator that the tint level supplied SHOULD be used for all device colorants [S8.7]. 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 and uses the profile to compute XYZ values instead [S8.8].~~

15.5 Alpha and Gradient Blending

~~For consumers that handle colors other than sRGB, it is necessary to understand how they can be blended to create gradient or transparency effects. [A page-level PrintTicket setting can be used to specify the blending color space that SHOULD be used for blending gradients and transparencies \[S8.9\] \(see §9.1.9\).](#)~~The PrintTicket specifies the color space that SHOULD be used for blending gradients and transparencies in the PageBlendColorSpace setting [S8.9]. These settings apply to the page level.~~~~

~~If a consumer understands the [blending color space PageBlendColorSpace](#) PrintTicket setting, it SHOULD convert all color to the specified blending color space before performing a blend operation [S8.9]. For gradients, the specified blending color space 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 [blending color space PageBlendColorSpace](#) PrintTicket setting, the color interpolation mode of the gradient brush MUST be used instead [M8.25].~~

~~Consumers MUST support [alpha and gradient blending in sRGB](#) [M8.1], but they MAY support alpha and gradient blending with other color spaces such as scRGB or CMYK [O8.6]. The behavior of documents using non-sRGB alpha and gradient blending is implementation-~~

- 1 [specific](#)~~defined~~. Consumers that encounter any document using non-sRGB colors MAY process
- 2 those colors using conversion to the simpler sRGB color space, resulting in deviations,
- 3 especially for alpha blending [O8.6].

15.6 ~~Color Rendering Intent~~ ~~PrintTicket-Color Settings~~

This subclause summarizes the color-related ~~PrintTicket~~ settings. For more information, refer to the ~~Print Schema~~.

~~Table 15-5. 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	profileUri-properties	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 <code>PageDeviceColorSpaceProfileURI</code> 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 <code>Uri</code> property of the option specifies an ICC profile defining the color space that SHOULD be used for blending [S11.13]. The <code>Uri</code> is an absolute part name relative to the package root. The profile MUST be an output profile (containing <code>AToB0Tag</code> , <code>BToA0Tag</code> , <code>AToB1Tag</code> , <code>BToA1Tag</code> , <code>AToB2Tag</code> , and <code>BToA2Tag</code>), otherwise it MUST be ignored [M8.28]. Elements using the profile specified by <code>PageBlendColorSpace</code> 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 profiles compliant with ICC.1:2001-04. This value SHOULD be ignored for elements using an ICC profile [S8.13].
DocumentImpositionColor	colorName-properties	Elements using the named color identified by the colorName-properties MUST appear on all color separations [M8.29]. See §15.4 and 15.2.5 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.

1 [ICC profiles contain multiple color transformation options, identified in the ICC Color Profiles](#)
 2 [specification as ICC rendering intents. For color elements that are to be color managed, a page](#)
 3 [level default color rendering intent, can be identified using a PrintTicket \(see §9.1.9\).](#)

4 [In the absence of such information, in a typical case, with ICC profiles conforming to the ICC](#)
 5 [Color Profile specification, ICC.1:2001-04 \[M8.12\], a consumer SHOULD apply the defaults](#)
 6 [shown in Table 15.4 \[rcj4\]\[S8.20\].](#)

7 [Table 15-3. Recommended ICC rendering intent usage](#)

Color type	Object type	ICC Source Rendering Intent	ICC Destination Rendering Intent
sRGB	Raster image	Perceptual	Perceptual
sRGB	Vector	Relative Colorimetric	Relative Colorimetric*
scRGB	Raster image	Perceptual	Perceptual
scRGB	Vector	Relative Colorimetric	Relative Colorimetric*
RGB color space	Raster image	Perceptual	Perceptual
RGB color space	Vector	Relative Colorimetric	Relative Colorimetric*
CMYK, Gray color space	Raster image	Relative Colorimetric	Perceptual
CMYK, Gray color space	Vector	Relative Colorimetric	Relative Colorimetric*
Named color,	Any	Relative	Relative

N-Channel

ColorimetricColorimetric

- 1 [*In the optional case, with ICC profiles conforming to the requirements of ISO 15076-1, based](#)
- 2 [on ICC.1:2004-10 \[08.9\], the Saturation Rendering Intent components of the profiles should](#)
- 3 [be optimized for business graphics and may provide preferred results.](#)

16. Document Structure and Interactivity

Some consumers support enhanced interactive functionality through features such as text selection, navigation, and hyperlinking. Others, such as screen readers, provide enhanced accessibility. These features rely on structural information beyond what can be inferred from the page markup. Producers can author this information explicitly.

The methods for adding document structure described here are OPTIONAL [O9.1]. Consumers MAY ignore any authored document structure or hyperlinks [O9.1], particularly where they are not relevant (such as in the case of printers). Recommended consumer behavior in the absence of document structure information is also described.

Document structure is defined with markup in the FixedPage, FixedDocument, DocumentStructure, and StoryFragments parts.

16.1 Document Structure Markup

Document structure markup consists of two structural concepts. The first is the *document outline*, which contains a structured list of indices into the XPS Document, similar to a table of contents. The second is the *document content*, which identifies blocks of individually readable content. Each of these blocks is called a *story*.

A story can extend across multiple pages, and several stories can share a single page. A story can include the entire contents of an XPS Document, or it can include only an individual block of readable content, such as a single newspaper article. Like a newspaper article, the story can appear in blocks throughout the XPS Document. [*Example*: The first part could appear on page 1 and the second part on page 5. *end example*] Since a story can span multiple pages, the document content identifies which FixedPage parts contain fragments of a particular story.

A *story fragment* is the portion of a story that appears within a single fixed page. The story fragment contains the structural markup for all text and images related to a particular story on a particular page. When a producer specifies the document structure, every FixedPage part has a corresponding StoryFragments part that contains all of the story fragments for that page.

Each story fragment contains content structure information. *Content structure* is the set of markup elements that allow expression of well-understood semantic blocks, such as paragraphs, tables, lists, and figures. Content structure markup enables features such as paragraph and table selection, screen reading, and rich-format copying.

Producers MAY provide either the document outline or the document content, or both; consumers MAY ignore either or both [O9.2].

16.1.1 DocumentStructure Part

The fundamental building block of document structure markup is the named element. A *named element* refers to an element in the fixed page markup with a specified Name attribute. Every meaningful element in the fixed page markup SHOULD specify a Name attribute in order for the document structure markup to refer to it [S9.1].

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

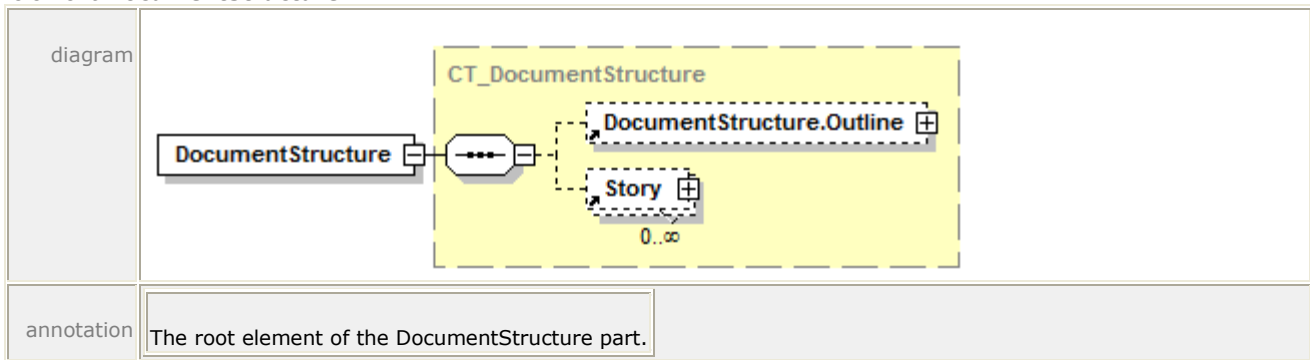
1 refers to. When referring to a <Canvas> element, producers SHOULD consider all descendant
 2 elements to be referenced in markup order [S9.3]. Consumers MAY choose to interpret these
 3 scenarios as duplicate document content [O9.3].

4 Children of <VisualBrush> elements SHOULD NOT be referenced by document structure
 5 markup [S9.30].

6 Because each named element in a FixedPage part that is intended as an addressable location is
 7 specified in the <PageContent.LinkTargets> element in the FixedDocument part, consumers
 8 MAY first attempt to locate named elements directly from the FixedDocument part [O9.4].

9 **16.1.1.1 <DocumentStructure> Element**

10 element **DocumentStructure**



11 The <DocumentStructure> element is the root element of the DocumentStructure part. [That](#)
 12 [element](#) ~~it~~ MAY contain a single <DocumentStructure.Outline> element and zero or more
 13 <Story> elements [[O9.14](#)~~M2.72~~].

14 *Example 16–1. Document structure markup*

```

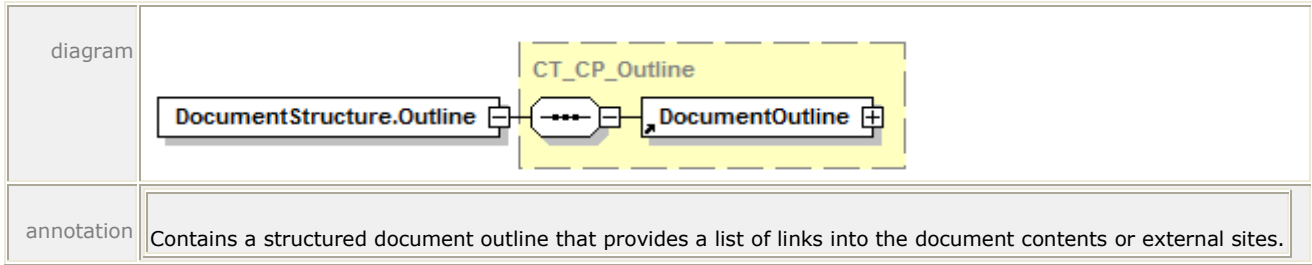
15 <DocumentStructure
16   xmlns="http://schemas.microsoft.com/xps/2005/06/documentstructure">
17   <DocumentStructure.Outline>
18     ...
19   </DocumentStructure.Outline>
20   <Story>
21     ...
22   </Story>
23   <Story>
24     ...
25   </Story>
26 </DocumentStructure>

```

27 *end example]*

1 **16.1.1.2 <DocumentStructure.Outline> Element**

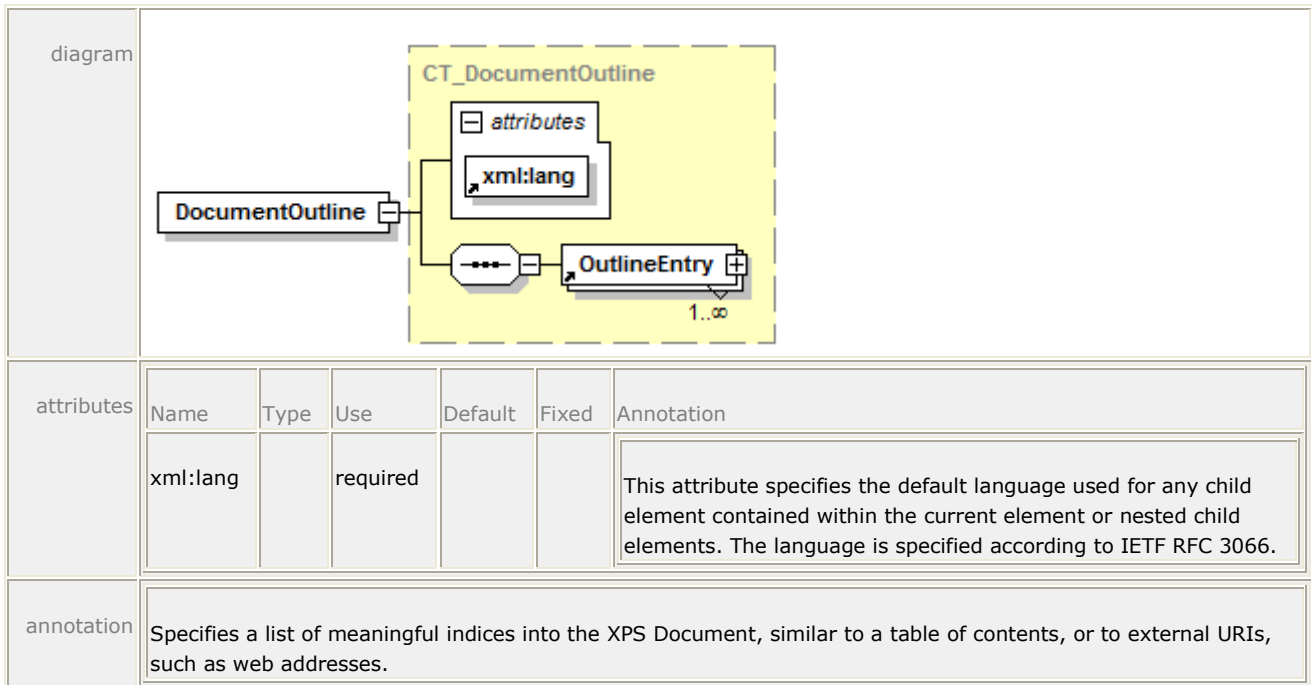
2 element **DocumentStructure.Outline**



3 The <DocumentStructure.Outline> element is the root element of the document outline. The
 4 <DocumentStructure.Outline> element contains only a single <DocumentOutline> element.

5 **16.1.1.3 <DocumentOutline> Element**

6 element **DocumentOutline**



7 The <DocumentOutline> element lets producers specify an organizational hierarchy in the form
 8 of a list of URIs to locations in the fixed page markup or to external addresses, similar to a
 9 table of contents or a set of bookmarks. The <DocumentOutline> element contains only
 10 <OutlineEntry> elements.

11 The xml:lang attribute specifies the default language used by the Description attribute of the child
 12 <OutlineEntry> element.

13 Consumers can use the document outline to implement such features as a table of contents or
 14 a navigation pane.

15 **16.1.1.4 <OutlineEntry> Element**

16 element **OutlineEntry**

<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>OutlineLevel</td> <td>ST_IntGEOne</td> <td>optional</td> <td>1</td> <td></td> <td>A description of the level where the outline entry exists in the hierarchy. A value of 1 is the root.</td> </tr> <tr> <td>OutlineTarget</td> <td>xs:anyURI</td> <td>required</td> <td></td> <td></td> <td>The URI to which the outline entry is linked. This can 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.</td> </tr> <tr> <td>Description</td> <td>xs:string</td> <td>required</td> <td></td> <td></td> <td>The friendly text associated with this outline entry.</td> </tr> <tr> <td>xml:lang</td> <td></td> <td>optional</td> <td></td> <td></td> <td>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.</td> </tr> </tbody> </table>	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 can 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.					
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 can 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.																															
<p>annotation</p>	<p>Represents an index to a specific location in the document.</p>																																			

1 Each <OutlineEntry> element represents an index to a specific location in the document or a
 2 specific location external to the document. Consumers can use the document outline
 3 information to support interactive functionality.

4 *Example 16–2. Document outline markup*

5 A viewing consumer can create a navigation pane that uses the Unicode value of the Description
 6 attribute of each <OutlineEntry> element. The corresponding location is specified by the
 7 OutlineTarget attribute, which are specified in a manner identical to hyperlinks. The OutlineLevel
 8 attribute allows consumers to indent entries in the navigation pane.

```

9     <DocumentStructure
10         xmlns="http://schemas.microsoft.com/xps/2005/06/documentstructure">
11         <DocumentStructure.Outline>
12             <DocumentOutline>
13                 <OutlineEntry
14                     OutlineLevel="1"
15                     Description="1. Documents"
    
```

```

1         OutlineTarget="../FixedDoc.fdoc#Documents_1" />
2     <OutlineEntry
3         OutlineLevel="2"
4         Description="1.1. Paragraphs"
5         OutlineTarget="../FixedDoc.fdoc#Paragraphs_1_1" />
6     </DocumentOutline>
7 </DocumentStructure.Outline>
8 </DocumentStructure>

```

9 A consumer might display this information as follows, with the first entry linked to `Documents_1`
10 and the second entry linked to `Paragraphs_1_1`.

```

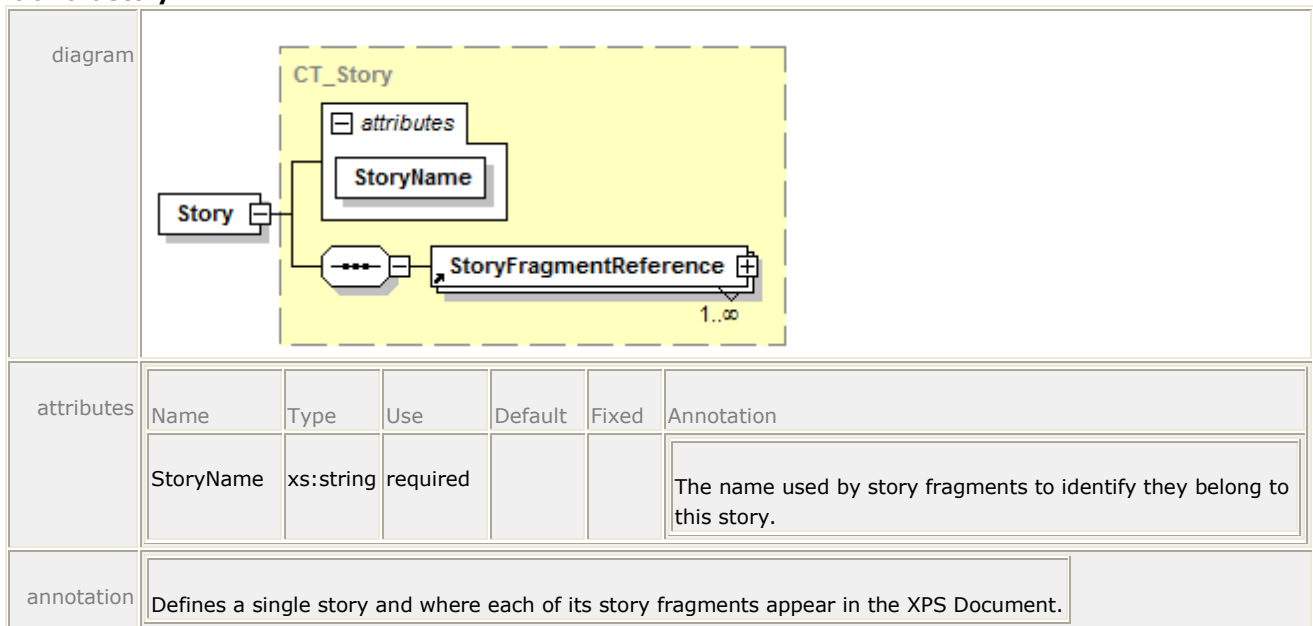
11     1. Documents
12         1.1. Paragraphs

```

13 *end example]*

14 16.1.1.5 <Story> Element

15 element **Story**



16 The <Story> element is the root for a single story and orders all of the story fragments
17 containing content structure information such as sections, paragraphs, and tables. Each story
18 has a unique name that is used to correlate the content structure for each page to that story.
19 The <Story> element contains one or more <StoryFragmentReference> elements.

1 **16.1.1.6 <StoryFragmentReference> Element**

2 element **StoryFragmentReference**

diagram						
attributes	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	ST_IntGEOne	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.					

3 The <StoryFragmentReference> element identifies the page with a relationship to the
 4 StoryFragments part in which the single story fragment is defined. By identifying where in the
 5 XPS Document each story fragment appears, consumers can easily access only the pages that
 6 contain a particular story.

7 Each page that contains a story fragment is identified by number. This number refers to the *n*th
 8 page of the XPS Document referenced within the fixed document sequence and fixed document
 9 markup, starting at the fixed payload root. This value is identified in the Page attribute. The
 10 StoryFragments part containing the corresponding content structure is referenced by retrieving
 11 the part associated via relationship from the indicated page. This allows consumers to access
 12 only the pages of the document that contain the story of interest. It is also possible for a single
 13 story to return to a page containing a different fragment of the same story.

14 The FragmentName attribute MUST be unique within the scope of the story [[M9.11](#)~~M2.72~~].

1 *Example 16-3. Simple multi-story document*

2 The following markup describes a four-page document containing one story that covers the first
3 one and one-half pages and then continues on page 4. It is interrupted by a second story that
4 begins in the middle of page 2 and concludes on page 3.

```
5     <DocumentStructure
6         xmlns="http://schemas.microsoft.com/xps/2005/06/documentstructure">
7         <Story StoryName="Story1">
8             <StoryFragmentReference Page="1"/>
9             <StoryFragmentReference Page="2"/>
10            <StoryFragmentReference Page="4"/>
11        </Story>
12        <Story StoryName="Story2">
13            <StoryFragmentReference Page="2"/>
14            <StoryFragmentReference Page="3"/>
15        </Story>
16    </DocumentStructure>
```

17 *end example]*

18 *Example 16-4. Story flowing back and forth across a page boundary*

19 The following markup describes a page containing two tables, arranged side-by-side, each of
20 which continues to the following page. In this case, the fragment is split and a fragment name
21 is specified. `FragmentA` refers to the content leading up to the middle of the first (left) table and
22 `FragmentB` is the continuation of this table on the following page. The flow then returns to the
23 second (right) table on page 1 (`FragmentC`) before continuing with the rest of the story in
24 `FragmentD`.

```
25     <DocumentStructure
26         xmlns="http://schemas.microsoft.com/xps/2005/06/documentstructure">
27         <Story StoryName="Story_1">
28             <StoryFragmentReference FragmentName="FragmentA" Page="1"/>
29             <StoryFragmentReference FragmentName="FragmentB" Page="2"/>
30             <StoryFragmentReference FragmentName="FragmentC" Page="1"/>
31             <StoryFragmentReference FragmentName="FragmentD" Page="2"/>
32         </Story>
33     </DocumentStructure>
```

34 *end example]*

35 **16.1.2 StoryFragments Part**

36 The StoryFragments part contains content structure markup (describing such things as tables
37 and paragraphs) for each story fragment that appears on the page. The content structure is
38 expressed by tags that ultimately wrap `<NamedElement>` references that point to fixed page
39 markup.

40 *Table 16-1. StoryFragments part elements*

Name	Description
<code><StoryFragments></code>	Root element.
<code><StoryFragment></code>	Contains all content structure markup elements for a single story fragment.

<StoryBreak>	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 can be split across pages, the <StoryBreak>
 2 element is provided to identify that a given element continues *to* the next story fragment or
 3 continues *from* a previous story fragment. A <StoryBreak> element MUST NOT be included in a
 4 position other than the first or last child element of a <StoryFragment> element [M9.12M2.72].

5 If a <StoryBreak> element is not present at the beginning of the content structure markup,
 6 consumers SHOULD consider the markup a continuation of the previous story fragment that
 7 must be merged [S9.4]. Likewise, if a <StoryBreak> element is not present at the end of the
 8 content structure markup, consumers SHOULD consider the markup a continuation to the next
 9 story fragment that must be merged to determine the cross-fragment content structure [S9.4].

10 Content structure is merged on an element-by-element basis, merging the last element closed
 11 in the leading story fragment with the first element opened in the trailing story fragment. This
 12 process continues until the closing tag from the leading story fragment no longer matches the
 13 opening tag from the trailing story fragment.

14 <TableCellStructure> elements require special merging, such that all <TableCellStructure>
 15 elements within a <TableRowStructure> element are merged. In order to merge the table cells
 16 and rows correctly, producers MUST specify empty <TableCellStructure> elements for cells that
 17 do not break across story fragments [M9.1].

1 *Example 16-5. Content structure spanning pages*

2 Given the following two StoryFragments parts, consumers can construct the content structure
3 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" />

```

```

1         </ParagraphStructure>
2         <StoryBreak />
3     </StoryFragment>
4 </StoryFragments>

5
6 <!-- Second StoryFragments part -->
7
8 <StoryFragments
9     xmlns="http://schemas.microsoft.com/xps/2005/06/documentstructure">
10    <StoryFragment FragmentType="Header">
11        <StoryBreak />
12        <ParagraphStructure>
13            <NamedElement NameReference="Block9" />
14        </ParagraphStructure>
15        <StoryBreak />
16    </StoryFragment>
17    <StoryFragment StoryName="Story1" FragmentType="Content">
18        <SectionStructure>
19            <TableStructure>
20                <TableRowGroupStructure>
21                    <TableRowStructure>
22                        <TableCellStructure />
23                        <TableCellStructure>
24                            <ParagraphStructure>
25                                <NamedElement NameReference="Block10" />
26                                <NamedElement NameReference="Block11" />
27                            </ParagraphStructure>
28                        </TableCellStructure>
29                    </TableRowStructure>
30                    <TableRowStructure>
31                        <TableCellStructure>
32                            <ParagraphStructure>
33                                <NamedElement NameReference="Block12" />
34                            </ParagraphStructure>
35                        </TableCellStructure>
36                        <TableCellStructure>
37                            <ParagraphStructure>
38                                <NamedElement NameReference="Block13" />
39                            </ParagraphStructure>
40                        </TableCellStructure>
41                    </TableRowStructure>
42                </TableRowGroupStructure>
43            </TableStructure>
44        </SectionStructure>
45        <StoryBreak />
46    </StoryFragment>
47    <StoryFragment FragmentType="Footer">
48        <StoryBreak />
49        <ParagraphStructure>
50            <NamedElement NameReference="Block14" />
51        </ParagraphStructure>
52        <StoryBreak />
53    </StoryFragment>
54 </StoryFragments>

```

```

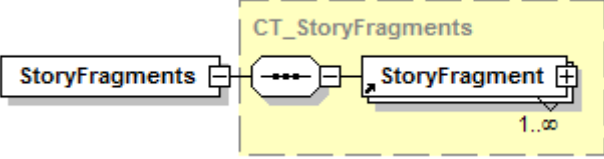
1
2   <!-- Resulting merged content structure for Story1 -->
3
4   <SectionStructure>
5     <TableStructure>
6       <TableRowGroupStructure>
7         <TableRowStructure>
8           <TableCellStructure>
9             <ParagraphStructure>
10              <NamedElement NameReference="Block2" />
11              <NamedElement NameReference="Block3" />
12            </ParagraphStructure>
13          </TableCellStructure>
14          <TableCellStructure>
15            <ParagraphStructure>
16              <NamedElement NameReference="Block4" />
17            </ParagraphStructure>
18          </TableCellStructure>
19        </TableRowStructure>
20        <TableRowStructure>
21          <TableCellStructure>
22            <ParagraphStructure>
23              <NamedElement NameReference="Block5" />
24              <NamedElement NameReference="Block6" />
25            </ParagraphStructure>
26          </TableCellStructure>
27          <TableCellStructure>
28            <ParagraphStructure>
29              <NamedElement NameReference="Block7" />
30              <NamedElement NameReference="Block10" />
31              <NamedElement NameReference="Block11" />
32            </ParagraphStructure>
33          </TableCellStructure>
34        </TableRowStructure>
35        <TableRowStructure>
36          <TableCellStructure>
37            <ParagraphStructure>
38              <NamedElement NameReference="Block12" />
39            </ParagraphStructure>
40          </TableCellStructure>
41          <TableCellStructure>
42            <ParagraphStructure>
43              <NamedElement NameReference="Block13" />
44            </ParagraphStructure>
45          </TableCellStructure>
46        </TableRowStructure>
47      </TableRowGroupStructure>
48    </TableStructure>
49  </SectionStructure>

```

50 *end example]*

51 **16.1.2.1 <StoryFragments> Element**

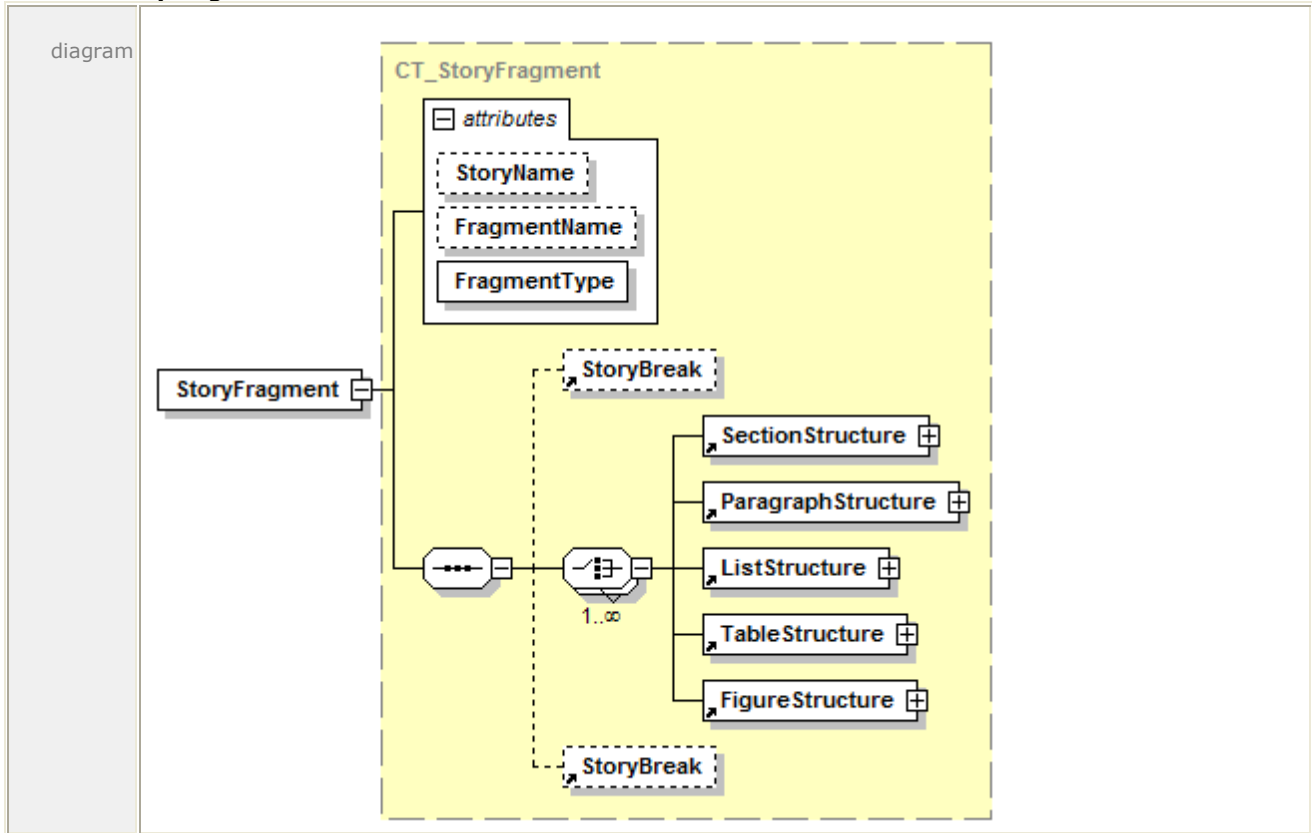
52 element **StoryFragments**

<p>diagram</p>	 <p>The diagram shows a class <code>StoryFragments</code> on the left and a class <code>StoryFragment</code> on the right. A solid line with an open arrowhead at the <code>StoryFragment</code> end connects them, indicating a composition relationship. The <code>StoryFragment</code> class has a multiplicity of <code>1..∞</code> at its bottom right corner. A dashed yellow box encloses the <code>StoryFragment</code> class and the relationship line, with the label <code>CT_StoryFragments</code> above it.</p>
<p>annotation</p>	<p>The root of a StoryFragments part. Contains all story fragments that appear on a specific page.</p>

- 1 The `<StoryFragments>` element groups all of the `<StoryFragment>` elements on a page.

1 **16.1.2.2 <StoryFragment> Element**

2 element **StoryFragment**



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.

- 3
- 4 Each <StoryFragment> has a StoryName attribute that associates it with a story defined in the
- 5 DocumentStructure part. It also has a FragmentType attribute, the values for which are Content
- 6 (the default), Header, or Footer.

1 Headers and footers are defined in their own story fragment on each page. These stories do not
 2 specify a StoryName value, so they are essentially unreferenced stories that exist only on a
 3 single page.

4 Producers authoring document structure information SHOULD reference every element of the
 5 fixed page markup that has semantic meaning (such as text or images) in the StoryFragments
 6 parts [S9.5].

7 *Example 16–6. StoryFragments part markup*

8 The following markup describes the StoryFragments part of a one-page document:

```

9     <StoryFragments
10         xmlns="http://schemas.microsoft.com/xps/2005/06/documentstructure">
11         <StoryFragment FragmentType="Header">
12             <StoryBreak />
13             <ParagraphStructure>
14                 <NamedElement NameReference="Block13" />
15                 <NamedElement NameReference="Block14" />
16             </ParagraphStructure>
17             <StoryBreak />
18         </StoryFragment>
19         <StoryFragment StoryName="Story1" FragmentType="Content">
20             <StoryBreak />
21             <ParagraphStructure>
22                 <NamedElement NameReference="Block1" />
23                 <NamedElement NameReference="Block2" />
24             </ParagraphStructure>
25             <TableStructure>
26                 <TableRowGroupStructure>
27                     <TableRowStructure>
28                         <TableCellStructure>
29                             <ParagraphStructure>
30                                 <NamedElement NameReference="Block3" />
31                                 <NamedElement NameReference="Block4" />
32                             </ParagraphStructure>
33                         </TableCellStructure>
34                         <TableCellStructure>
35                             <ParagraphStructure>
36                                 <NamedElement NameReference="Block5" />
37                             </ParagraphStructure>
38                         </TableCellStructure>
39                     </TableRowStructure>
40                 </TableRowGroupStructure>
41             </TableStructure>
42             <SectionStructure>
43                 <ParagraphStructure>
44                     <NamedElement NameReference="Block6" />
45                 </ParagraphStructure>
46                 <ParagraphStructure>
47                     <NamedElement NameReference="Block7" />
48                     <NamedElement NameReference="Block8" />
49                 </ParagraphStructure>
50             </SectionStructure>
51             <SectionStructure>
52                 <FigureStructure>

```

```

1         <NamedElement NameReference="Block9" />
2     </FigureStructure>
3     <ListStructure>
4         <ListItemStructure>
5             <ParagraphStructure>
6                 <NamedElement NameReference="Block10" />
7             </ParagraphStructure>
8         </ListItemStructure>
9         <ListItemStructure>
10            <ParagraphStructure>
11                <NamedElement NameReference="Block11" />
12            </ParagraphStructure>
13        </ListItemStructure>
14        <ListItemStructure>
15            <ParagraphStructure>
16                <NamedElement NameReference="Block12" />
17            </ParagraphStructure>
18        </ListItemStructure>
19    </ListStructure>
20 </SectionStructure>
21 <StoryBreak />
22 </StoryFragment>
23 <StoryFragment FragmentType="Footer">
24     <StoryBreak />
25     <ParagraphStructure>
26         <NamedElement NameReference="Block15" />
27         <NamedElement NameReference="Block16" />
28         <NamedElement NameReference="Block17" />
29     </ParagraphStructure>
30     <StoryBreak />
31 </StoryFragment>
32 </StoryFragments>

```

33 *end example]*

34 A <StoryFragment> element MAY be identified with a FragmentName attribute to distinguish it
35 from other fragments for the same story on a single page [M2.72].

36 *Example 16–7. Story fragments markup using a fragment name*

```

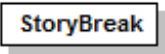
37     <StoryFragments
38         xmlns="http://schemas.microsoft.com/xps/2005/06/documentstructure">
39         <StoryFragment
40             StoryName="Story1"
41             FragmentName="Fr1"
42             FragmentType="Content">
43             <StoryBreak />
44             <ParagraphStructure>
45                 <NamedElement NameReference="Block1" />
46                 <NamedElement NameReference="Block2" />
47             </ParagraphStructure>
48             <StoryBreak />
49         </StoryFragment>
50         <StoryFragment
51             StoryName="Story1"
52             FragmentName="Fr2"

```

```
1      FragmentType="Content">
2      <StoryBreak />
3      <ParagraphStructure>
4          <NamedElement NameReference="Block8" />
5      </ParagraphStructure>
6      <StoryBreak />
7  </StoryFragment>
8  </StoryFragments>
9  end example]
```


1 **16.1.2.3 <StoryBreak> Element**

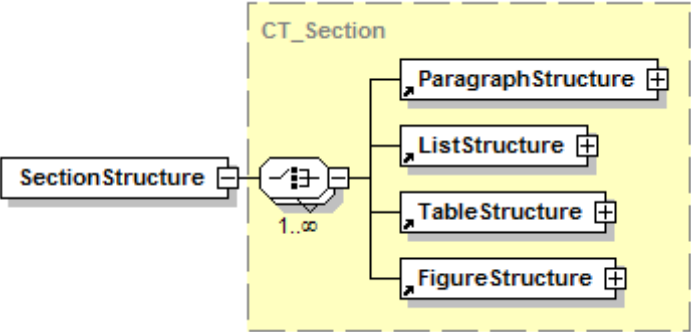
2 element **StoryBreak**

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 story
4 fragments to determine the content structure.

5 **16.1.2.4 <SectionStructure> Element**

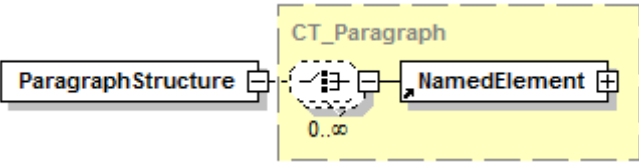
6 element **SectionStructure**

diagram	
annotation	Provides an arbitrary grouping of content structural markup elements.

7 The <SectionStructure> element provides an arbitrary grouping of <Paragraph>,
8 <TableStructure>,<ListStructure>,<FigureStructure> elements.

9 **16.1.2.5 <ParagraphStructure> Element**

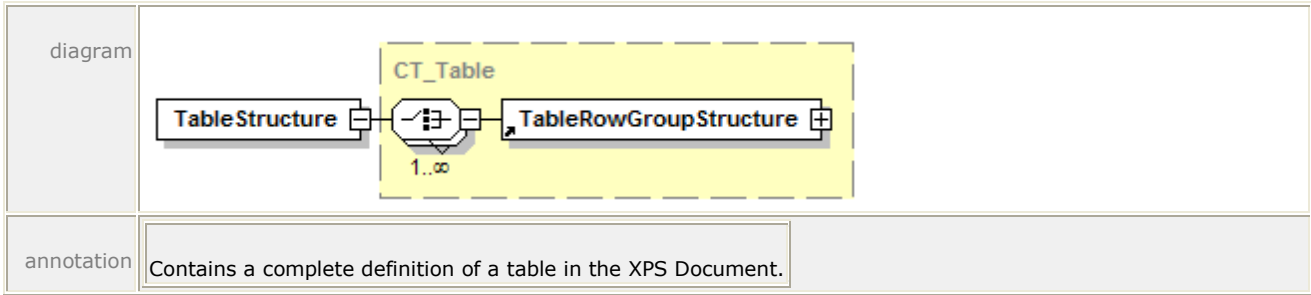
10 element **ParagraphStructure**

diagram	
annotation	Contains the named elements that constitute a single paragraph.

11 A <ParagraphStructure> element describes the list of <NamedElement> elements that
12 constitute a single paragraph.

1 **16.1.2.6 <TableStructure> Element**

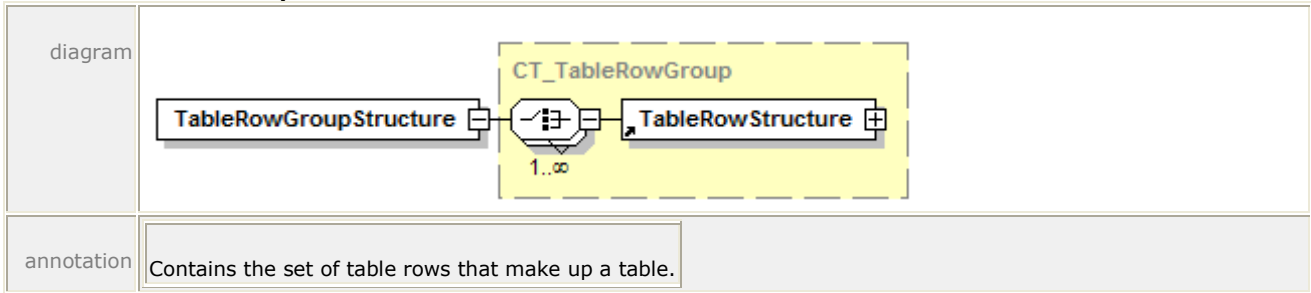
2 element **TableStructure**



3 A <TableStructure> element is the complete definition of a table. An implementation MAY use it
 4 to build special functionality, such as row or column selection [O9.5].

5 **16.1.2.7 <TableRowGroupStructure> Element**

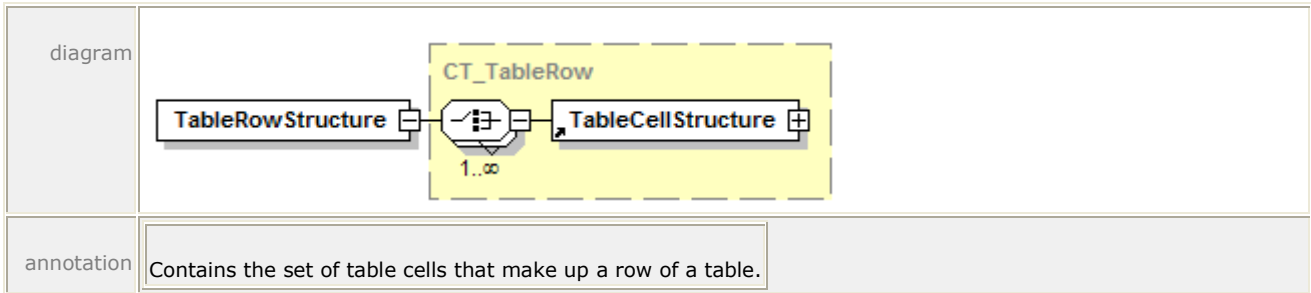
6 element **TableRowGroupStructure**



7 A <TableRowGroupStructure> element is REQUIRED in order to specify a set of
 8 <TableRowStructure> elements [M9.13M2.72].

9 **16.1.2.8 <TableRowStructure> Element**

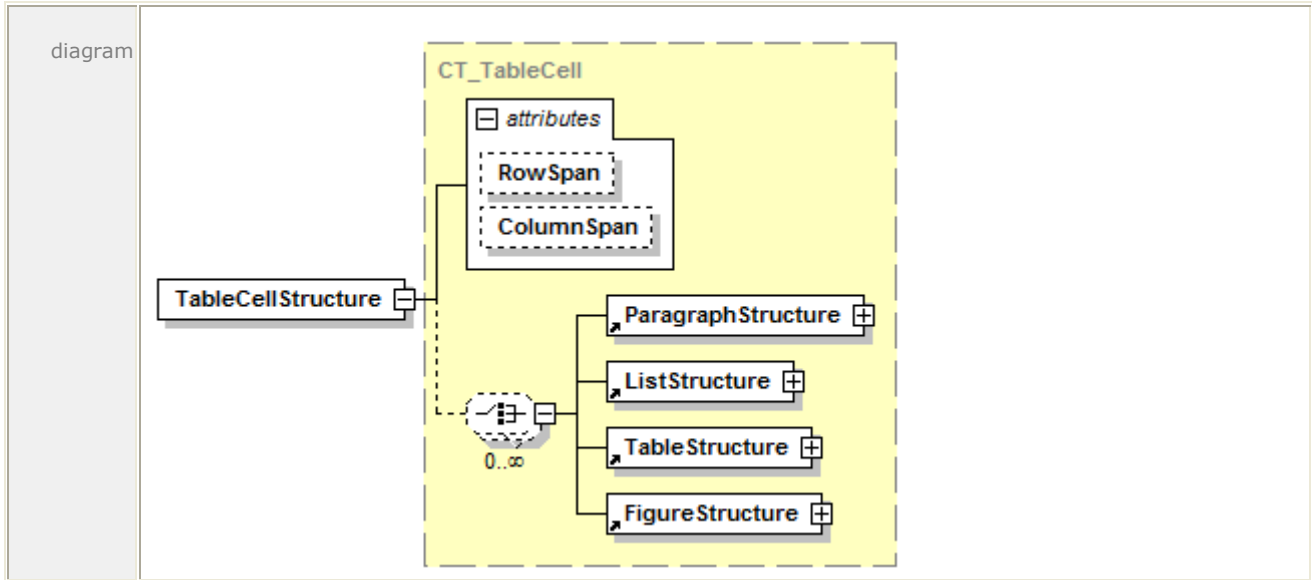
10 element **TableRowStructure**



11 This element groups <TableCellStructure> child elements that define a single row of a table.

1 **16.1.2.9 <TableCellStructure> Element**

2 element **TableCellStructure**



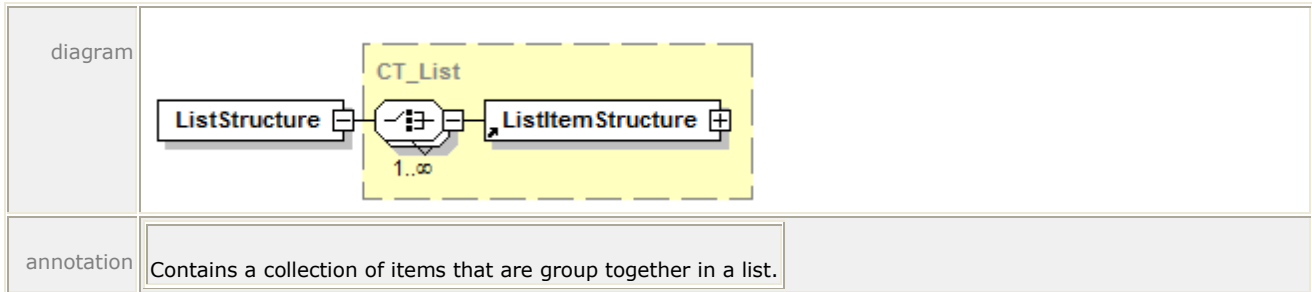
attributes	Name	Type	Use	Default	Fixed	Annotation
	RowSpan	<u>ST_TableSpan</u>	optional	1		Indicates the number of rows this cell spans, or merges into a single cell.
	ColumnSpan	<u>ST_TableSpan</u>	optional	1		Indicates the number of columns this cell spans, or merges into a single cell.

annotation: Contains the elements that occupy a single cell of a table.

3 This element defines the appearance of a table cell. It MAY contain nested <TableStructure>
 4 elements [09.16M2.72].

5 **16.1.2.10 <ListStructure> Element**

6 element **ListStructure**



annotation: Contains a collection of items that are group together in a list.

7 The <ListStructure> element is the complete definition of a list of related items.

1 **16.1.2.11 <ListItemStructure> Element**

2 element **ListItemStructure**

diagram						
attributes	Name	Type	Use	Default	Fixed	Annotation
	Marker	<u>ST_NameUnique</u>	optional			The named element that represents the marker for this list items, such as a bullet, number, or image.
annotation	Describes a single structural block. These structural blocks are grouped together in a list.					

3 A <ListItemStructure> element defines a single item in a list.

4 **16.1.2.12 <FigureStructure> Element**

5 element **FigureStructure**

diagram						
annotation	Groups the named elements that constitute a single drawing or diagram.					

6 A <FigureStructure> element includes a group of named elements that comprise a single
 7 drawing or diagram.

1 **16.1.2.13 <NamedElement> Element**2 element **NamedElement**

diagram						
attributes	Name	Type	Use	Default	Fixed	Annotation
	NameReference	<u>ST_Name</u>	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 NameReference
4 attribute to specify an element in the fixed page markup with a corresponding name.

5 If the targeted fixed page uses markup compatibility markup that changes the presence of
6 certain named elements, the StoryFragments part should also use it in order to reference each
7 element in either representation.

8 **16.2 Hyperlinks**

9 If consumers enable user interactivity, they SHOULD support hyperlink activation and
10 addressing [S9.6].

11 **16.2.1 Hyperlink Activation**

12 Hyperlinks are specified inline on any <Canvas>, <Path>, or <Glyphs> element by means of
13 the FixedPage.NavigateUri attribute. The value of the attribute is the destination URI. If
14 hyperlinked <Path> or <Glyphs> elements are rendered as overlapping on the page,
15 consumers MUST treat the topmost element as the only hyperlink that can be activated in the
16 overlapping region [M9.2].

17 When activating a hyperlink, consumers SHOULD load the specified resource if they understand
18 the URI type. If the URI is an internal reference to the XPS Document, consumers SHOULD
19 navigate to the URI [S9.7].

20 If a producer specifies a FixedPage.NavigateUri attribute on a <Canvas> element, consumers
21 MUST treat all child elements of that canvas as having an associated hyperlink [M9.3]. Child or
22 descendant elements can override this value with their own FixedPage.NavigateUri attribute.

23 Relative internal hyperlinks between FixedPage parts MUST specify, at a minimum, the named
24 address relative to the FixedDocument part [M9.4].

1 Producers can mark any <FixedPage>, <Canvas>, <Path>, or <Glyphs> element as an
2 addressable location within the XPS Document by specifying a value for the Name attribute. The
3 name SHOULD be unique within the scope of the fixed document [S9.8]. If it is not unique, only
4 the first occurrence of the named address is addressable.

5 These elements, if specified as a <VisualBrush.Visual> property element are not addressable by
6 a hyperlink.

7 It is RECOMMENDED that Name attribute values be unique within an entire fixed document
8 sequence [S9.9]. If they are not, only the first occurrence of the named address is addressable
9 from an external location. Internal hyperlinks can specify a named element fragment relative to
10 a particular fixed document, but consumers MAY interpret such a URI relative to the entire fixed
11 document sequence instead [O9.6].

12 In order to be addressable by either a hyperlink or the document outline, the named address
13 MUST appear in the <PageContent.LinkTargets> element in the fixed document [M9.5]. If a
14 named address appears in the <PageContent.LinkTargets> element in the fixed document but
15 is not found in the Name attribute of an element within the associated fixed page, consumers
16 MUST treat the top of the associated fixed page as the named address [M9.6]. If the named
17 address in a URI fragment is not found, consumers MUST ignore the fragment portion of the
18 URI [M9.7].

19 *Example 16–8. A relative, internal, named-address hyperlink*

```
20 FixedPage.NavigateUri=" ../MyDocument.fdoc#MyAddress"
```

21 *end example]*

22 16.2.2 Hyperlink Addressing

23 XPS Documents specify two forms of URI fragment identifiers to address locations within an
24 XPS Document. The first is a named address. [*Example*: "http://xps/MyPackage#MyAddress",
25 where "http://xps/MyPackage" is an XPS Document and "MyAddress" is a named address within
26 the document. *end example*] The second is an absolute page number within the XPS Document.
27 [*Example*: "http://xps/MyPackage#15", where "15" references the FixedPage part associated
28 with the fifteenth <PageContent> entry among all the fixed documents in the fixed document
29 sequence. *end example*]

30 Page number fragment identifiers refer to the absolute page number (1-based) in the fixed
31 document sequence. [*Example*: If an XPS Document has a 3-page fixed document, followed by
32 a 10-page fixed document, followed by an 8-page fixed document, the fragment identifier
33 "#15" refers to the second page of the third fixed document in the fixed document sequence.
34 *end example*] Internal references MUST specify a page address relative to the fixed document
35 sequence [M9.8].

36 *Example 16–9. A relative internal page address hyperlink*

```
37 FixedPage.NavigateUri=" ../ ../ ../MyDocSeq.fdoc#12"
```

38 *end example]*

39 16.2.3 Name Attribute

40 The Name attribute contains a string value that identifies the current element as a named,
41 addressable point for the purpose of hyperlinking. The Name attribute is optional. Names
42 SHOULD be unique within a fixed document [S9.8], and it is RECOMMENDED that they be

1 unique within a fixed document sequence [S9.9]. The Name attribute MUST NOT be specified on
2 any children of a <ResourceDictionary> element [M9.10].

3 If the Name attribute is specified, producers SHOULD also create a corresponding <LinkTarget>
4 element in the FixedDocument part within the <PageContent> element that links to the parent
5 fixed page [S9.10]. Consumers MAY ignore this attribute [O9.7], but devices that support user
6 interaction with the contents of XPS Documents SHOULD support hyperlinks [S9.6].

7 The Name value, if specified, MUST meet the following requirements [M9.14M2.72]:

- 8 1. The initial character MUST be an underscore character or a letter, that is, it falls within
9 the Lu, Ll, Lo, Lt, and Nl categories [M9.14M2.72].
- 10 2. Trailing characters MUST be an underscore character or a letter or number, that is, they
11 fall within the Lu, Ll, Lo, Lt, Nl, Mn, Mc, and Nd categories [M9.14M2.72].

12 [*Note*: These requirements match those of XML identifiers with additional restrictions. *end note*]

13 The category abbreviations, as defined within the Unicode Character Database, are partially
14 reproduced in Table 16–2.

15 *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

16 **16.2.4 FixedPage.NavigateUri Attribute**

17 The FixedPage.NavigateUri attribute associates a hyperlink URI with an element, making it a
18 hyperlink source. Its value can be a relative or absolute URI that addresses a resource that is
19 internal or external to the XPS Document package, respectively. The base URI used to resolve a
20 relative URI is that of the FixedPage part in which the element with the FixedPage.NavigateUri
21 attribute appears. Therefore, a hyperlink to a destination within the fixed document of the
22 source MUST specify the destination in the context of the FixedDocument part [M9.4].
23 [*Example*: “./FixedDoc1.fdoc#MyDestination”. *end example*] A destination in the same fixed
24 document SHOULD be expressed as a relative URI [S9.11].

25 The FixedPage.NavigateUri attribute is OPTIONAL [O9.17M2.72]. It SHOULD be included *only* if
26 the element is intended to be a hyperlink. Consumers MAY ignore this attribute [O9.8], but
27 devices that support user interaction with the contents of XPS Documents SHOULD support
28 hyperlinks [S9.6].

16.3 Selection

Viewing consumers that support interactivity MAY support selection and copying [O9.9]. Selection order within an XPS Document SHOULD follow reading order [S9.13].

Consumers MAY use the FragmentType 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 [O9.10~~O8.10~~].

16.4 Accessibility

Accessibility refers to features that are important to provide equal access to XPS Documents for users of all abilities. One common example of an accessibility application is a screen reader, which reads the contents of a document aloud for vision-impaired individuals.

16.4.1 Reading Order

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 [O9.11], but SHOULD, at minimum, rely on the markup order to determine reading order [S9.14]. Producers SHOULD order the markup in FixedPage parts to reflect the order in which it is intended to be read [S9.15]. 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 [S9.16].

The RECOMMENDED reading order of a page-centric application is as follows [S9.17]:

- Order the content by page.
- Within a page, order by story fragment in the order the <StoryFragment> elements are specified in the StoryFragments part for that page. Producers SHOULD order <StoryFragment> elements in their intended reading order [S9.18].
- Within a <StoryFragment> element, order by <NamedElement> reference.
- Append all un-referenced elements that appear in the fixed page markup, ordered by markup order.

Although producers SHOULD reference every element of the fixed page markup in the content structure markup [S9.10], 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 [M9.9].

Consumers MAY use the FragmentType attribute of the <StoryFragment> element to determine reading order by interpreting elements that have FragmentType values of Header and Footer as belonging first or last in the reading order, respectively [O9.12].

The RECOMMENDED reading order of a story-centric application is as follows [S9.19]:

- Order content by story in the sequence the <Story> elements appear in the DocumentStructure part. Producers SHOULD order <Story> elements in their intended reading order [S9.20].
- Within a story, order <StoryFragmentReference> elements in the sequence they appear in the DocumentStructure part. Producers SHOULD order <StoryFragmentReference> elements in their intended reading order [S9.21].

- 1 • Within a story fragment, order by <NamedElement> references in the StoryFragments
2 part markup.
- 3 • Append all un-referenced elements that appear in the fixed page markup, ordered by
4 page number, then markup order.

5 **16.4.2 Screen Reader Applications**

6 Screen reader applications read the contents of the document aloud. A screen reader consumer
7 SHOULD read the document according to its reading order [S9.22]. The application SHOULD
8 use the UnicodeString attribute of each <Glyphs> element [S9.23]. In addition, screen readers
9 MAY inspect the Indices attribute to resolve potential ambiguities [O9.13].

10 If the screen reader provides features to navigate the document by structural elements, such as
11 paragraphs or table rows, it SHOULD use any document structure information included in the
12 XPS Document [S9.24].

13 If the screen reader provides features to describe images, it SHOULD read the text provided in
14 the AutomationProperties.Name and AutomationProperties.HelpText attributes [S9.25].

15 If the screen reader provides features to describe hyperlink addresses, it SHOULD read the text
16 provided in the FixedPage.NavigateUri attribute [S9.26].

17 **16.4.3 Text Alternatives for Graphics and Images**

18 Images and graphics SHOULD specify text alternatives for images and graphics to make this
19 content accessible to vision-impaired individuals [S9.27]. There are short and long textual
20 descriptions, specified in the AutomationProperties.Name and AutomationProperties.HelpText
21 attributes of <Path> and <Canvas>, respectively.

22 The AutomationProperties.Name attribute SHOULD contain a short description of the basic
23 contents of the image or vector graphic [S9.27]. [*Example: "A sitting dog." end example*]The
24 AutomationProperties.HelpText attribute can contain a more detailed description of the image or
25 graphic. [*Example: "A cocker spaniel with brown eyes, golden fur, and its tongue hanging out.
26 It is sitting on a beanbag directly facing the camera." end example*]

27 An image SHOULD specify the AutomationProperties.Name and AutomationProperties.HelpText
28 attributes on the <Path> element that is filled with an <ImageBrush> [S9.28]. These attributes
29 describe the content specified by the ImageSource attribute of the <ImageBrush> element.

30 A vector graphic (a collection of one or more <Path> elements representing a single drawing)
31 SHOULD specify the AutomationProperties.Name and AutomationProperties.HelpText attributes only
32 once, directly on a <Canvas> element wrapping the <Path> elements comprising the graphic
33 [S9.29].

34 Individual <Path> elements that do not provide any semantic meaning (such as a line between
35 sections or outlining a table) SHOULD NOT specify these text alternative attributes [S9.27].

36

17. XPS Document Package Features

The XPS Document format extends package-level interleaving and digital signatures as described in the OPC Standard.

17.1 Interleaving Optimizations

Interleaving concerns the physical organization of XPS Documents, rather than their logical structure. It allows consumers to process linearly the bytes that make up a physical package from start to finish, without regard for context. In other words, consumers can make correct determinations about the types of logical parts and the presence of relationships on a logical part when consuming packages in a linear fashion. Consumers are never required to return to previously encountered parts and revise their determination of the content type or presence of relationships.

Interleaving is OPTIONAL [O10.1]. However, if the XPS Document is interleaved, these rules SHOULD be followed:

- The Content Types stream SHOULD be interleaved according to the recommendations in the OPC Standard [S10.1].
- PrintTicket parts SHOULD be written to the package before the part to which they are attached [S10.2].
- 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 [S10.3].
- 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 which it is attached [S10.4].
- 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 [S10.5].
- The relationships for the DiscardControl part and the StartPart SHOULD both be written in the first piece of the package relationship part, and that piece SHOULD be before the first FixedPage part in the package [S10.20].
- The piece of the DiscardControl part that includes a Discard element with a SentinelPage attribute referencing a FixedPage part SHOULD be written to the package before that FixedPage part [S10.21].

Following these recommendations allows more efficient processing by certain consumers. Not following these recommendations could result in less efficient processing by most consumers because they will need to wait until all parts required to process a part (attached PrintTicket, required resources) have been consumed. However, consumers MUST be prepared to process correctly packages in which the PrintTicket or the portion of the relationship data attaching the PrintTicket appears in the package after the affected part [M10.1].

- 1 Consumers can choose to parse an XPS Document in a head-first or tail-first manner. Tail-first
2 parsing reveals certain package errors earlier, such as inconsistencies between the ZIP central
3 directory and local file headers. Head-first XPS Document consumers SHOULD attempt to detect
4 inconsistent packages as soon as possible and SHOULD ~~generate~~ [instantiate an error condition](#)
5 ~~an error message~~, even if they have already processed the pages that resulted in the error
6 [S10.18]. Head-first consumers that discard parts would need to retain the name and length of
7 any discarded part to comply with this recommendation.
- 8 [*Note*: Streaming and handling of discard control are complicated significantly by any
9 requirement for out-of-order page handling, such as in the production of booklets. *end note*]

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/printschemafra
5 work" version="1"/>
```

6 It is RECOMMENDED that one empty PrintTicket be shared for all parts that attach an empty
7 PrintTicket [S10.6]. [The content of an empty PrintTicket is implementation-defined \(see](#)
8 [§9.1.9\).](#)

9 **17.1.2 Optimizing Interleaving Order**

10 Producers MAY optimize the interleaving order of parts to help consumers avoid stalls during
11 read-time streaming, and to allow consumers to manage their memory resources more
12 efficiently [O10.2].

13 The optimization strategy is suggested by the consumer architecture. Therefore, interleaving
14 optimization is typically implemented by a software component such as a driver or filter that is
15 specific to (or aware of) the consumer architecture.

16 **17.1.2.1 Single-Threaded Parsing Architectures**

17 An optimal interleaving scheme for consumers with a single-threaded parsing model interleaves
18 parts so that each part that is required to consume a single page (FixedPage, images, and
19 fonts) is contained in the package in its entirety, prior to the FixedPage part being referenced
20 from the FixedDocument part's markup.

21 Single-threaded parsing architectures typically require more run-time memory resources than
22 multi-threaded parsing architectures because the context in which a resource is used is
23 unknown at the time the resource is received. This requires deferred processing and additional
24 buffering.

25 *[Note: When interleaving entities containing XML markup, such as the DiscardControl part, the*
26 *Content Types stream, and the FixedDocument part, there is no guarantee that XML element*
27 *boundaries will align with piece boundaries in the physical package. This adds a complexity to*
28 *single-threaded parsing architectures: the parser must be pre-emptable. Certain existing XML*
29 *parser implementations might require a 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 two
32 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	<pre><Relationships xmlns= "http://schemas.openxmlformats.org/package/2006/re lationships"> <Relationship Type= "http://schemas.microsoft.com/xps/2005/06 /required-resource" Target="Font1.ttf"/> </Relationships></pre>
FixedDocument1/[0].piece	<pre><FixedDocument xmlns= "http://schemas.microsoft.com/xps/2005/06"> <PageContent Source="Page1"/></pre>
Sequence1/[0].piece	<pre><FixedDocumentSequence xmlns= "http://schemas.microsoft.com/xps/2005/06"> <DocumentReference Source="FixedDocument1"/></pre>
_rels/.rels/[0].piece	<pre><Relationships xmlns= "http://schemas.openxmlformats.org/package/2006/re lationships"> <Relationship Type="StartPart" Target="Sequence1"/></pre>
Page2	<pre><FixedPage xmlns= "http://schemas.microsoft.com/xps/2005/06" ...>...</FixedPage></pre>
FixedDocument1/[1].last.pie ce	<pre><PageContent Source="Page2"/> </FixedDocument></pre>
Page3	<pre><FixedPage xmlns= "http://schemas.microsoft.com/xps/2005/06" ...>...</FixedPage></pre>
FixedDocument2	<pre><FixedDocument xmlns= "http://schemas.microsoft.com/xps/2005/06"> <PageContent Source="Page3"/> </FixedDocument></pre>
Sequence1/[1].last.piece	<pre><DocumentReference Source="FixedDocument2" /> </FixedDocumentSequence></pre>
_rels/.rels/[1].last.piece	<pre></Relationships></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 interleaves
4 parts so that each resource part that is required to consume a single page (images and fonts) is
5 contained in the package after the FixedPage part referencing it.

6 Multi-threaded parsing architectures typically require less run-time memory resources than
7 single-threaded parsing architectures because the context in which resources appear is fully
8 determined and, therefore, resources can be processed immediately.

9 [*Note: When interleaving entities containing XML markup, such as the DiscardControl part, the*
10 *content type stream, and the FixedDocument part, there is no guarantee that XML element*
11 *boundaries will align with piece boundaries in the physical package. A multi-threaded parsing*
12 *architecture is naturally suited to address this 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 having two
3 FixedPage parts and the second having one FixedPage part:

4

Part/Piece	Markup
_rels/.rels/[0].piece	<pre><Relationships xmlns="http://schemas.microsoft.com/package /2005/06/relationships"> <Relationship Type="StartPart" Target= "Sequence1"/></pre>
Sequence1/[0].piece	<pre><FixedDocumentSequence xmlns= "http://schemas.microsoft.com/xps/2005/06"> <DocumentReference Source="FixedDocument1"/></pre>
FixedDocument1/[0].piece	<pre><FixedDocument xmlns= "http://schemas.microsoft.com/xps/2005/06"> <PageContent Source="Page1"/></pre>
Page1.rels	<pre><Relationships xmlns= "http://schemas.openxmlformats.org/package/2006/re lationships"> <Relationship Type= "http://schemas.microsoft.com/xps/2005/06 /required-resource" Target="Font1.ttf"/> </Relationships></pre>
Page1	<pre><FixedPage xmlns="http://schemas.microsoft.com /xps/2005/06" ...> <Glyphs FontURI="Font1.ttf"/> </FixedPage></pre>
Font1.ttf	...binary font data...
Other resources	...resource data...
FixedDocument1/[1].last.pie ce	<pre><PageContent Source="Page2"/></pre>
Page2	<pre><FixedPage xmlns= "http://schemas.microsoft.com/xps/2005/06" ...>...</FixedPage></pre>
FixedDocument1/[2].last.pie ce	<pre></FixedDocument></pre>
Sequence1/[1].last.piece	<pre><DocumentReference Source="FixedDocument2" /> </FixedDocumentSequence></pre>
FixedDocument2	<pre><FixedDocument xmlns= "http://schemas.microsoft.com/xps/2005/06"> <PageContent Source="Page3"/> </FixedDocument></pre>
Page3	<pre><FixedPage xmlns="http://schemas.microsoft.com/xps/2005/06"</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 structure
4 [M10.2]. ~~Consumers that lack the resources to process a part MUST indicate an error condition~~
5 ~~[M10.3]. Such a resource constraint exists when a consumer lacks sufficient memory resources~~
6 ~~to hold enough of the package to resolve all the references required to process a part.~~

7 To address resource constraints:

- 8 • Consumers MAY discard FixedPage parts once they have been processed [O10.3]
- 9 • Consumers MAY discard FixedDocument and FixedDocumentSequence parts after all their
10 child elements and their closing tags have been processed [O10.4].
- 11 • In the absence of explicit directives to the contrary (see §17.1.4), consumers MAY
12 discard parts as directed by the DiscardControl part [O10.5]. Consumers MUST NOT
13 discard any other parts [*Example*: Such as parts containing fonts, images, or other
14 resources *end example*] unless they have the ability to access the parts again [M10.4].

15 If a consumer encounters a reference to an unknown part, it MUST continue to receive further
16 bytes of the package until the unknown part has been transmitted *or* until the end of the
17 package is reached (indicating an error condition) [M10.5].

18 **17.1.4 Consumers with Resource Constraints**

19 To produce an XPS Document for streaming consumption by consumers with limited memory
20 resources, some producers MAY choose a suitable interleaving order by modeling the resource
21 management behavior of the consumer [O10.6]. These producers, referred to as *drivers*, must
22 have specific knowledge of the XPS Document consumer. Due to resource constraints, some
23 consumers are unable to consume arbitrary XPS Documents and always require assistance from
24 an external driver.

25 When some consumers with limited memory resources receive a XPS Document in a streaming
26 fashion, there might be an opportunity to discard parts when necessary and reload them again
27 when needed. Producers, such as drivers, that target such consumers SHOULD follow these
28 steps [S10.7]:

- 29 • Conservatively model the memory usage of the device.
- 30 • Interleave pieces of parts in the correct order.
- 31 • Decide when certain parts can be discarded by the consumer and inform the consumer
32 within the package stream (see §17.1.4.1).
- 33 • Add to the package a uniquely named copy of a resource that could have been discarded,
34 if the resource is referenced by a part sent later in the stream. Those later references
35 are also updated to refer to the new copy of the resource.

36 **17.1.4.1 DiscardControl Part**

37 In addition to optimally ordering interleaved parts, producers can support consumers with
38 resource constraints by means of the DiscardControl part. The DiscardControl part is a well-
39 known part containing a list of resources that are safe for the consumer to discard.

1 DiscardControl parts are stored in XPS Documents in an interleaved fashion, allowing a
 2 resource-constrained consumer to discard a part as soon as it appears in the DiscardControl
 3 part. DiscardControl parts are targeted with a DiscardControl package relationship, as specified
 4 in §A. There MUST NOT be more than one DiscardControl package relationship [M10.23]. The
 5 DiscardControl part MUST NOT reference itself [M10.6]; doing so is considered an error.

6 DiscardControl parts that are not well-formed SHOULD NOT be processed and an error
 7 [condition](#) SHOULD NOT be ~~reported~~ [instantiated](#) [S10.8]. The consumer MAY decide to ignore
 8 the malformed DiscardControl part in its entirety or from the first malformed node onward
 9 [O10.7].

10 In some cases, producers might rewrite the contents of a package so that parts are provided
 11 more than once, allowing consumers to discard a part in order to free resources for additional
 12 processing. Each instance of a part MUST be stored as a new, uniquely named part in the
 13 package [M10.24].

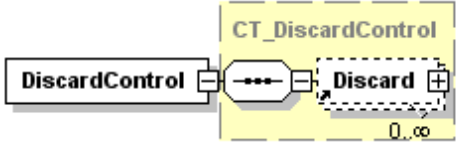
14 *Example 17–3. A DiscardControl part*

```
15 <DiscardControl xmlns="http://schemas.microsoft.com/xps/2005/06/discard-
16 control">
17 <!-- May discard partname1 as soon as starting to process
18 page11.xaml -->
19 <Discard SentinelPage="/page11.xaml" Target="/partname1" />
20 <!-- May discard partname2 as soon as starting to process
21 page13.xaml -->
22 <Discard SentinelPage="/page13.xaml" Target="/partname2" />
23 ...
24 </DiscardControl>
```

25 *end example]*

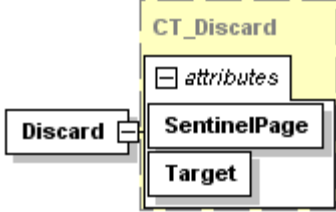
26 **17.1.4.1.1 <DiscardControl> Element**

27 element **DiscardControl**

diagram	
annotation	Contains a list of resources that are safe for a consumer to discard.

28 **17.1.4.1.2 <Discard> Element**

29 element **Discard**

diagram	
---------	---

attributes	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.					

1 Parts that can be discarded are identified in a <Discard> element by the Target attribute value,
2 which is expressed as relative to the package root, and by the SentinelPage attribute value,
3 which identifies the first FixedPage part that no longer requires the discarded part. (The
4 processing order for FixedPage parts is implied by the order of <PageContent> element
5 references in the FixedDocument part. Therefore, the value of the SentinelPage attribute is
6 unambiguous.)

7 If either the Target attribute or the SentinelPage attribute contain an invalid reference (refer
8 outside the package), the respective <Discard> element MUST be ignored [M10.7]. If a
9 <Discard> element is encountered where either or both of the Target attribute and SentinelPage
10 attribute identify a part which has not been processed yet (is still unknown), the <Discard>
11 element SHOULD be retained until both parts identified by the Target attribute and SentinelPage
12 attribute have been processed or until the end of the package is reached [S10.9].

13 17.1.5 Interleaving Optimizations and Digital Signatures

14 In general, it is not feasible to produce well-ordered, interleaved ZIP packages *and* apply digital
15 signatures in a way that enables reasonable consumption scenarios for the following reasons:

- 16 • The digital signature parts must be known to consumers before they process other signed
17 parts because the selected hash-methods and transforms must be known. A streaming
18 consumer might not be able to access part data after it has been processed for printing.
- 19 • Producers cannot create the digital signature parts before producing the signed
20 packages.
- 21 • There are cyclic dependencies with signed relationship parts containing the relationship
22 to the signature parts themselves.

23 Therefore, when adding a digital signature to an interleaved package, producers of digitally
24 signed documents that are intended for streaming consumption SHOULD add all digital
25 signature parts and the package relationship to the digital signature parts at the beginning of
26 the package, before adding any other part [S10.10].

27 17.2 Digital Signatures

28 The digital signature specification for XPS Documents is described in the OPC Standard. It
29 allows users to sign arbitrary parts, relationship parts, and individual relationships. Although
30 XPS Documents also use these digital signature mechanisms, they have a specific signature
31 policy and a specific signing mechanism for documents containing co-signing requests.

17.2.1 Signature Policy

This Standard defines the signature policy that governs the methods of signing and verifying signatures for XPS Documents. The XPS Document signature policy includes a specific set of signing rules and validity rules. All producers and consumers signing and verifying signatures for end users or applications MUST adhere to these rules consistently [M10.8] to ensure that end users can rely on applications to display accurate signature information.

When signing a document, users can choose to make any of the following actions invalidate the signature:

- Editing core properties
- Adding signatures

Consumers MUST NOT prevent an end user from taking an action solely because doing so will invalidate an existing signature [M10.9]. Consumers SHOULD, however, inform the end user if an action they are going to take will invalidate an existing signature [S10.11].

17.2.1.1 Signing Rules

An XPS Document MUST be considered signed according to the XPS Document signing policy, regardless of the validity of that signature, if the following *signing rules* are followed [M10.10]:

1. The following parts MUST be signed [M10.10]:
 - a. The <SignedInfo> portion of the Digital Signature XML Signature part containing this signature.
 - b. The FixedDocumentSequence part that is the target of the Start Part package relationship.
 - c. All FixedDocument parts referenced in the markup of the FixedDocumentSequence part. (Adding a FixedDocument part to a signed XPS Document will invalidate the signature.)
 - d. All FixedPage parts referenced by all signed FixedDocument parts.
 - e. All parts associated with each signed FixedPage part by means of a Required Resource relationship (such as fonts, images, color profiles, remote resource dictionaries).
 - f. All DocumentStructure parts associated via a Document Structure relationship with all signed FixedDocument parts.
 - g. All StoryFragments parts associated via Story Fragments relationship with all signed FixedPage parts.
 - h. All SignatureDefinitions parts associated via a Signature Definitions relationship with any signed FixedDocument part. (Once a document is signed, adding any new signature definitions will invalidate the signature.)
 - i. All Thumbnail parts associated via a Thumbnail relationship from the package root or with any signed FixedPage or FixedDocument part.
2. The following parts MAY be signed [[O10.16](#)~~M10.10~~]:
 - a. The CoreProperties part.
 - b. The Digital Signature Origin part.
 - c. A Digital Signature Certificate part.
 - d. PrintTicket parts.

- 1 e. DiscardControl parts.
- 2 3. All relationships with the following RelationshipTypes (see §A) MUST be signed [M10.10]:
- 3 a. StartPart relationship from the package root
- 4 b. DocumentStructure relationship from a FixedDocument part
- 5 c. StoryFragments relationship from a FixedPage part
- 6 d. Digital Signature Definitions relationship from a FixedDocument part
- 7 e. Required Resource relationship from a FixedPage part
- 8 f. Restricted Font relationship from a FixedDocument part
- 9 g. Thumbnail relationship from a FixedPage part, a FixedDocument part, or the package
- 10 root
- 11 4. All relationships with the following RelationshipTypes MUST be signed if their Target part
- 12 is signed [M10.10]:
- 13 a. Core Properties relationship
- 14 b. Digital Signature Origin relationship
- 15 c. Digital Signature Certificate relationship from a Digital Signature XML Signature part
- 16 d. PrintTicket relationship
- 17 e. DiscardControl relationship
- 18 5. Relationships with the following RelationshipTypes MAY be signed as a group (they MUST
- 19 NOT be signed individually) [M10.10]:
- 20 a. All Digital Signature XML Signature relationships from the Digital Signature Origin part
- 21 (signing all relationships of this RelationshipType will cause this signature to break
- 22 when a new signature is added).
- 23 6. All of the above-referenced parts and relationships MUST be signed using a single digital
- 24 signature [M10.10].

25 An XPS Document MUST NOT be considered signed according to the XPS Document signing
26 policy if [M10.11]:

- 27 1. Any part not covered by the signing rules above is included in the signature.
- 28 2. Any relationship not covered by the signing rules above is included in the signature.

29 An XPS Document digital signer MUST NOT sign an XPS Document that contains content (parts
30 or relationships parts) to be signed that defines the Markup Compatibility namespace when the
31 signer does not fully understand all elements, attributes, and alternate content representations
32 introduced through the markup compatibility mechanisms [M10.12]. An XPS Document digital
33 signer MAY choose not to sign any content (parts or relationships parts) that defines the
34 Markup Compatibility namespace, even when the content is fully understood [O10.8].

35 [An XPS Document digital signer MUST NOT sign a PrintTicket part if it does not fully understand](#)
36 [the PrintTicket content \[M10.25\].](#)

17.2.1.2 Signing Validity

2 An XPS Document digital signature MUST be shown as an *incompliant digital signature* if
3 [M10.13]:

- 4 • It violates any of the signing rules described above regarding parts or relationships that
5 MUST or MUST NOT be signed.

6 An XPS Document digital signature MUST be shown as a *broken digital signature* if [M10.14]:

- 7 • It is not an incompliant digital signature, but the signature fails the signature validation
8 routines described in the OPC.

9 An XPS Document digital signature MUST be shown as a *questionable digital signature* if any of
10 the following are true [M10.15]:

- 11 • It is not an incompliant or broken digital signature, but the certificate cannot be
12 authenticated against the certificate authority.
- 13 • It is not an incompliant or broken digital signature, but the signed content (parts and
14 relationships) contain elements or attributes from an unknown namespace introduced
15 through the Markup Compatibility mechanisms.

16 An XPS Document digital signature MAY be shown as a questionable digital signature if [O10.9]:

- 17 • It is not an incompliant or broken digital signature, but contains some other detectable
18 problem at the discretion of the consumer.

19 An XPS Document digital signature MUST be shown as a *valid digital signature* if [M10.16]:

- 20 • It is not an incompliant, broken, or questionable digital signature.

21 17.2.1.3 Adding Signatures

22 XPS Documents MAY be signed more than once [O10.10]. A user who signs an XPS Document
23 might or might not want to allow any additional signing of the document. To prohibit additional
24 signatures in an XPS Document, the signing application MUST sign all the Digital Signature
25 Origin part's relationships of relationship type Digital Signature with the same signature as the
26 rest of the content [M10.17].

27 17.2.1.4 Certificate Store

28 XPS Document signatures MUST NOT refer to a remote certificate store (certificate not
29 contained in the XPS Document). All certificates MUST be stored in the XPS Document either as
30 a Certificate part or in the Digital Signature XML Signature part [M10.18].

31 17.2.1.5 Printing Signed Documents

32 [Consumers that support printing of signed documents SHOULD support control through](#)
33 [PrintTicket settings pertaining to the treatment of XPS Documents with invalid or questionable](#)
34 [signatures \[S10.22\].](#)

35 [This setting can specify behaviors such as:](#)

- 36 [1. Print the job regardless of the validity of the digital signatures. Digital signatures can be](#)
37 [ignored.](#)
- 38 [2. Print the job regardless of the validity of the digital signatures. In the event an invalid](#)
39 [signature is encountered, an error page should print at the end of the job. Digital signatures](#)
40 [cannot be ignored.](#)

3. Print the job only if all digital signatures are valid. Digital signatures cannot be ignored.

~~When printing signed documents, the PrintTicket setting JobDigitalSignatureProcessing SHOULD be used to control the digital signature processing behavior [S10.12]. Producers MAY include the JobDigitalSignatureProcessing setting in the job level PrintTicket within the XPS Document content [O10.11]. Consumers SHOULD process this PrintTicket setting, if present [S10.12]. For more information, see the Print Schema specification.~~

~~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.

17.2.2 Signature Definitions

In some workflow scenarios, documents must be signed as a means of approving their content. [Example: Document producers might be required to sign their documents in order to provide proof of authenticity. end example] In other cases, reviewers might be required to co-sign content before it can be submitted for publication. These requirements can be fulfilled with a digitally signed XPS Document.

Whereas the XPS package model supports the signing of arbitrary content in a package, an XPS Document signing workflow requires additional features, including the ability to specify co-signature requirements and to include workflow-specific signature information in the document. XPS Document authors and signing parties provide such information in an XML *signature definition*.

Signature definitions are represented by <SignatureDefinition> elements within a single <SignatureDefinitions> element.

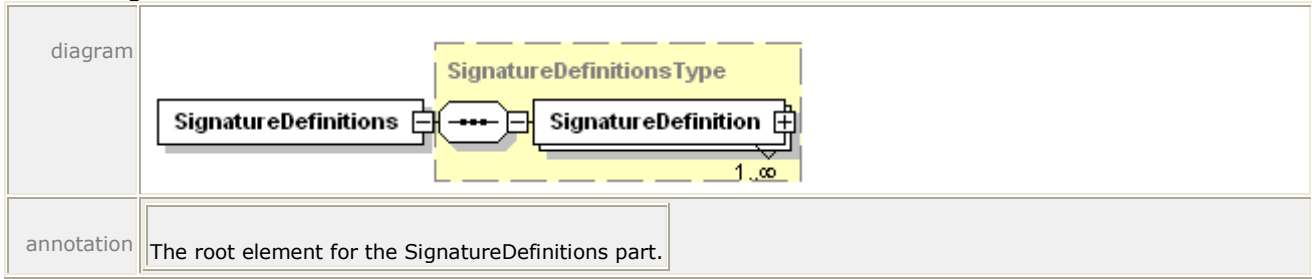
Example 17-4. A SignatureDefinitions part

```
<SignatureDefinitions xmlns="http://schemas.microsoft.com/xps/2005/06/
signature-definitions">
  <SignatureDefinition SignerName="Dorena Paschke"
    SpotID="0e0a7abb-48c9-595d-77db-305e84a05fc3">
    <SpotLocation
      PageURI="/Documents/1/Pages/2.fpage"
      StartX="0.0"
      StartY="0.0" />
    <Intent>I have read and agree</Intent>
    <SignBy>2005-08-20T23:59:59Z</SignBy>
    <SigningLocation>Redmond, WA</SigningLocation>
  </SignatureDefinition>
</SignatureDefinitions>
```

end example]

1 **17.2.2.1 <SignatureDefinitions> Element**

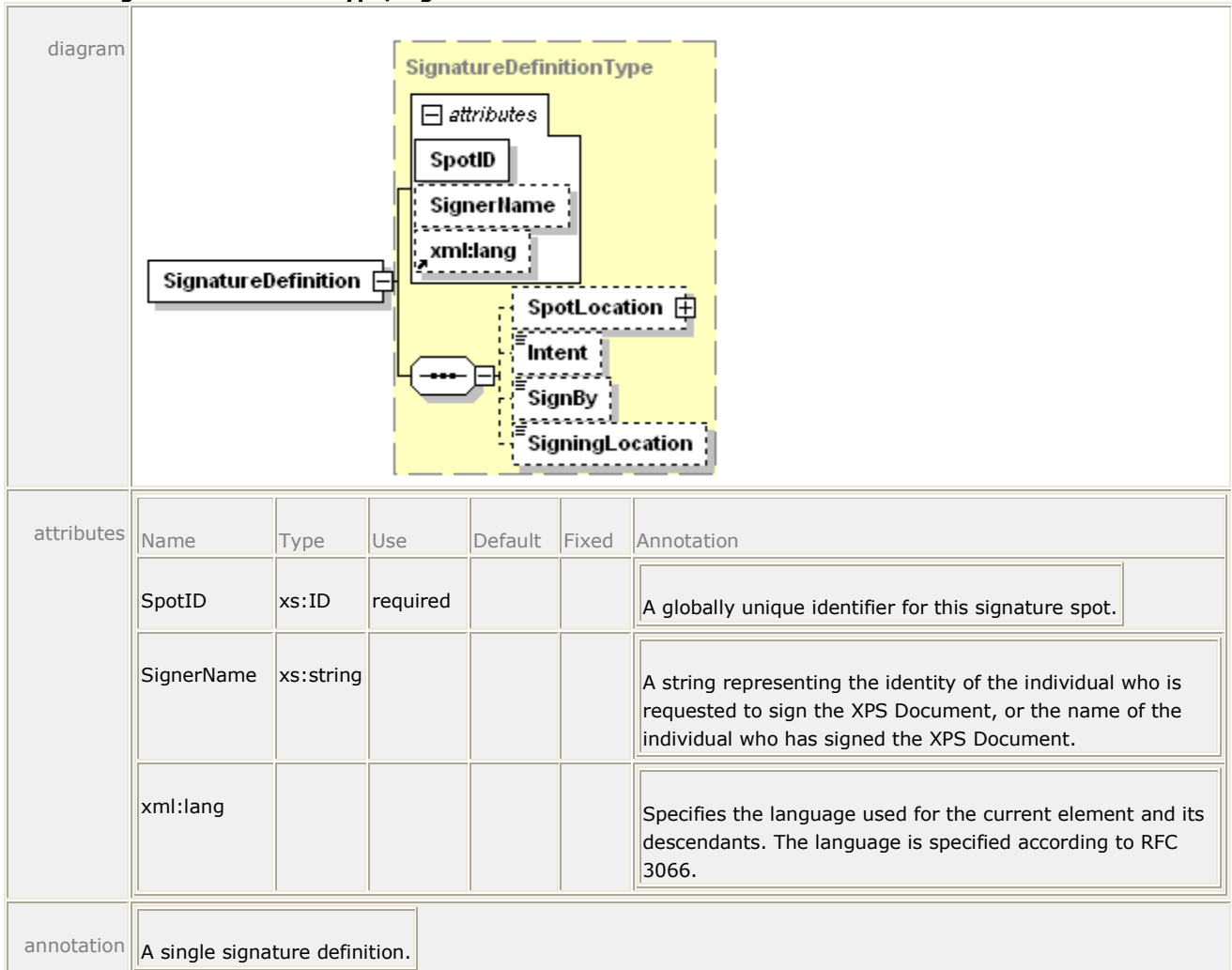
2 element **SignatureDefinitions**



3 If the SignatureDefinitions part exists, it MUST contain only one <SignatureDefinitions>
 4 element [M10.26M2.72]. The XML namespace for the <SignatureDefinitions> element is
 5 specified in §D.1.

6 **17.2.2.2 <SignatureDefinition> Element**

7 element **SignatureDefinitionsType/SignatureDefinition**



8 If the SignatureDefinitions part exists, there MUST be *at least* one <SignatureDefinition>
 9 element [M10.27M2.72].

1 **17.2.2.2.1 SpotID Attribute**

2 The SpotID attribute is REQUIRED [M2.72]. This attribute MAY be used to link an existing
 3 signature to the <SignatureDefinition> element [O10.12]. The value of this attribute MUST be
 4 globally unique to ensure that a Signature part can be linked to only one <SignatureDefinition>
 5 element [M10.29M2.72]. To link a <SignatureDefinition> to a signature, the value of the SpotID
 6 MUST be specified in the Id attribute of the corresponding <Signature> element in the Digital
 7 Signature XML Signature part [M10.19]. For more information, see “Digital Signatures” in the
 8 OPC Standard.

9 **17.2.2.3 <SpotLocation> Element**

10 element **SignatureDefinitionType/SpotLocation**

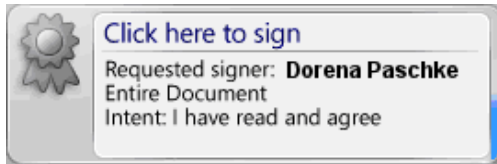
diagram						
attributes	Name	Type	Use	Default	Fixed	Annotation
	PageURI	xs:anyURI	required			Specifies the page on which the signature spot should be displayed.
	StartX	xs:double	required			Specifies the x coordinate of the origin point (upper-left corner) on the page where the signature spot should 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.					

11 The <SpotLocation> element is OPTIONAL [O10.15]. It specifies where an XPS Document
 12 viewer should place a visual representation or *signature spot* to indicate that a digital signature
 13 has been applied or requested. The viewing consumer SHOULD use the values specified in this
 14 element [O10.15]. Due to space and rendering limitations, producers MUST NOT assume that
 15 consumers will use these values [M10.20]. If the location specified by this element is not used,
 16 it is RECOMMENDED that consumers choose a location that does not contain any page content
 17 [S10.13].

18 The size and shape of the signature spot are determined by the consumer. Consumers MAY
 19 choose a size and shape based on the desired display information and page content [O10.13].
 20 However, it is RECOMMENDED that they render signature spots as consistently sized rectangles
 21 that include the signer name, the intent, the signing location, and the scope of the XPS


- 1 Document to be signed [S10.14]. It is also RECOMMENDED that the signature spot be a
 2 clickable area used to launch the digital signing process [S10.15].

3 *Figure 17-1. A sample signature spot*




4
 5 **17.2.2.4 <Intent> Element**

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 Consumers MUST display the full value of the <Intent> element to the signing party, either in
 8 the signature spot or through some other mechanism [M10.21].
- 9 [Note: Consumers that wish to display signature spots must consider the implications of
 10 supporting any Unicode character that can be specified in the <Intent> element, and of the
 11 possibility of Unicode non-characters being included. They must also make decisions about the
 12 appropriate font face and size to use as well as determine the proper layout and interactivity of
 13 the signature spot. In the interests of maximizing compatibility, creators are recommended to
 14 normalize the string using NFC. These decisions are implementation-defined. *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 using this
 4 particular signature spot after the date and time specified [S10.16]. The date and time MUST
 5 be specified in UTC time, using the format "Complete date plus hours, minutes and seconds"
 6 described in the W3C Note "Date and Time Formats" [M10.22], [*Example*: "2006-12-
 7 31T23:59:59Z" for 11:59 PM (UTC) on December 31, 2006. *end example*]

8 **17.2.2.6 <SigningLocation> Element**9 element **SignatureDefinitionType/SigningLocation**

diagram	
annotation	The legal location where the document is signed.

10 The <SigningLocation> element MAY be set by the original producer of the XPS Document or by
 11 the signing party at the time of signing [O10.14].

12 **17.3 Core Properties**

13 XPS Documents use the Core Properties part described in the OPC. The core properties
 14 specified in that part SHOULD refer to the entire fixed payload, including the root
 15 FixedDocumentSequence part and the compilation of all FixedDocument parts it references
 16 [S10.17].

18. Rendering Rules

The set of rules described here ensures precise and consistent rendering of XPS Document markup across various implementations. Producers MUST generate XPS Documents that can be accurately rendered by following the rules described in this clause [M11.1]. Consumers MUST adhere to the rules described in this clause when rendering XPS Documents [M11.1]. In addition to rules for visual elements, implementation limits are also discussed.

18.1 Coordinate System and Rendering Placement

In the x,y coordinate system, one unit is initially equal to 1/96", expressed as a real number. The initial origin of the coordinate system is the top left corner of the fixed page. The x -coordinate value increases from left to right; the y -coordinate value increases from top to bottom.

A RenderTransform property can be specified on any path, glyphs, or canvas to apply an affine transform to the current coordinate system.

A Transform property can be specified on any visual brush, image brush, linear gradient brush, radial gradient brush, or path geometry to apply an affine transform to the current coordinate system.

18.1.1 Page Dimensions

The logical page dimensions correspond to the [media-page](#) size specified in the application page layout and are specified by the Width and Height attributes of the <FixedPage> element. Further optional attributes on the <FixedPage> element are used to specify details about the areas of the fixed page that contain rendered content. For more information, see §10.3.

~~The physical page dimensions correspond to the media size specified for printing. The physical page dimensions are specified in the PrintTicket PageMediaSize keyword.~~

~~The interaction of the logical page dimensions and the physical page dimensions is print-specific. For more information, see §10.3.3 and §10.3.4.~~

18.1.2 Rounding of Coordinates

All computations on coordinate values SHOULD be performed with at least single floating-point precision [S11.1]. 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 [S11.1].

Very high resolution devices MAY use lower fractional precision to represent device coordinates [O11.1].

When converting from real-number coordinate values to device coordinate values, rounding is performed according to the following rule:

$$\text{coord}_d = \text{ROUND}(\text{coord}_r * 16.0) / 16$$

Where coord_r expresses a real-number coordinate value and coord_d expresses a device coordinate value.

1 **18.1.3 Transforms**

2 XPS Document markup supports affine transforms as expressed through the `RenderTransform`
3 and `Transform` properties. An affine transform is represented as a list of six real numbers: `m11`,
4 `m12`, `m21`, `m22`, `dx`, `dy`. (For markup details, see §14.4.)

5 The full matrix is as follows:

$$\begin{bmatrix} m11 & m12 & 0 \\ m21 & m22 & 0 \\ dx & dy & 1 \end{bmatrix}$$

6 A given x,y coordinate is transformed with a render transform to yield the resulting coordinate
7 x',y' by applying the following computations:

$$\begin{aligned} 8 \quad x' &= x * m11 + y * m21 + dx \\ 9 \quad y' &= x * m12 + y * m22 + dy \end{aligned}$$

10 When rendering a child or descendant element, the effective transform used for rendering is the
11 concatenation of all the transforms specified by the `RenderTransform` or `Transform` property on
12 parent or ancestor elements, starting from the outermost ancestor.

13 Non-invertible effective transforms can be specified in markup or occur as a result of limited
14 numerical precision during concatenation. If a non-invertible transform is encountered during
15 rendering, consumers MUST omit rendering the affected element and all of its child and
16 descendant elements [M11.2].

17 If a non-invertible transform is encountered on a brush (as specified directly on the brush, as a
18 result of the `Viewbox` or `Viewport` attributes, or through concatenation), the brush is treated
19 according to §18.7.1.

20 The `Width` and `Height` values specified in the `Viewbox` and `Viewport` attributes of an
21 `<ImageBrush>` or `<VisualBrush>` element MUST NOT be negative [M11.10M2.72].

22 If a non-invertible transform is encountered on a geometry (as specified directly on the
23 geometry or through concatenation), the geometry MUST be considered to contain no area
24 [M11.3].

25 A final, device-dependent step using the horizontal resolution and vertical resolution of the
26 device converts the resulting coordinates x',y' to device coordinates x'',y'' , as follows:

$$\begin{aligned} 27 \quad x'' &= x' * R_x/96 \\ 28 \quad y'' &= y' * R_y/96 \end{aligned}$$

29 Where R_x is the horizontal resolution and R_y is the vertical resolution of the device, specified in
30 device pixels per inch.

31 **18.1.4 Pixel Center Location, Pixel Placement, and Pixel Inclusion**

32 A pixel covers the range from x to $x+1$.

33 An *ideal* consumer implementation SHOULD render pixels in an 8x8 sub-pixel space, perform an
34 8x8 box filter sampling, and set the pixel to the resulting color value [S11.2]. Other
35 implementations MAY use different rendering logic as long as it closely approximates this logic
36 [O11.2].

1 When rendering a shape, a *practical* implementation (such as a bi-tonal printing device)
2 SHOULD turn on each pixel whose center (at $x+0.5$) is covered by the shape, or is touched by
3 the shape with the shape extending beyond the pixel center in the positive x or y direction of
4 the device [S11.3]. Devices MAY use sub-pixel masking instead [O11.3].

5 By definition, a shape with an area width of 0 (that is, no included area) does not touch or
6 cover any pixel centers. A stroke with a width of 0 is treated in the same manner.

7 As a result of these rules, the behavior for very thin lines is implementation-defined:

- 8 • An implementation capable of anti-aliasing MAY draw a thin line in a way that blends with
9 the background to varying degrees [O11.4].
- 10 • A bi-tonal implementation on a printer MAY draw thin lines, or apply half-toning,
11 depending on the desired output quality [O11.5]. If such an implementation chooses to
12 draw thin lines, then it MAY choose to draw them with drop outs, following requirement
13 S11.3 in §18.1.4 above, or as solid rules of 1 pixel thickness [O11.26].

14 [Note: Also see §18.6.12 for discussion of thin strokes. *end note*]

15 **18.1.5 Maximum Placement Error**

16 When rendering geometries, consumers SHOULD render curves so they appear smooth from a
17 normal viewing distance [S11.4]. Producers MUST NOT assume a specific placement error for
18 curve decomposition or rely on side-effects of a specific consumer implementation [M11.4].

19 **18.1.6 Pixel Placement for Glyphs**

20 Regardless of other rules expressed here, consumers MAY apply pixel placement rules
21 optimized for character rendering to individual glyphs in a <Glyphs> element [O11.6]. Such
22 rules can result from font hinting applied by the typeface scaler used by a consumer
23 implementation.

24 **18.1.7 Abutment of Shapes**

25 When no anti-aliasing is used, abutting shapes that share the same device coordinates for the
26 end-points and control-points of an edge SHOULD be rendered without overlap and without
27 gaps [S11.5]. Ideally, an implementation SHOULD also follow this rule for shapes that are
28 mathematically abutting without sharing device coordinates for end-points and control-points of
29 edges [S11.5].

30 **18.1.8 Clipping Behavior**

31 Clipping occurs as if a mask were created from the clip geometry according to the pixel
32 placement rules defined in §18.1.4. An ideal consumer SHOULD create such a mask in an 8x8
33 sub-pixel space and subsequently draw only those sub-pixels of a shape that correspond to
34 "ON" sub-pixels in the mask [S11.6].

35 A practical implementation (such as a bi-tonal printing device) SHOULD create a pixel mask
36 according to Point2 of §18.1.4, and subsequently draw only those pixels of a shape that
37 correspond to "ON" pixels in the mask. In creating the mask and drawing the shape, the
38 abutment of shapes rule SHOULD be observed so that no pixel of the shape is drawn that would
39 not have been drawn if the clip geometry were another abutting shape [S11.7]. Devices MAY
40 use sub-pixel masking instead [O11.3].

1 18.2 Implementation Limits

2 XPS Document markup does not assume fixed implementation limits. However, consumers can
3 have specific implementation limits imposed by their operating environment. XPS Document
4 markup has been designed so that even complex pages can be represented accurately and with
5 high fidelity.

6 A typical consumer implementation SHOULD be able to process markup with the characteristics
7 indicated in Table 18–1 [S11.8]. ~~If a consumer encounters~~ ~~Encountering~~ markup with
8 characteristics outside ~~its of the consumer-specific~~ ~~implementation-defined~~ limits, ~~it~~ MUST
9 ~~cause-instantiate~~ an error condition [M11.5].

10 Table 18–1 provides the RECOMMENDED minimum requirements for individual elements
11 [S11.8]. Consumers also have limits on the total number of elements, as imposed by available
12 memory. Producers SHOULD produce only XPS Documents that stay within these
13 implementation limits [S11.8].

14 In order to process pages that contain a large number of elements, consumers MAY implement
15 support for the DiscardControl part in order to discard elements that have already been
16 processed [O10.5].

17 *Table 18–1. Recommended minimum processing requirements*

Characteristic	Type	Limit	Description
Coordinates/transformation matrix elements	Real number	+/- 10 ¹²	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 ⁻¹²	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 coordinates		Single floating point	Coordinates are real numbers and SHOULD be computed with at least single floating point precision [S11.1].
Nested <Canvas> elements		16	Depth of nested <Canvas> elements.
Nested <VisualBrush> elements		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	

Characteristic	Type	Limit	Description
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.
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 A.1), 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 rendered
3 according to the guidelines described in this subclause [S11.11].

4 18.3.1 All Gradients

5 Linear gradients and radial gradients share a common set of recommended operations for pre-
6 processing gradient stops and blending colors. These are described below.

7 18.3.1.1 Gradient Stop Pre-Processing

8 Consumers SHOULD pre-process gradient stops for all gradients using the following steps
9 [S11.12]:

- 10 1. Sort all gradient stops by their respective offset values in ascending order. When two or
11 more gradient stops have the same offset value, preserve their relative order from the
12 markup while sorting. When more than two gradient stops have the same offset value,
13 remove all but the first and last gradient stops having the same offset value.
- 14 2. If no gradient stop with an offset of 0.0 exists,

- 1 a. And no gradient stop with an offset less than 0.0 exists, create an artificial gradient
2 stop having an offset of 0.0 and a color of the gradient stop with the smallest offset
3 value.
- 4 b. And a gradient stop with an offset less than 0.0 exists and a gradient stop with an
5 offset greater than 0.0 exists, create an artificial gradient stop having an offset of 0.0
6 and a color interpolated between the two gradient stops surrounding 0.0. Discard all
7 gradient stops with an offset less than 0.0.
- 8 c. And a gradient stop with an offset less than 0.0 exists and no gradient stop with an
9 offset greater than 0.0 exists, create an artificial gradient stop having an offset of 0.0
10 and a color of the gradient stop with the largest offset value. Discard all gradient stop
11 elements with an offset less than 0.0.
- 12 3. If no gradient stop with an offset of 1.0 exists,
 - 13 a. And no gradient stop with an offset of greater than 1.0 exists, create an artificial
14 gradient stop having an offset of 1.0 and a color equal to the color of the gradient stop
15 with the largest offset value.
 - 16 b. And a gradient stop with an offset greater than 1.0 exists and a gradient stop with an
17 offset less than 1.0 exists, create an artificial gradient stop having an offset of 1.0 and
18 a color interpolated between the two surrounding gradient stops. Discard all gradient
19 stops with an offset greater than 1.0.
 - 20 c. And a gradient stop with an offset greater than 1.0 exists and no gradient stop with an
21 offset less than 1.0 exists, create an artificial gradient stop having an offset of 1.0 and
22 a color equal to that of the gradient stop with the smallest offset value. Discard all
23 gradient stops with an offset greater than 1.0.

24 18.3.1.2 Blending Colors

25 If any gradient stops use an sRGB or scRGB color specification ~~or the consumer does not~~
26 ~~understand the PageBlendColorSpace-PrintTicket setting~~, consumers SHOULD blend colors
27 between gradient stops in the color space indicated by the ColorInterpolationMode attribute of the
28 gradient brush, unless a PrintTicket setting provides an alternative blending color space that the
29 consumer understands (see §9.1.9 and §15.5) [S11.13]. If none of the gradient stop elements
30 uses an sRGB or scRGB color specification and the consumer understands the blending color
31 space PageBlendColorSpace-PrintTicket setting, the blending color space PageBlendColorSpace
32 PrintTicket setting SHOULD be used [S11.13].

33 The function used for blending is:

34 $\text{BLEND}(\text{offset}, c_{lo}, c_{hi})$

35 Where the offset is between 0 and 1. c_{lo} and c_{hi} designate the color values for an offset of -0
36 and -1, respectively.

37 If a ColorInterpolationMode value of SRgbLinearInterpolation is used, the BLEND() function
38 SHOULD convert the color values to sRGB first, and then perform a linear interpolation between
39 them [S11.14].

40 If a ColorInterpolationMode value of ScRgbLinearInterpolation is used, the BLEND() function
41 SHOULD convert the color values to scRGB first, and then perform a linear interpolation
42 between them [S11.15].

43 ~~If blending is performed in the color space identified by the PageBlendColorSpace-PrintTicket~~
44 ~~setting, it SHOULD be a linear, channel-by-channel blend operation [S11.13].~~

1 In the presence of transformations or when individual gradient stops are very close (separated
 2 by a few pixels or less in the device space), the local color gradient at the offset used in the
 3 BLEND() function might be large, resulting in a large change over the extent of a single device
 4 pixel. In this case, it is RECOMMENDED that the BLEND() function interpolate the gradient over
 5 the extent of each device pixel [S11.16]. However, the behavior MAY differ from this
 6 recommendation in an implementation-defined manner [O11.7] and, therefore, producers
 7 SHOULD NOT rely on a specific effect for such dense gradient specifications [S11.16].

8 As a consequence of this interpolation, radial gradients that define the gradient origin on or
 9 outside ellipse create an “outside” area that can be rendered inconsistently. The radial
 10 gradients that are affected are those that define multiple gradient stops that are of different
 11 colors and are very close in Offset value to 0.0 or 1.0 (the gradient end points), for radial
 12 gradients with a SpreadMethod value of Repeat or Reflect, respectively. For these affected
 13 gradients, consumers MAY use an interpolated color value for the outside area [O11.8].
 14 Depending on the resolution, this can result in different colors than those defined by the
 15 gradient end points. The closer a gradient stop is to the affected gradient end point, the more
 16 the rendering results on different consumers and at different display resolutions can differ.
 17 Producers SHOULD therefore either avoid such close gradient stops to the gradient end point
 18 when specifying radial gradients where the outside area is visible or avoid specifying radial
 19 gradients with a gradient origin on or outside the ellipse (in which case there is no outside area)
 20 to ensure consistent rendering results [S11.17].

21 18.3.2 Linear Gradients

22 Consumers SHOULD render an element filled with a linear gradient brush such that the
 23 appearance is the same as if the following steps had been taken [S11.33S11.11]:

- 24 1. Transform the StartPoint and EndPoint attribute values using the current effective render
 25 transform (including the render transform for the element being filled by the linear
 26 gradient brush and the brush’s transform itself).
- 27 2. If the SpreadMethod value is Pad, the colors of points on the line defined by the StartPoint
 28 and EndPoint attributes are defined by interpolating the coordinates linearly, and each
 29 color component (such as R, G, B for sRGB and scRGB) as well as the alpha component
 30 is interpolated between the component values of the closest enclosing gradient stops:

```
31 For each offset (real number)  $t < 0$ :
32 {
33    $x(t) = (EndPoint_x - StartPoint_x) * t + StartPoint_x$ 
34    $y(t) = (EndPoint_y - StartPoint_y) * t + StartPoint_y$ 
35    $c(t) = c_{first}$ 
36    $a(t) = a_{first}$ 
37 }
```

38 Where c is the color component and a is the alpha component. c_{first} are the color
 39 component values of the first gradient stop (after sorting) and a_{first} is the alpha
 40 component value at the first gradient stop (after sorting).

```
41 For each offset (real number)  $0 \leq t \leq 1$ :
42 {
43    $x(t) = (EndPoint_x - StartPoint_x) * t + StartPoint_x$ 
44    $y(t) = (EndPoint_y - StartPoint_y) * t + StartPoint_y$ 
45    $c(t) = BLEND((t - t_{10}) / (t_{hi} - t_{10}), c_{10}, c_{hi})$ 
46    $a(t) = [(t - t_{10}) / (t_{hi} - t_{10})] * (a_{hi} - a_{10}) + a_{10}$ 
47 }
```

1 Where t_{lo} and t_{hi} are the offsets, c_{lo} and c_{hi} are the color component values at the
 2 closest enclosing gradient stops (that is, $t_{lo} \leq t \leq t_{hi}$) and a_{lo} and a_{hi} are the alpha
 3 component values at the closest enclosing gradient stops ($t_{lo} \leq t \leq t_{hi}$).

4 For each offset (real number) $t > 1$:
 5 {
 6 $x(t) = (EndPoint_x - StartPoint_x) * t + StartPoint_x$
 7 $y(t) = (EndPoint_y - StartPoint_y) * t + StartPoint_y$
 8 $c(t) = c_{last}$
 9 $a(t) = a_{last}$
 10 }

11 Where c_{last} are the color component values of the last gradient stop (after sorting) and
 12 a_{last} is the alpha component value at the last gradient stop (after sorting).

13 3. If the SpreadMethod value is Repeat, the colors of points on the line defined by the
 14 StartPoint and EndPoint attributes are defined by interpolating the coordinates linearly, and
 15 each color component (such as R, G, B for sRGB and scRGB) as well as the alpha
 16 component is interpolated between the component values of the closest enclosing
 17 gradient stops:

18 For each repetition (all integers) N :
 19 {
 20 For each offset (real number) $0 \leq t < 1$:
 21 {
 22 $x(t) = (EndPoint_x - StartPoint_x) * (N+t) + StartPoint_x$
 23 $y(t) = (EndPoint_y - StartPoint_y) * (N+t) + StartPoint_y$
 24 $c(t) = BLEND((t-t_{lo})/(t_{hi}-t_{lo}), c_{lo}, c_{hi})$
 25 $a(t) = [(t-t_{lo})/(t_{hi}-t_{lo})] * (a_{hi}-a_{lo}) + a_{lo}$
 26 }
 27 }

28 Where c is the color component and a is the alpha component. t_{lo} and t_{hi} are the offsets,
 29 c_{lo} and c_{hi} are the color component values at the closest enclosing gradient stops (that is,
 30 $t_{lo} \leq t \leq t_{hi}$) and a_{lo} and a_{hi} are the alpha component values at the closest enclosing
 31 gradient stops ($t_{lo} \leq t \leq t_{hi}$).

32 4. If the SpreadMethod value is Reflect, the colors of points on the line defined by the
 33 StartPoint and EndPoint attributes are defined by interpolating the coordinates linearly, and
 34 each color component (such as R, G, B for sRGB and scRGB) as well as the alpha
 35 component is interpolated between the component values of the closest enclosing
 36 gradient stops:

37 For each repetition (all integers) N :
 38 {
 39 For each offset (real number) $0 \leq t \leq 1$:
 40 {
 41 If (N is EVEN)
 42 {
 43 $x(t) = (EndPoint_x - StartPoint_x) * (N+t) + StartPoint_x$
 44 $y(t) = (EndPoint_y - StartPoint_y) * (N+t) + StartPoint_y$
 45 }
 46 Else
 47 {
 48 $x(t) = (EndPoint_x - StartPoint_x) * (N+1-t) + StartPoint_x$
 49 $y(t) = (EndPoint_y - StartPoint_y) * (N+1-t) + StartPoint_y$
 50 }

```

1      c(t) = BLEND((t-t1o)/(thi-t1o), c1o, chi)
2      a(t) = [(t-t1o)/(thi-t1o)]*(ahi-a1o)+a1o
3  }
4  }
```

5 Where c is the color component and a is the alpha component. t_{1o} and t_{hi} are the offsets,
6 c_{1o} and c_{hi} are the color component values at the closest enclosing gradient stops (that is,
7 $t_{1o} \leq t \leq t_{hi}$) and a_{1o} and a_{hi} are the alpha component values at the closest enclosing
8 gradient stops ($t_{1o} \leq t \leq t_{hi}$).

- 9 5. The colors of points not on the extended line defined by the StartPoint and EndPoint
10 attributes are the same as the color of the closest point on the line defined by the
11 StartPoint and EndPoint attributes, measured in the coordinate space as transformed by
12 the current effective render transform (including the render transform for the element
13 being filled by the linear gradient brush and the brush's transform itself).
- 14 6. Clip the resulting set of points to the intersection of the current clip geometry and the
15 path or glyphs to be filled. Both the clip and path (or glyphs) must be transformed
16 according to the current effective render transform, including the render transform for
17 the element being filled, but *not* including the transform of the linear gradient brush.

18 For purposes of the above steps, the closest enclosing gradient stops mean the gradient stops
19 that, if the relative sequencing of the gradient stop offsets in the markup order is respected,
20 are numerically closest to the interpolation point if that interpolation point were converted to an
21 offset value and inserted in a sorted fashion into the list of gradient stops. [Example: If a
22 gradient contains three gradient stops at offset values 0.0, 0.0, and 1.0, the closest enclosing
23 gradient stops for any value $0 \leq \text{value} \leq 1$ are the second gradient stop (offset 0.0) and the
24 third gradient stop (offset 1.0). end example]

25 18.3.3 Radial Gradients

26 Consumers SHOULD render an element filled with a radial gradient brush such that the
27 appearance is the same as if these steps had been followed [S11.34S11.11]:

- 28 1. The boundary of the area filled by a radial gradient brush is defined by interpolating
29 ellipses from the GradientOrigin value to the circumference of the ellipse centered at the
30 point specified by the Center attribute with radii equal to the RadiusX and RadiusY
31 attribute values(the interpolated ellipses and point being transformed by the current
32 effective render transform, including the render transform for the element being filled by
33 the radial gradient brush and the brush's transform itself). If the gradient origin is
34 outside the circumference of the ellipse specified, the effect will be as if a cone were
35 drawn, tapering to the gradient origin.
- 36 2. If the SpreadMethod value is Pad, the centers and radii of the interpolated ellipses are
37 defined by linearly interpolating the center of the ellipse from the GradientOrigin attribute
38 value to the Center attribute value, and simultaneously linearly interpolating the radii of
39 the ellipse from 0 to the RadiusX and RadiusY attribute values:

```

40 For each offset (real number)  $0 \leq t \leq 1$ :
41 {
42   cx(t) = (Centerx-GradientOriginx)*t+GradientOriginx
43   cy(t) = (Centery-GradientOriginy)*t+GradientOriginy
44   rx(t) = RadiusX*t
45   ry(t) = RadiusY*t
46 }
```

1 The ellipses defined by the interpolation are transformed by the current effective render
 2 transform, including the render transform for the element being filled by the radial
 3 gradient brush and the brush's transform itself.

4 3. The colors of the points within the boundary of this shape are defined as the color of the
 5 smallest interpolated ellipse containing the point. The color of an interpolated ellipse is
 6 defined by interpolating each color component (such as R, G, B for sRGB and scRGB) as
 7 well as the alpha component between the component values of the closest enclosing
 8 gradient stops:

9 For each offset (real number) $0 \leq t \leq 1$:

```
10 {
11   c(t) = BLEND((t-t10)/(thi-t10), c10, chi)
12   a(t) = [(t-t10)/(thi-t10)]*(ahi-a10)+a10
13 }
```

14 Where t_{10} and t_{hi} are the offsets, c_{10} and c_{hi} are the color component values at the
 15 closest enclosing gradient stops (that is, $t_{10} \leq t \leq t_{hi}$) and a_{10} and a_{hi} are the alpha
 16 component values at the closest enclosing gradient stops ($t_{10} \leq t \leq t_{hi}$).

17 4. If the SpreadMethod value is Repeat, the centers and radii of the interpolated ellipses are
 18 defined by linearly interpolating the center of the ellipse from the GradientOrigin attribute
 19 value to the Center attribute value, and simultaneously linearly interpolating the radii of
 20 the ellipse from 0 to RadiusX and RadiusY attribute values:

21 For each repetition (all non-negative integers) N:

```
22 {
23   For each offset (real number)  $0 \leq t < 1$ :
24   {
25     cx(t) = (Centerx-GradientOriginx)*(N+t)+GradientOriginx
26     cy(t) = (Centery-GradientOriginy)*(N+t)+GradientOriginy
27     rx(t) = RadiusX*(N + t)
28     ry(t) = RadiusY*(N + t)
29   }
30 }
```

31 The ellipses defined by the interpolation are transformed by the current effective render
 32 transform, including the render transform for the element being filled by the radial
 33 gradient brush and the brush's transform itself.

34 5. The colors of the points within the boundary of this shape are defined as the color of the
 35 smallest interpolated ellipse containing the point. The color of an interpolated ellipse is
 36 defined by interpolating each color component (such as R, G, B for sRGB and scRGB) as
 37 well as the alpha component between the component values of the closest enclosing
 38 gradient stops:

39 For each repetition (all non-negative integers) N:

```
40 {
41   For each offset (real number)  $0 \leq t < 1$ :
42   {
43     c(t) = BLEND((t-t10)/(thi-t10), c10, chi)
44     a(t) = [(t-t10)/(thi-t10)]*(ahi-a10)+a10
45   }
46 }
```

47 Where t_{10} and t_{hi} are the offsets, c_{10} and c_{hi} are the color component values at the
 48 closest enclosing gradient stops (that is, $t_{10} \leq t \leq t_{hi}$) and a_{10} and a_{hi} are the alpha
 49 component values at the closest enclosing gradient stops ($t_{10} \leq t \leq t_{hi}$).

- 1 6. If the SpreadMethod value is Reflect, the centers and radii of the interpolated ellipses are
 2 defined by linearly interpolating the center of the ellipse from the GradientOrigin attribute
 3 value to the Center attribute value, and simultaneously linearly interpolating the radii of
 4 the ellipse from 0 to the RadiusX and RadiusY attribute values:

```

5 For each non-negative integer N:
6 {
7   For each offset (real number)  $0 \leq t \leq 1$ :
8   {
9      $c_x(t) = (Center_x - GradientOrigin_x) * (N+t) + GradientOrigin_x$ 
10     $c_y(t) = (Center_y - GradientOrigin_y) * (N+t) + GradientOrigin_y$ 
11     $r_x(t) = RadiusX * (N+t)$ 
12     $r_y(t) = RadiusY * (N+t)$ 
13   }
14 }

```

15 The ellipses defined by the interpolation are transformed by the current effective render
 16 transform, including the render transform for the element being filled by the radial
 17 gradient brush and the brush's transform itself.

- 18 7. The colors of the points within the boundary of this shape are defined as the color of the
 19 smallest interpolated ellipse containing the point. The color of an interpolated ellipse is
 20 defined by interpolating each color component (such as R, G, B for sRGB and scRGB) as
 21 well as the alpha component between the component values of the closest enclosing
 22 gradient stops:

```

23 For each non-negative integer N:
24 {
25   For each offset (real number)  $0 \leq t \leq 1$ :
26   {
27     If N is ODD
28        $t' = 1-t$ 
29     Else
30        $t' = t$ 
31
32      $c(t) = BLEND((t' - t_{lo}) / (t_{hi} - t_{lo}), c_{lo}, c_{hi})$ 
33      $a(t) = [(t' - t_{lo}) / (t_{hi} - t_{lo})] * (a_{hi} - a_{lo}) + a_{lo}$ 
34   }
35 }

```

36 Where t_{lo} and t_{hi} are the offsets, c_{lo} and c_{hi} are the color component values at the
 37 closest enclosing gradient stops (that is, $t_{lo} \leq t \leq t_{hi}$) and a_{lo} and a_{hi} are the alpha
 38 component values at the closest enclosing gradient stops ($t_{lo} \leq t \leq t_{hi}$).

- 39 8. The colors of points outside the boundary of this shape (points which cannot be drawn by
 40 any combination of non-negative N and t) are defined as having the color and alpha
 41 defined in the gradient stop with the offset of 0.0 for radial gradients with a SpreadMethod
 42 value of Reflect and the color and alpha defined in the gradient stop with the offset of 1.0
 43 for radial gradients with a SpreadMethod value of Repeat or Pad. The colors outside of the
 44 boundary of this shape can also vary in an implementation-defined manner
 45 (see §18.3.1.2 for more details).
- 46 9. Clip the resulting set of points by the intersection of the current clip geometry and the
 47 path or glyphs to be filled. Both the clip and path (or glyphs) must be transformed
 48 according to the current effective render transform, including the render transform for
 49 the element being filled, but *not* including the transform of the radial gradient brush.

1 For purposes of the above steps, the closest enclosing gradient stops mean the gradient stops
 2 that, if the relative sequencing of the gradient stop offsets in the markup order is respected,
 3 are numerically closest to the interpolation point if that interpolation point were converted to an
 4 offset value and inserted in a sorted fashion into the list of gradient stops. [*Example*: If a
 5 gradient contains three gradient stops at offset values 0.0, 0.0, and 1.0, the closest enclosing
 6 gradient stops for any value $0 \leq \text{value} \leq 1$ are the second gradient stop (offset 0.0) and the
 7 third gradient stop (offset 1.0). *end example*]

8 **18.4 Opacity Computations**

9 Opacity is used to transparently blend two elements when rendering, also known as alpha
 10 blending. The value of the Opacity property ranges from 0.0 (fully transparent) to 1.0 (fully
 11 opaque), inclusive. Values outside of this range are invalid.

12 The opacity is applied through the following computations, assuming source and destination
 13 values are not pre-multiplied. All opacity calculations SHOULD be performed with at least 8-bit
 14 precision to provide sufficient quality for nested content [S11.18].

15 Individual pixels are blended as defined below.

16 *Table 18–2. Opacity computation symbols*

Symbol	Description
O_E	Opacity attribute of element
O_M	Alpha value at corresponding pixel position in the OpacityMask 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_{R^*}	Resulting Alpha value for destination surface
R_{R^*}	Resulting Red value for destination surface
G_{R^*}	Resulting Green value for destination surface
B_{R^*}	Resulting Blue value for destination surface

17 All values designated with a T subscript (as in R_{T1}) are temporary values.

18 The opacity is calculated as follows:

19 ~~10.1.~~ 10.1. Multiply source alpha value with opacity value and alpha value of opacity mask.

20 $A_{S1} = A_S * O_E * O_M$

21 ~~11.2.~~ 11.2. Pre-multiply source alpha.

1 ~~Omit this step~~ If the source data specifies pre-multiplied alpha (see §18.4.1 for details):
2 $A_{T1} = 0, R_{T1} = R_S, G_{T1} = G_S, B_{T1} = B_S$; otherwise:

3 $A_{T1} = A_{S1}$
4 $R_{T1} = R_S * A_{S1}$
5 $G_{T1} = G_S * A_{S1}$
6 $B_{T1} = B_S * A_{S1}$

7 ~~12.3.~~ Pre-multiply destination alpha.

8 ~~Omit this step~~ If an consumers supports ~~ing~~ superluminous colors (see §18.4.1 for
9 details): $A_{T2} = A_D, R_{T2} = R_D, G_{T2} = G_D, B_{T2} = B_D$; otherwise:

10 $A_{T2} = A_D$
11 $R_{T2} = R_D * A_D$
12 $G_{T2} = G_D * A_D$
13 $B_{T2} = B_D * A_D$

14 ~~13.4.~~ Blend.

15 See §18.4.1 for special case handling.

16 $A_{T3} = (1 - A_{T1}) * A_{T2} + A_{T1}$
17 $R_{T3} = (1 - A_{T1}) * R_{T2} + R_{T1}$
18 $G_{T3} = (1 - A_{T1}) * G_{T2} + G_{T1}$
19 $B_{T3} = (1 - A_{T1}) * B_{T2} + B_{T1}$

20 ~~14.5.~~ Reverse pre-multiplication.

21 ~~Omit this step in consumers supporting superluminous colors. See also §18.4.1.~~

22 The resulting color channel values are divided by the resulting alpha value. If the
23 resulting alpha value is 0, all color channels are set to 0 by definition, as expressed in the
24 If condition below. Each of R_{T3}, G_{T3}, B_{T3} is smaller than or equal to A_{T3} and, therefore, each
25 of the resulting R_R^*, G_R^*, B_R^* is in the valid interval of [0.0,1.0] after the pre-
26 multiplication is reversed.

27 If a consumer supports superluminous colors
28 {
29 $A_R = A_{T3}, R_R = R_{T3}, G_R = G_{T3}, B_R = B_{T3}$
30 }
31 Else If $A_{T3} = 0$
32 {
33 set all $A_R^*, R_R^*, G_R^*, B_R^*$ to 0.
34 }
35 Else
36 {
37 $A_R^* = A_{T3}$
38 $R_R^* = R_{T3} / A_{T3}$
39 $G_R^* = G_{T3} / A_{T3}$
40 $B_R^* = B_{T3} / A_{T3}$
41 }

42 When blending colors in a color space other than sRGB, color channels are independently
43 interpolated in a manner analogous to the RGB channel blending method described above.
44 Colors in subtractive color spaces (such as CMYK) are complemented before and after the
45 blending steps 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 Windows
3 Media Photo images using pixel formats WICPixelFormat32bppPBGRA,
4 WICPixelFormat64bppPRGBA or WICPixelFormat128bppPRGBAFloat MUST be interpreted as
5 pre-multiplied alpha information [M11.6]. [Lex5] In certain scenarios (such as when rendering 3D
6 scenes to a bitmap), producers MAY choose to create pre-multiplied bitmap data specifying
7 "superluminous" colors [O11.9].

8 Superluminous colors are defined as a subset case of the pre-multiplied RGB source color
9 values case with in which the source alpha value is smaller than the individual color channel
10 values but greater than or equal to 0.

11 The effect of composing superluminous colors on a background is similar to adding additional
12 light of the source color to the destination color, as opposed to regular alpha composition which
13 works more like a colored filter. One can easily verify this statement by substituting 0 for A_{T1} in
14 step 4 of the above opacity computations, which is simplified as follows (note that R_{T1}, G_{T1}, B_{T1}
15 are not 0, because the pre-multiplication in step 2 has been skipped):

$$\begin{aligned} 16 \quad A_{T3} &= A_{T2} \\ 17 \quad R_{T3} &= R_{T2} + R_{T1} \\ 18 \quad G_{T3} &= G_{T2} + G_{T1} \\ 19 \quad B_{T3} &= B_{T2} + B_{T1} \end{aligned}$$

20 Consumers supporting superluminous colors retain all temporary information in pre-multiplied
21 formats. Note, that throughout the XPS Standard non-pre-multiplied alpha processing is
22 assumed. It is up to the implementer of such a consumer to identify equivalent composition and
23 rendering rules for processing in pre-multiplied space.

24 Also note, when composing superluminous colors, management of out-of-gamut colors SHOULD
25 be deferred until the result is rendered to the final target, at which point out-of-gamut colors
26 are clipped or color managed [S11.19].

27 Consumers MAY handle superluminous colors or MAY instead choose to convert pre-multiplied
28 source data containing superluminous colors to non-pre-multiplied data before composition by
29 ignoring the superluminous portion of each color channel value [O11.10], as described in the
30 following steps:

```
31 |   For each superluminous pixel with  $A_S < R_S$  or  $A_S < G_S$  or  $A_S < B_S$ 
32 |   {
33 |     If  $A_S = 0$ 
34 |     {
35 |        $A_R = 0$ 
36 |        $R_R^* = 1$ 
37 |        $G_R^* = 1$ 
38 |        $B_R^* = 1$ 
39 |     }
40 |     Else
41 |     {
42 |        $A_R^* = A_S$ 
43 |        $R_R^* = \min(R_S/A_S, 1)$ 
44 |        $G_R^* = \min(G_S/A_S, 1)$ 
45 |        $B_R^* = \min(B_S/A_S, 1)$ 
46 |     }
47 |   }
```


18.5 Composition Rules

XPS Document page markup uses the painter's model with alpha channel. Composition MUST have the same effect as the application of the following rules, in sequence [M11.7]:

~~15.1.~~ In order to render a fixed page or canvas, a surface is created to hold the drawing content as it is composed. The color and appearance of this surface SHOULD match the destination color and appearance, typically a solid white background for a fixed page or transparent for a canvas [S11.20]. An implementation MAY choose to meet this goal by always initializing this surface's alpha channel to 0.0 (transparent) and the color value to black [O11.5].

~~16.2.~~ The fixed page or canvas represents a surface onto which child elements are drawn. The child elements are drawn in the order they appear in markup. In practice, an implementation might represent the surface by a bitmap buffer large enough to hold all the drawing content produced when the child elements are rendered.

~~17.3.~~ The contents appearing on the surface of canvas are transformed using the affine transform specified by the RenderTransform property of the canvas. (A fixed page does not have a RenderTransform property.)

~~18.4.~~ All child elements are rendered to the surface and clipped to the imageable area of the physical display (such as a sheet of paper) of the fixed page or according to the Clip property of a canvas. The geometry value of the canvas' Clip property is also transformed using the affine transform specified by the RenderTransform property of the canvas.

~~19.5.~~ If a path has a Stroke and a Fill property, and also specifies Opacity or OpacityMask property values, additional composition steps must be followed:

- a. Create a temporary canvas with the opacity, opacity mask, clip, and render transform specified by the path.
- b. Create a copy of the original path, remove all but the Fill property from the copy, and add the copy to the temporary canvas.
- c. Create another copy of the original path, remove all but the stroke-related properties (such as Stroke, StrokeThickness, and StrokeDashArray) from the copy, and add the copy to the temporary canvas.
- d. Do not draw the original path.
- e. Draw the temporary canvas, while recursively applying the composition rules.

~~20.6.~~ If a grouping element (a <Canvas> element) has an Opacity or OpacityMask property, additional composition steps must be followed:

- a. Create a temporary surface and set its alpha channel to 0.0 (transparent) and its color value to black.
- b. Compose all child elements of the grouping element onto the temporary surface, while recursively applying the composition rules.
- c. Cumulatively apply the opacity of the grouping element and opacity mask to the alpha channel of the temporary surface.
- d. Draw the contents of the temporary surface onto the containing surface.

1 | ~~21.7.~~ If a non-grouping element (a <Path> or <Glyphs> element) has an Opacity
 2 | property, an OpacityMask property, or a fill or stroke using transparency, the following
 3 | additional composition steps must be taken:

- 4 | a. If the element has a RenderTransform property, apply it to the element and its Clip,
 5 | Fill, Stroke, and OpacityMask properties, if present.
- 6 | b. Create a mask from the set of all painted pixels representing the child element (after
 7 | the Clip property of the element has been applied).

8 | ~~22.8.~~ Combine the Fill or Stroke property with the OpacityMask and the Opacity
 9 | property and apply to the surface through the computed mask. For more information,
 10 | see §14.1.

11 | The behavior that results from this process is:

- 12 | • Opacity is not applied cumulatively to self-overlapping areas created when rendering an
 13 | individual <Glyphs> element.
- 14 | • Opacity is not applied cumulatively to self-overlapping areas created by <PathFigure>
 15 | elements within the same path (see Example 18–1).
- 16 | • Opacity is not applied cumulatively if the border of a path has self-intersections. When
 17 | the border of a path is stroked, the area of the path is filled by first applying the brush
 18 | specified by the Fill property. After filling the area, the border is drawn using the stroke-
 19 | related properties including the brush specified by the Stroke property, with half the
 20 | stroke width extending outside the filled area and half extending inside (see Example
 21 | 18–2). If the path has self-intersections, the opacity is not accumulated.
- 22 | • The color of the stroke and the color of the filled area are combined on the inside half of
 23 | a stroked border (overlapping the filled area of the path) if the brush specified by the
 24 | Stroke property is transparent.
- 25 | • If a path that has a stroked border has an opacity of less than 1.0 or an opacity mask,
 26 | the path (filled area and stroked border) is first rendered onto a temporary surface using
 27 | an opacity of 1.0 and no opacity mask (while preserving any transparency of the fill or
 28 | the stroked border themselves), and the resulting figure is drawn onto the background
 29 | using the specified opacity and opacity mask (see Example 18–3).

30 | 18.5.1 Optimization Guidelines

31 | The composition rules above describe the behavior of an ideal implementation. Practical
 32 | implementations can optimize the processing of the composition rules according to the following
 33 | guidelines:

34 | ~~23.1.~~ If all elements on a canvas and the canvas itself are opaque (an opacity of 1.0)
 35 | and parent or ancestor <Canvas> elements are also opaque, the elements MAY be drawn
 36 | directly to the containing fixed page (or canvas), provided all render transform and clip
 37 | values are observed [O11.12].

38 | ~~24.2.~~ If an element is fully transparent (an opacity of 0.0), it MAY be skipped [O11.13].

39 | ~~25.3.~~ If a canvas has an opacity of 0.0, it and all of its child and descendant elements
 40 | MAY be skipped [O11.14].

41 | ~~26.4.~~ If a canvas has a Clip property with no contained area, the canvas and all of its
 42 | child and descendant elements MAY be skipped [O11.15].

1 **27.5.** When creating a temporary surface, a consumer MAY further restrict the size of
 2 the temporary surface by the effective extent of the geometry specified by the Clip
 3 property of the canvas [O11.16].

4 **28.6.** A consumer MAY use methods to achieve transparency other than creating a
 5 temporary surface [O11.17]. Such methods MAY include planar mapping (that is,
 6 computation of intersections of transparent elements and resulting colors) [O11.17].

7 **18.5.2 Composition Examples**

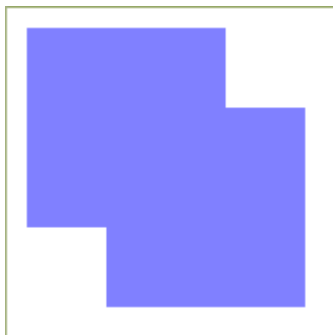
8 The following examples illustrate the composition rules described above.

9 *Example 18-1. Path opacity behavior for overlapping path figures*

10 In the following markup, opacity is not applied cumulatively to self-overlapping areas created
 11 by path figures within the same path.

```
12 <Path Opacity="0.5">
13   <Path.Fill>
14     <SolidColorBrush Color="#0000FF" />
15   </Path.Fill>
16   <Path.Data>
17     <PathGeometry FillRule="NonZero">
18       <PathFigure StartPoint="10,10" IsClosed="true">
19         <PolyLineSegment Points="110,10 110,110 10,110 10,10" />
20       </PathFigure>
21       <PathFigure StartPoint="50,50" IsClosed="true">
22         <PolyLineSegment Points="150,50 150,150 50,150 50,50" />
23       </PathFigure>
24     </PathGeometry>
25   </Path.Data>
26 </Path>
```

27 This markup is rendered as follows:



28

29 *end example]*

30 *Example 18-2. Opacity behavior of path stroke intersections*

31 In the following markup, opacity is not applied cumulatively if the border of a path has self-
 32 intersections.

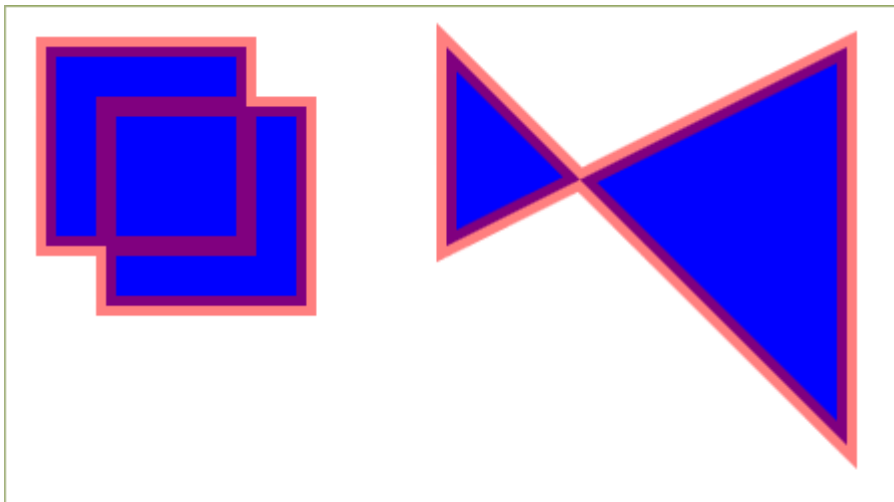
```
33 <Path Stroke="#80FF0000" StrokeThickness="10">
34   <Path.Fill>
35     <SolidColorBrush Color="#0000FF" />
36   </Path.Fill>
```

```

1      <Path.Data>
2          <PathGeometry FillRule="NonZero">
3              <PathFigure StartPoint="20,20" IsClosed="true">
4                  <PolyLineSegment Points="120,20 120,120 20,120 20,20" />
5              </PathFigure>
6              <PathFigure StartPoint="50,50" IsClosed="true">
7                  <PolyLineSegment Points="150,50 150,150 50,150 50,50" />
8              </PathFigure>
9          </PathGeometry>
10     </Path.Data>
11 </Path>
12 <Path Stroke="#80FF0000" StrokeThickness="10" StrokeMiterLimit="10">
13     <Path.Fill>
14         <SolidColorBrush Color="#0000FF" />
15     </Path.Fill>
16     <Path.Data>
17         <PathGeometry>
18             <PathFigure StartPoint="220,20" IsClosed="true">
19                 <PolyLineSegment Points="420,220 420,20 220,120" />
20             </PathFigure>
21         </PathGeometry>
22     </Path.Data>
23 </Path>

```

24 This markup is rendered as follows:



25

26 *end example]*

27 *Example 18-3. Opacity behavior of paths with stroked edges*

28 The following markup describes a path with a stroked border and an opacity of less than 1.0:

```

29     <Path>
30         <Path.Fill>
31             <SolidColorBrush Color="#7F7F7F" />
32         </Path.Fill>
33         <Path.Data>
34             <PathGeometry>
35                 <PathFigure StartPoint="0,110" IsClosed="true">

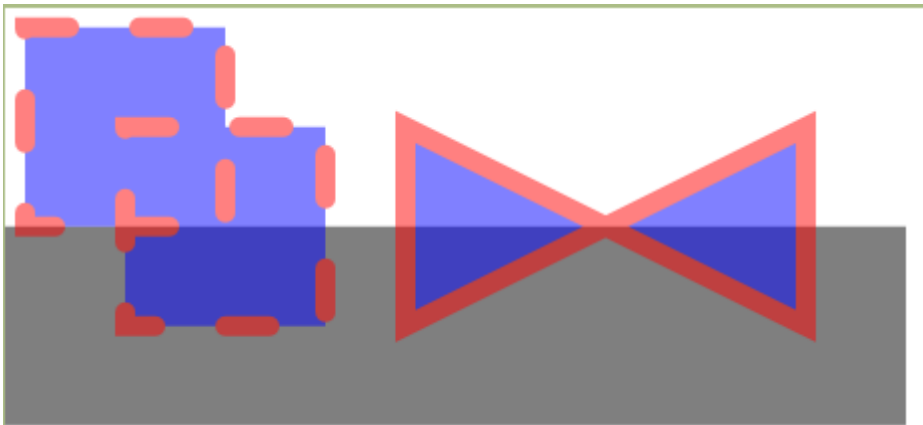
```

```

1         <PolyLineSegment Points="450,110 450,210 0,210" />
2     </PathFigure>
3 </PathGeometry>
4 </Path.Data>
5 </Path>
6 <Path
7     Stroke="#FF0000"
8     StrokeThickness="10"
9     StrokeDashArray="2.2 3.5"
10    StrokeDashCap="Round"
11    Opacity="0.5">
12 <Path.Fill>
13     <SolidColorBrush Color="#0000FF" />
14 </Path.Fill>
15 <Path.Data>
16     <PathGeometry FillRule="NonZero">
17         <PathFigure StartPoint="10,10" IsClosed="true">
18             <PolyLineSegment Points="110,10 110,110 10,110 10,10" />
19         </PathFigure>
20         <PathFigure StartPoint="60,60" IsClosed="true">
21             <PolyLineSegment Points="160,60 160,160 60,160 60,60" />
22         </PathFigure>
23     </PathGeometry>
24 </Path.Data>
25 </Path>
26 <Path Stroke="#FF0000" StrokeThickness="10" Opacity="0.5">
27 <Path.Fill>
28     <SolidColorBrush Color="#0000FF" />
29 </Path.Fill>
30 <Path.Data>
31     <PathGeometry>
32         <PathFigure StartPoint="200,60" IsClosed="true">
33             <PolyLineSegment Points="400,160 400,60 200,160" />
34         </PathFigure>
35     </PathGeometry>
36 </Path.Data>
37 </Path>

```

38 This markup is rendered as follows:



39

1 *end example]*

18.6 Stroke Rendering

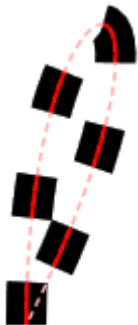
Strokes follow the contours of each segment in a path figure, as specified by the various stroke-related properties.

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 can 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 18-1. 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 other words, a line from 0,0 to 102,50 should result in the same pixel set as a line from 102,50 to 0,0.

1 **18.6.4 Rules for Dash Cap Rendering**

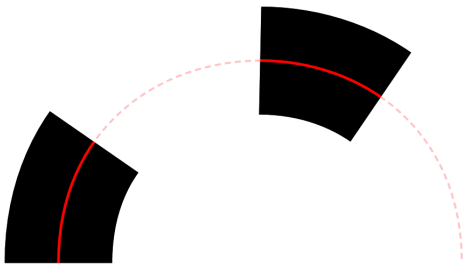
2 The appearance of dash caps is controlled by the StrokeDashCap attribute. Valid values are Flat,
3 Square, Round, and Triangle. The StrokeDashCap attribute is ignored for paths that have no
4 StrokedDashArray attribute or that have a StrokedDashArray attribute with value 0,0.

5 **18.6.4.1 Flat Dash Caps**

6 The effective render transform of the path being stroked is used to transform the control points
7 of the contour of the dash.

8 The length of the dash is the approximate distance on the curve between the two intersections
9 of the flat lines ending the dash and the contour of the shape. The distance from the end of one
10 dash to the start of the next dash is the specified dash gap length. Dashes with a length greater
11 than 0 are drawn, and degenerate dashes with a length of 0 are not drawn.

12 *Figure 18–2. Flat dash caps*



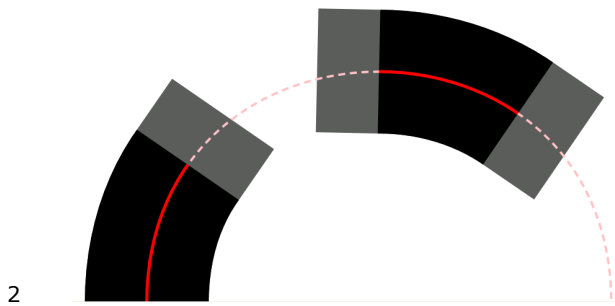
13 **18.6.4.2 Square Dash Caps**

14 The effective render transform of the path being stroked is used to transform the control points
15 of the contour of the dash.

16 The length of the dash is the approximate distance on the curve between the two *contour*
17 *intersection points*, that is, the intersection of the flat line ending the dash (without the square
18 caps attached) and the contour of the shape.

19 The caps are drawn as half-squares attached to the ends of the dash. The boundaries of the
20 square caps are not curved to follow the contour, but are transformed using the effective
21 render transform.

22 The distance between the contour intersection points of consecutive dashes is the specified
23 dash gap length. Degenerate dashes with a length of 0 are drawn as squares. [If a dash with a
24 length of 0 appears at, or very near to, a join in a path then differences in rendering resolution
25 and in precision in the calculation of coordinates may lead to differing orientation of the dash
26 caps between consumers.](#)
27

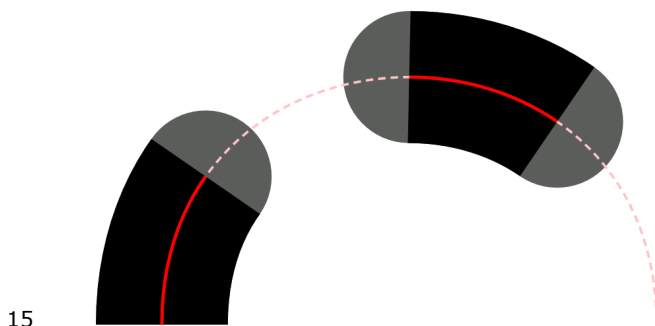
1 *Figure 18-3. Square dash caps*2
3 **18.6.4.3 Round Dash Caps**

4 The effective render transform of the path being stroked is used to transform the control points
5 of the contour of the dash.

6 The length of the dash is the approximate distance on the curve between the two contour
7 intersection points, that is, the intersection of the flat line ending the dash (without the round
8 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 of the
10 round caps are not distorted to follow the contour, but are transformed using the effective
11 render transform.

12 The distance between the contour intersection points of consecutive dashes is the specified
13 dash gap length. Degenerate dashes with a length of 0 are drawn as circles.

14 *Figure 18-4. Round dash caps*15
16 **18.6.4.4 Triangular Dash Caps**

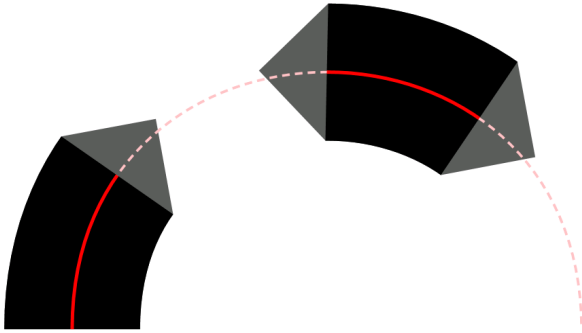
17 The effective render transform of the path being stroked is used to transform the control points
18 of the contour of the dash.

19 The length of the dash is the approximate distance on the curve between the two contour
20 intersection points, that is, the intersection of the flat line ending the dash (without the
21 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. The
23 boundaries of the triangular caps are not distorted to follow the contour, but are transformed
24 using the effective render transform. The height of the triangles is half of the stroke width.

1 The distance between the contour intersection points of consecutive dashes is the specified
 2 dash gap length. Degenerate dashes with a length of 0 are drawn as diamonds. [If a dash with a
 3 length of 0 appears at, or very near to, a join in a path then differences in rendering resolution
 4 and in precision in the calculation of coordinates may lead to differing orientation of the dash
 5 caps between consumers.](#)

6 *Figure 18-5. Triangular dash caps*

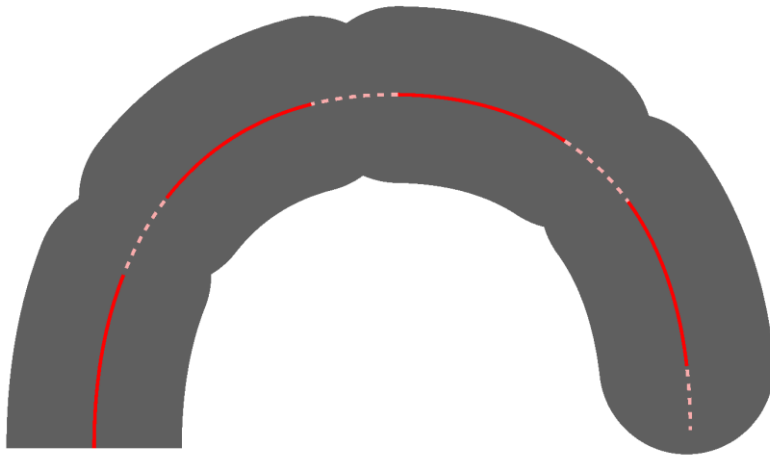


7

8 **18.6.4.5 Overlapping Dashes**

9 It is possible to specify dash sequences with overlapping dash caps. In this circumstance, the
 10 union of the dash segments (inclusive of dash caps), is used as a mask through which the
 11 brush is applied as illustrated in Figure 18-6 with a stroke dash cap value of Round.

12 *Figure 18-6. Overlapping dash segments*



13

14 **18.6.4.6 Extreme Degenerate Dash Case**

15 The previous subclasses include a description of the behaviour for degenerate dashes of zero
 16 length, with non-zero gaps, for each dash cap shape.

17 Producers SHOULD NOT create files containing the extreme degenerate case of
 18 StrokeDashArray = "0 0". Such lines SHOULD be rendered as a solid line [S11.32].

19 **18.6.5 Rules for Line Cap Rendering**

20 The appearance of line caps is controlled by the StrokeStartLineCap and StrokeEndLineCap
 21 attribute. Valid values are Flat, Square, Triangle, and Round. Every start line cap can be used in

1 combination with any end line cap. Line caps only ever appear at the start and end of an open
 2 path, and then only if the initial/final segment is stroked.

3 The rules for line caps on curved lines are analogous to the rules for dash cap rendering. For
 4 more information, see §18.6 and §18.6.4.

5 *Figure 18-7. Flat start line cap, flat end line cap*



6

7 *Figure 18-8. Square start line cap, square end line cap*



8

9 *Figure 18-9. Triangular start line cap, triangular end line cap*



10

11 *Figure 18-10. Round start line cap, round end line cap*



12

13 **18.6.6 Line Caps for Dashed Strokes**

14 If the start point of a stroke is within a dash or touches the start or end of a dash, a start line
 15 cap is appended to the stroke. Similarly, if the end point of a stroke is within a dash or touches
 16 the start or end of a dash, an end line cap is appended to the stroke.

17 *Figure 18-11. Stroke start or end point within a dash for flat dash caps*



18

19 *Figure 18-12. Stroke start or end point within a dash for non-flat dash caps*



20

21 [Note: Because the right-most line cap begins at the point exactly coincident with the start of
 22 the next dash in the sequence, it is rendered. *end note*]

1 However, if the start point of a stroke is within a gap (as can result from a StrokeDashOffset
 2 attribute), no start line cap is appended to the stroke. If the end point of a stroke is within a
 3 gap, no end line cap is appended to the stroke.

4 *Figure 18–13. Stroke start or end point within a gap for flat dash caps*



5

6 *Figure 18–14. Stroke start or end point within a gap for not-flat dash caps*



7

8 [Note: Differences in precision in the calculation of coordinates can lead to differing output
 9 between consumers depending on whether they determine that the start or end point of a
 10 stroke exactly touches the start or end point of a dash. *end note*]

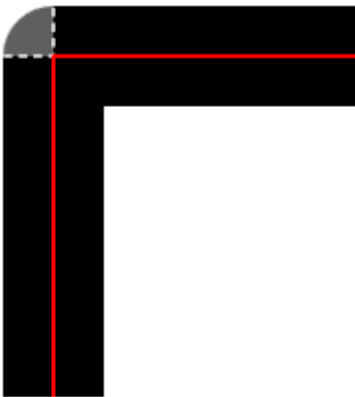
11 **18.6.7 Rules for Line Join Rendering**

12 The appearance of line joins is controlled by the StrokeLineJoin attribute. Valid values are Round,
 13 Bevel, and Miter.

14 **18.6.7.1 Round Line Joins**

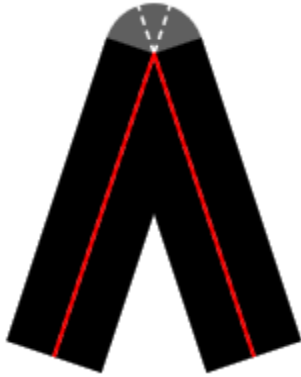
15 A StrokeLineJoin attribute value of Round indicates that the outer corner of the joined lines
 16 should be filled by enclosing the rounded region with its center point at the point of intersection
 17 between the two lines and a radius of one-half the stroke thickness value.

18 *Figure 18–15. Round line join with right angle*



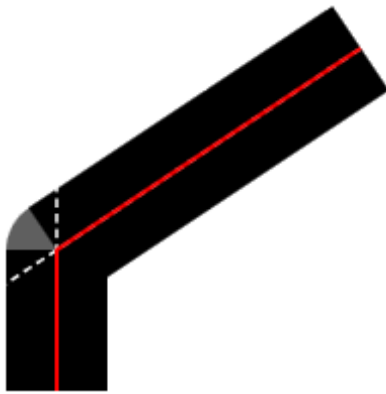
19

1 *Figure 18–16. Round line join with acute angle*



2

3 *Figure 18–17. Round line join with obtuse angle*

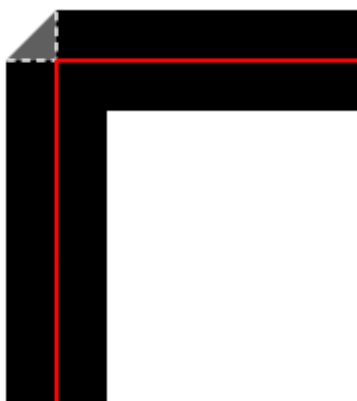


4

5 **18.6.7.2 Beveled Line Joins**

6 A StrokeLineJoin attribute value of Bevel indicates that the outer corner of the joined lines should
 7 be filled by enclosing the triangular region of the corner with a straight line between the outer
 8 corners of each stroke.

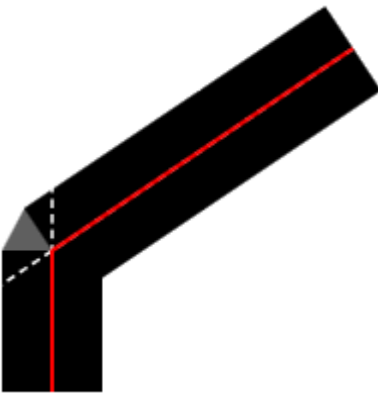
9 *Figure 18–18. Beveled line join with right angle*



10

1 *Figure 18–19. Beveled line join with acute angle*

2

3 *Figure 18–20. Beveled line join with obtuse angle*

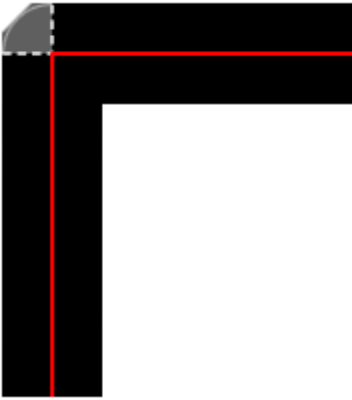
4

5 **18.6.7.3 Mitered Line Joins**

6 If the StrokeLineJoin attribute value is Miter, the value of the StrokeMiterLimit attribute value is
 7 also used for rendering these joins. A StrokeLineJoin value of Miter indicates that the region to
 8 be filled includes the intersection of the strokes projected to infinity, and then clipped at a
 9 specific distance. The intersection of the strokes is clipped at a line perpendicular to the bisector
 10 of the angle between the strokes, at the distance equal to the stroke miter limit value multiplied
 11 by half the stroke thickness value.

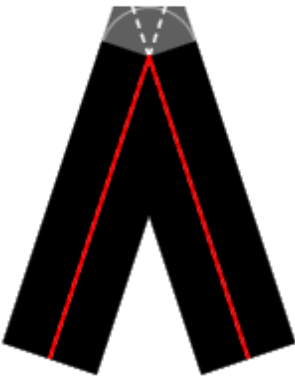
12 When drawing mitered line joins, the presence of one or more degenerate line segments
 13 between the non-degenerate line segments to be joined results in a mitered line join of only the
 14 two non-degenerate line segments with an implied StrokeMiterLimit attribute value of 1.0.

1 *Figure 18–21. Mitered line join with right angle and miter limit of 1.0*



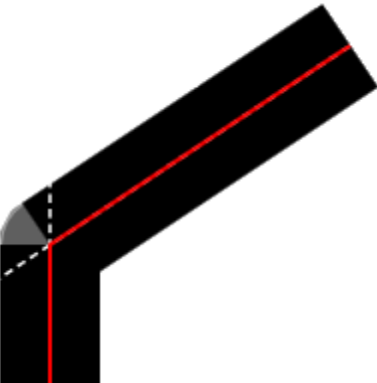
2

3 *Figure 18–22. Mitered line join with acute angle and miter limit of 1.0*



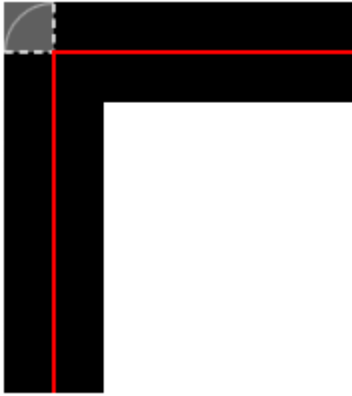
4

5 *Figure 18–23. Mitered line join with obtuse angle and miter limit of 1.0*

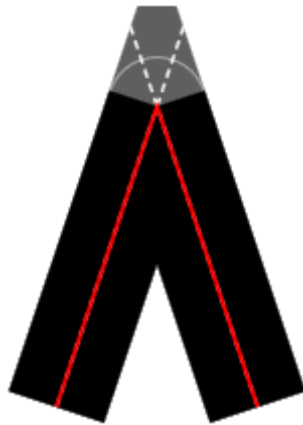


6

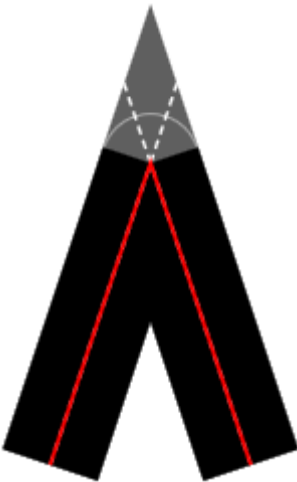
1 *Figure 18–24. Mitered line join with right angle and miter limit of 2.0*



2
3 *Figure 18–25. Mitered line join with acute angle and miter limit of 2.0*



4
5 *Figure 18–26. Mitered line join with acute angle and miter limit of 10.0*



6
7 **18.6.8 Rules for Degenerate Line and Curve Segments**

8 Degenerate line segments (that is, where the start point and end point coincide) are not drawn.

1 Degenerate curve segments (where the start point, end point, and all control points coincide)
2 are not drawn.

3 If an open degenerate path (formed from degenerate line or curve segments) with non-flat
4 start cap and/or non-flat end line cap is stroked, only the start line cap and/or end line cap is
5 drawn, in the x direction relative to the current effective render transform (that is, as if a
6 segment were drawn from x,y to $x+d,y$, with $d \rightarrow 0$).

7 If a closed degenerate path (formed from degenerate line or curve segments) is stroked, a
8 circular dot with a diameter of the stroke thickness is drawn instead.

9 If the current render transform is an invertible matrix, consumers SHOULD perform
10 computations on poly line segments and poly Bézier segments with sufficient accuracy to avoid
11 producing zero-length segments [S11.25].

12 **18.6.9 Stroking and Fill Rule**

13 Stroking a path is independent of the fill rule. The fill rule affects the filled area only.

14 **18.6.10 Mixing Stroked and Non-Stroked Segments**

15 When a path figure contains multiple segments and one or more of the segments has an
16 `IsStroked` value of false, the phase for dashes starts anew with the next stroked segment,
17 including application of the dash offset.

18 When a segment of a dashed path is stroked and the subsequent segment has an `IsStroked`
19 value of false, thus causing a dash to be truncated, the dash cap is drawn for both ends of the
20 truncated dash, exactly as it would for a non-truncated dash. For the case of a closed dashed
21 path, this rule also applies to dashes exposed at the beginning or end of the path by an
22 unstroked final or initial segment respectively.

23 **18.6.11 Stroke Behavior with Multiple Path Figures**

24 When a geometry containing multiple path figures is stroked, the phase for dashes (including
25 application of the dash offsets) starts anew with each new path figure.

26 In general, for any path geometry, each path figure is drawn independently of every other path
27 figure, so the dash array is reset for each. Dashes are also reset after every unstroked
28 segment.

29 **18.6.12 Consistent Nominal Stroke Width**

30 For certain scenarios, it is desirable for producers to generate documents targeted at specific
31 aliasing consumers with particular lines in the document indicated as hairlines or consistent-
32 width strokes. The following recommendation allows these producers and consumers to handle
33 these strokes consistently.

34 Producers MAY generate a `<Path>` element intended to be treated as having a consistent
35 nominal stroke width by specifying the `StrokeDashArray` attribute and by specifying a
36 `StrokeDashOffset` attribute value less than -1.0 times the sum of all the numbers in the
37 `StrokeDashArray` attribute value [O11.25].

38 For a solid line, the producer would set the `StrokeDashArray` to the value `"1 0"` and the
39 `StrokeDashOffset` to a value such as `"-2"`. The `"-2"` value fulfills the restriction on the
40 `StrokeDashOffset` value in a numerically stable manner, and the phase of the dash pattern is

1 identical to a StrokeDashOffset value of "0". Values less than "-2" can be used to specify a shifted
2 phase of the dash pattern.

3 A stroke using the consistent nominal stroke width convention SHOULD be rendered with a
4 width consistent with other strokes using the convention that have the same StrokeThickness
5 attribute value, and consumers aware of this convention SHOULD render such a stroke no
6 thinner than the thinnest visible line that consumer supports without dropouts [S11.31]. See
7 §18.1.4~~11.1.4~~, for further considerations for rendering thin lines.

8 **18.7 Brushes and Images**

9 Images require the following special considerations for scaling and tile placement.

10 **18.7.1 Small Tiles**

11 Tiles for visual brushes and image brushes can be specified with a viewport width or height of a
12 few device pixels, or even less than a single device pixel in size.

13 If both width and height are nearly zero, implementations SHOULD average the color values of
14 the brush contents, resulting in a constant-color brush [S11.26]. [*Example:*

- 15 • A visual brush or image brush that contains a blue and white checkerboard pattern
16 results in a solid light-blue fill as either the width or the height value approaches 0.0.
- 17 • A visual brush or image brush whose viewbox is constant-colored produces a constant-
18 colored brush regardless of the width and height values of the viewport.

19 *end example]*

20 If only one of the width and height values is nearly zero, the brush should be constant-colored
21 along lines parallel to the narrow side of the viewport. For cases such as these,
22 implementations MAY differ [O11.21]. Producers SHOULD avoid producing such extreme cases
23 and SHOULD NOT rely on any specific behavior when they do [S11.27].

24 **18.7.2 Image Scaling**

25 Source sampling SHOULD be done from the center of the pixel and should be mapped to the
26 center of the pixel in the device-space [S11.28]. With one extent of the viewbox zero, sampling
27 SHOULD be done along a line parallel to the non-zero side [S11.28]. With both extents of the
28 viewbox zero, a point sample SHOULD be taken [S11.28].

29 When up-sampling an image presented at a lower resolution than the device resolution, bilinear
30 filtering SHOULD be used [S11.29]. The precise source coordinates as specified by the viewbox
31 MUST be used to place the up-sampled image tile, which is equivalent to using fractional pixels
32 of the original source image [M11.8].

33 When down-sampling, at least a bilinear filter SHOULD be used [S11.30]. Consumers MAY
34 choose to implement a more sophisticated algorithm, such as a Fant scaler, to prevent aliasing
35 artifacts [O11.22].

36 **18.7.3 Tile Placement**

37 Consumers MUST precisely position the tiles specified by the image brush and visual brush. If
38 the specified values result in fractional device pixels, the consumer MUST calculate a running
39 placement-error delta and adjust the placement of the next tile where the delta reaches a full

1 device pixel in order to keep the tiles from being increasingly out of phase as the expanse of
2 the path is filled [M11.9]. Consumers MAY choose any technique desired to achieve this
3 requirement, such as linear filtering for seams, stretching of the tile (up-sampling or down-
4 sampling), or pre-computing multiple tiles and adjusting behavior according to how the tiles fit
5 on a grid [O11.23].

6 **18.7.4 Tiling Transparent Visual Brushes and Image Brushes**

7 The contents of a visual brush's Visual property are first rendered to a temporary work canvas
8 [\(according to see the composition rules in §18.5.\)](#) ~~with an opacity of 0.0~~. The viewBox of the
9 visual brush defines the tile or portion of the temporary canvas that is copied onto the specified
10 geometry, stroke, or text. Likewise, an image specified by an image brush is also copied to a
11 temporary work canvas. The viewBox also defines the tile for an image brush. In either case,
12 the work canvas is scaled to properly match the edges of the tile to the size specified by the
13 viewport.

14 Each pixel of the resultant tile is separately blended with the background of its destination,
15 using the alpha of each pixel. This process is repeated for each tile replication, while respecting
16 the TileMode attribute value, although the temporary work canvas MAY be re-used [O11.24].

17

19. Elements

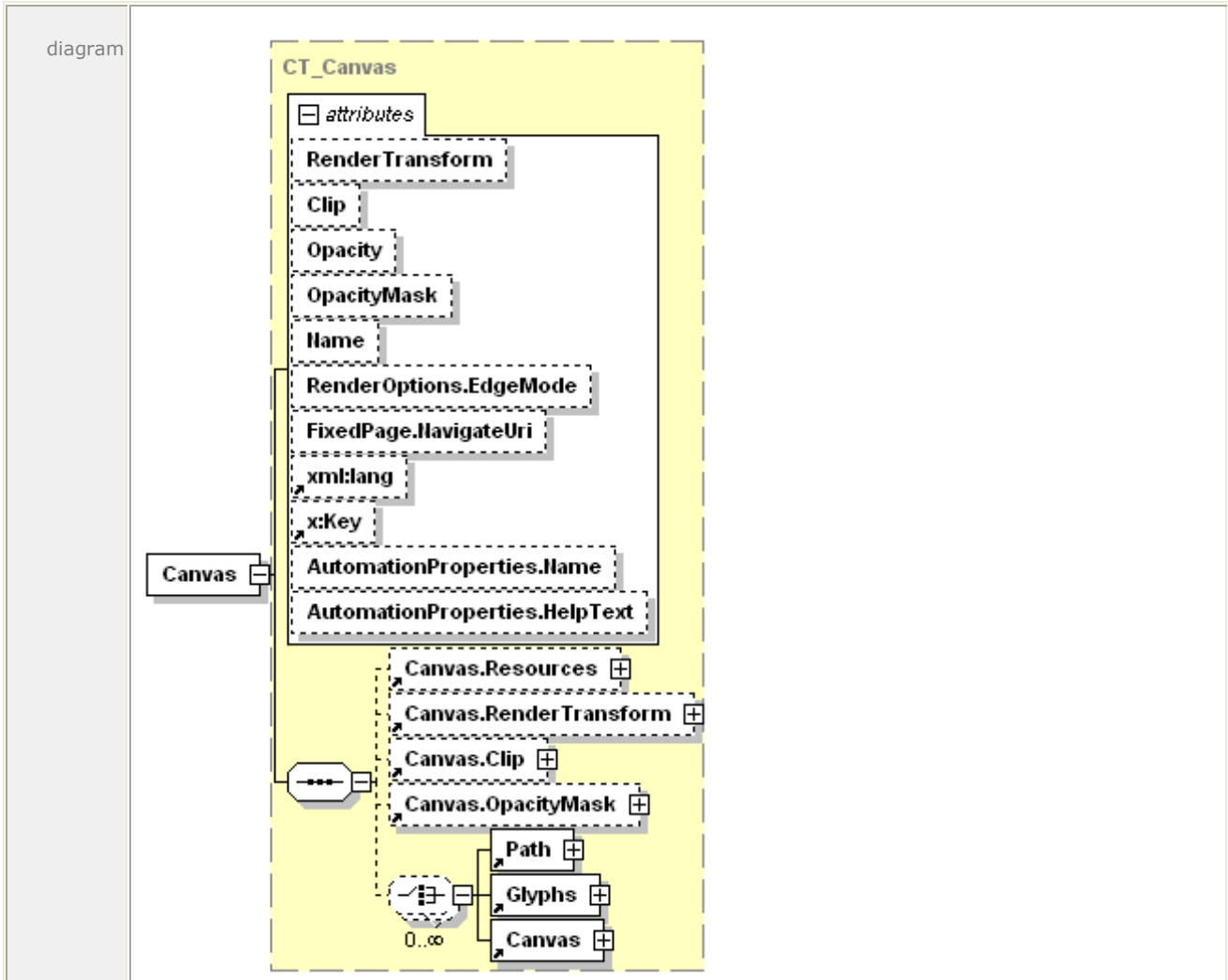
19.1 ArcSegment

element **ArcSegment**

<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>Point</td> <td>ST_Point</td> <td>required</td> <td></td> <td></td> <td>Specifies the endpoint of the elliptical arc.</td> </tr> <tr> <td>Size</td> <td>ST_PointGE0</td> <td>required</td> <td></td> <td></td> <td>Specifies the x and y radius of the elliptical arc as an x,y pair.</td> </tr> <tr> <td>RotationAngle</td> <td>ST_Double</td> <td>required</td> <td></td> <td></td> <td>Indicates how the ellipse is rotated relative to the current coordinate system.</td> </tr> <tr> <td>IsLargeArc</td> <td>ST_Boolean</td> <td>required</td> <td></td> <td></td> <td>Determines whether the arc is drawn with a sweep of 180 or greater. Can be true or false.</td> </tr> <tr> <td>SweepDirection</td> <td>ST_SweepDirection</td> <td>required</td> <td></td> <td></td> <td>Specifies the direction in which the arc is drawn. Valid values are Clockwise and Counterclockwise.</td> </tr> <tr> <td>IsStroked</td> <td>ST_Boolean</td> <td></td> <td>true</td> <td></td> <td>Specifies whether the stroke for this segment of the path is drawn. Can be true or false.</td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	Annotation	Point	ST_Point	required			Specifies the endpoint of the elliptical arc.	Size	ST_PointGE0	required			Specifies the x and y radius of the elliptical arc as an x,y pair.	RotationAngle	ST_Double	required			Indicates how the ellipse is rotated relative to the current coordinate system.	IsLargeArc	ST_Boolean	required			Determines whether the arc is drawn with a sweep of 180 or greater. Can be true or false.	SweepDirection	ST_SweepDirection	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.				
Name	Type	Use	Default	Fixed	Annotation																																										
Point	ST_Point	required			Specifies the endpoint of the elliptical arc.																																										
Size	ST_PointGE0	required			Specifies the x and y radius of the elliptical arc as an x,y pair.																																										
RotationAngle	ST_Double	required			Indicates how the ellipse is rotated relative to the current coordinate system.																																										
IsLargeArc	ST_Boolean	required			Determines whether the arc is drawn with a sweep of 180 or greater. Can be true or false.																																										
SweepDirection	ST_SweepDirection	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.																																										
<p>annotation</p>	<p>Represents an elliptical arc between two points.</p>																																														

1 **19.2 Canvas**

2 element **Canvas**



attributes	Name	Type	Use	Default	Fixed	Annotation
	RenderTransform	ST_RscRefMatrix				Establishes a new coordinate frame for the child and descendant elements of the canvas, such as another canvas. Also affects clip and opacity mask.
	Clip	ST_RscRefAbbrGeomF				Limits the rendered region of the element.
	Opacity	ST_ZeroOne		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	<u>ST_RscRef</u>				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	<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.
RenderOptions.EdgeMode	<u>ST_EdgeMode</u>				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.

2 19.3 Canvas.Clip

3 element **Canvas.Clip**

diagram	
annotation	Limits the rendered region of the element.

4 For more information, see §14.3.1 ~~14.3~~.

5 19.4 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.

1 For more information, see §14.5.1.

2 **19.5 Canvas.RenderTransform**

3 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.

4 For more information, see §14.4.2~~14.4~~.

5 **19.6 Canvas.Resources**

6 element **Canvas.Resources**

diagram	
annotation	Contains the resource dictionary for the <Canvas> element.

7 For more information, see §14.2.2~~14.2~~.

8 **19.7 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.
------------	--

1 For more information, see §17.1.4.1.2.

2 **19.8 DiscardControl**

3 element **DiscardControl**

diagram	
annotation	Contains a list of resources that are safe for a consumer to discard.

4 For more information, see §17.1.4.1.1.

5 **19.9 DocumentOutline**

6 element **DocumentOutline**

diagram													
attributes	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Name</th> <th style="width: 10%;">Type</th> <th style="width: 10%;">Use</th> <th style="width: 10%;">Default</th> <th style="width: 10%;">Fixed</th> <th style="width: 45%;">Annotation</th> </tr> </thead> <tbody> <tr> <td>xml:lang</td> <td></td> <td>required</td> <td></td> <td></td> <td> <p>Specifies the default language used for the current element and for any child or descendant elements. The language is specified according to RFC 3066.[rcj6] 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.</p> </td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	Annotation	xml:lang		required			<p>Specifies the default language used for the current element and for any child or descendant elements. The language is specified according to RFC 3066.[rcj6] 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.</p>
Name	Type	Use	Default	Fixed	Annotation								
xml:lang		required			<p>Specifies the default language used for the current element and for any child or descendant elements. The language is specified according to RFC 3066.[rcj6] 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.</p>								
annotation	Specifies a list of meaningful indices into the XPS Document, similar to a table of contents, or to external URIs, such as web addresses.												

7 For more information, see §16.1.1.3.

1 **19.10 DocumentReference**

2 element **DocumentReference**

diagram	<p>The diagram shows a box labeled 'DocumentReference' with a small square icon on its right side. A line connects this icon to a larger box labeled 'CT_DocumentF'. Inside 'CT_DocumentF', there is a sub-box labeled 'attributes' containing a box labeled 'Source'.</p>					
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.

4 **19.11 DocumentStructure**

5 element **DocumentStructure**

diagram	<p>The diagram shows a box labeled 'DocumentStructure' with a small square icon on its right side. A line connects this icon to a larger box labeled 'CT_DocumentStructure'. Inside 'CT_DocumentStructure', there is a dashed-line box containing two sub-boxes: 'DocumentStructure.Outline' and 'Story'. Both sub-boxes have a small square icon with a plus sign on their right side. Below the dashed-line box is the text '0..∞'.</p>					
annotation	The root element of the DocumentStructure part.					

6 For more information, see §16.1.1.1.

7 **19.12 DocumentStructure.Outline**

8 element **DocumentStructure.Outline**

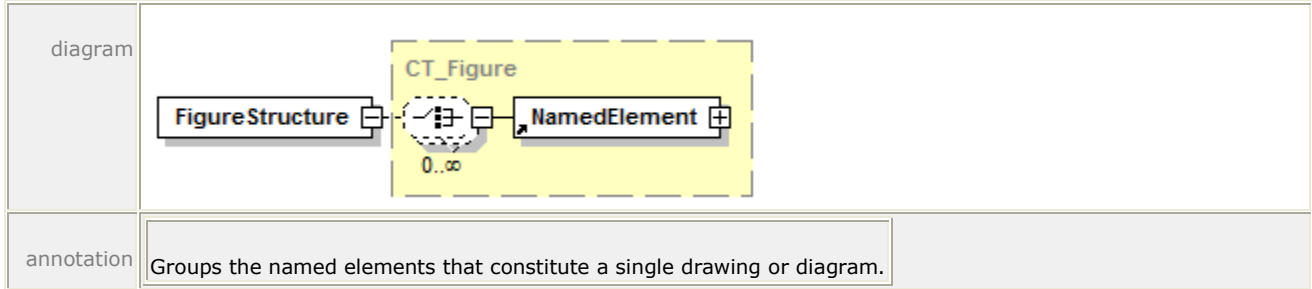
diagram	<p>The diagram shows a box labeled 'DocumentStructure.Outline' with a small square icon on its right side. A line connects this icon to a larger box labeled 'CT_CP_Outline'. Inside 'CT_CP_Outline', there is a sub-box labeled 'DocumentOutline' with a small square icon with a plus sign on its right side.</p>					
annotation	Contains a structured document outline that provides a list of links into the document contents or external sites.					

9

1 For more information see §16.1.1.2.

2 **19.13 FigureStructure**

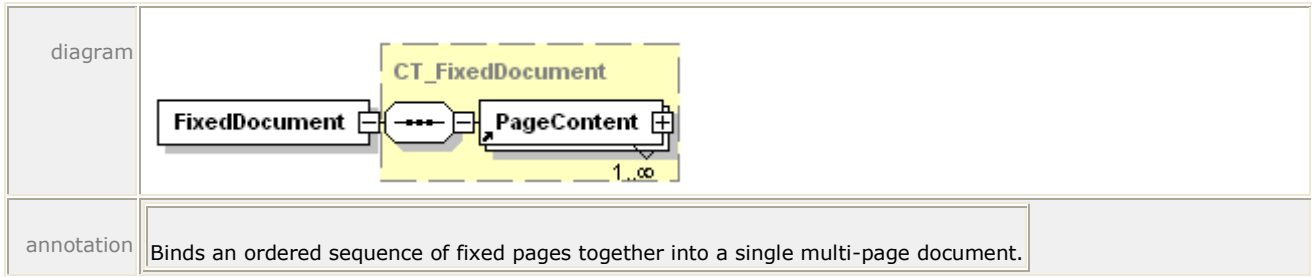
3 element **FigureStructure**



4 For more information, see §16.1.2.12.

5 **19.14 FixedDocument**

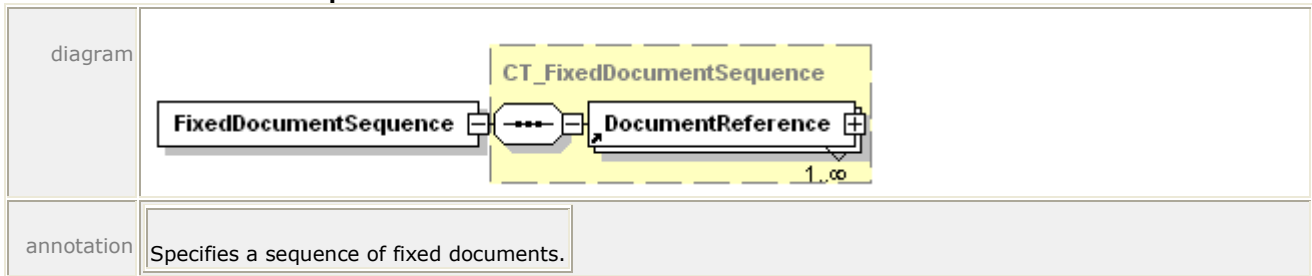
6 element **FixedDocument**



7 For more information, see §10.2.

8 **19.15 FixedDocumentSequence**

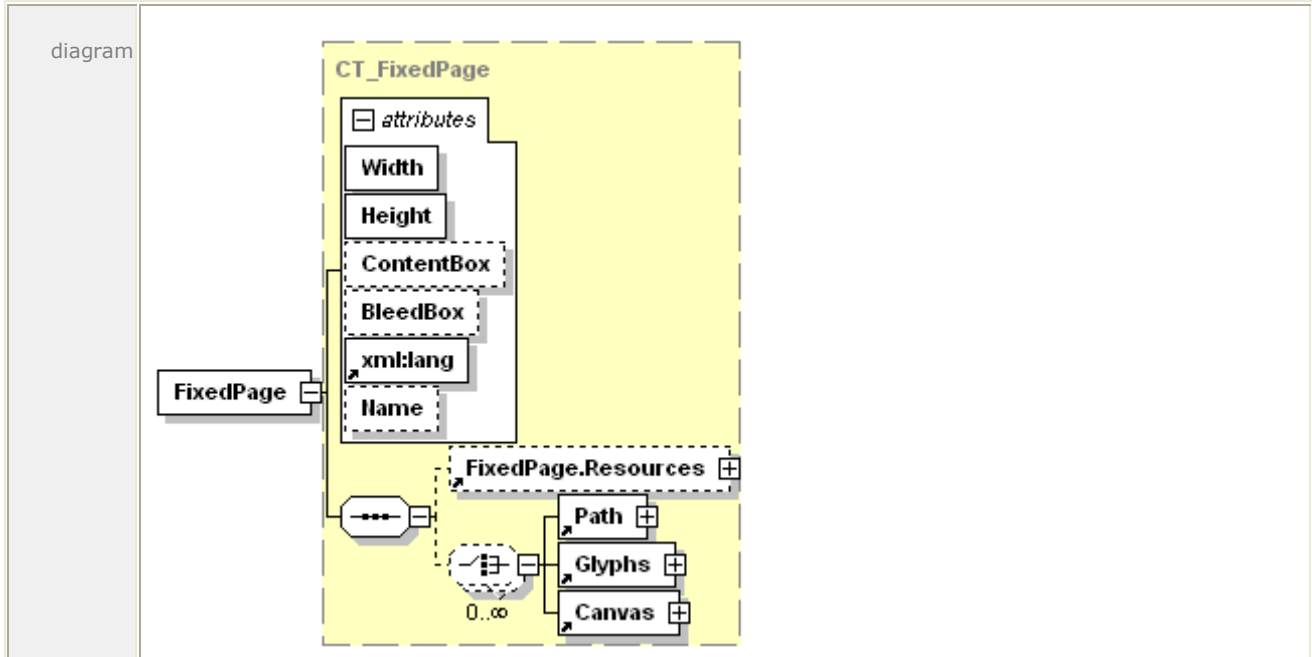
9 element **FixedDocumentSequence**



10 For more information, see §10.1.

11 **19.16 FixedPage**

12 element **FixedPage**



Name	Type	Use	Default	Fixed	Annotation
Width	<u>ST_GEOne</u>	required			Width of the page, expressed as a real number in units of the effective coordinate space.
Height	<u>ST_GEOne</u>	required			Height of the page, expressed as a real number in units of the effective coordinate space.
ContentBox	<u>ST_ContentBox</u>				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	<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.

2 **19.17 FixedPage.Resources**

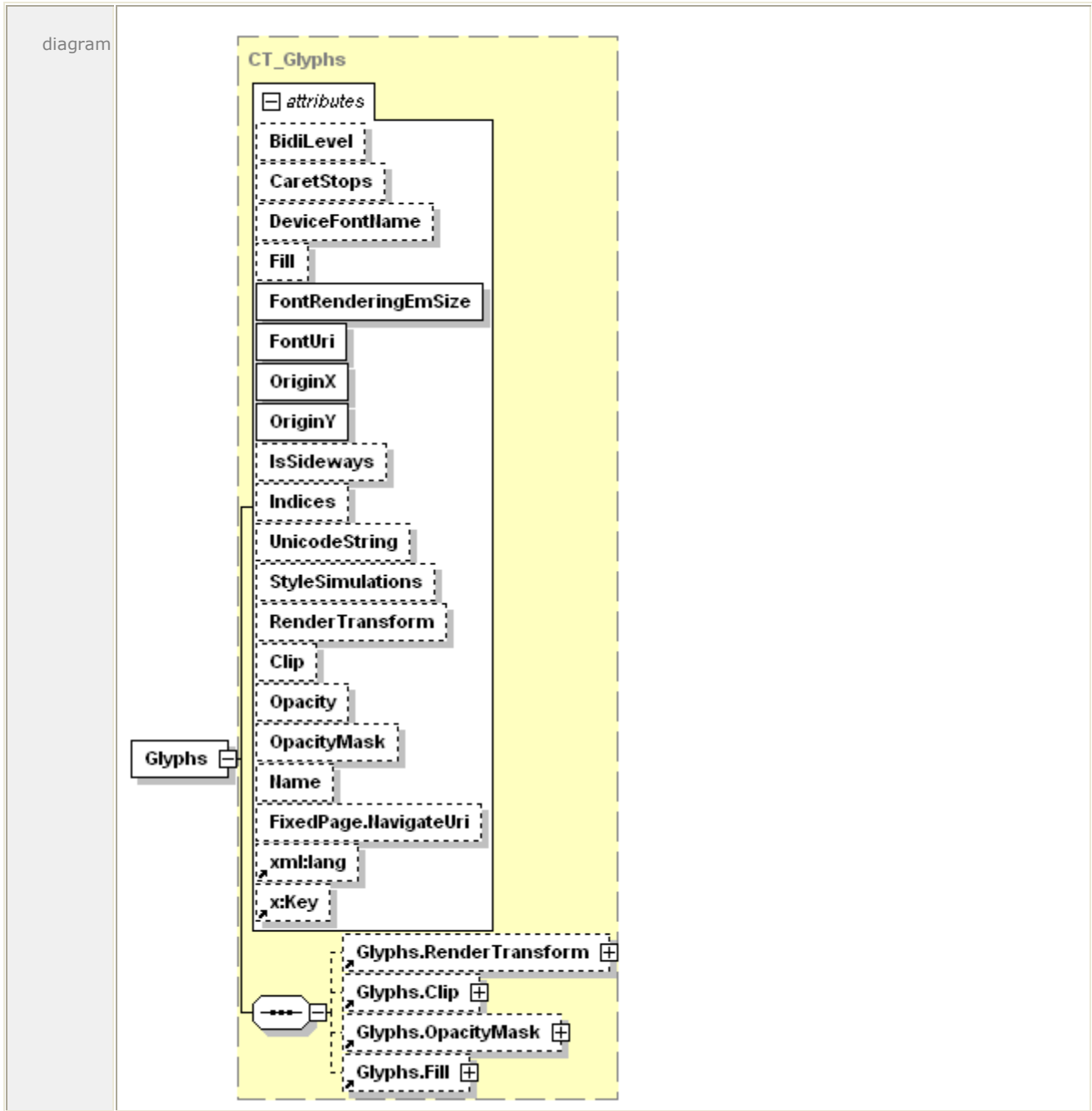
3 element **FixedPage.Resources**

diagram	<pre> classDiagram class FixedPageResources["FixedPage.Resources"] class ResourceDictionary FixedPageResources ..> ResourceDictionary class CT_CP_Resources["CT_CP_Resources"] ResourceDictionary .. > CT_CP_Resources </pre>					
annotation	Contains the resource dictionary for the <FixedPage> element.					

4 For more information, see §14.2.

5 **19.18 Glyphs**

7 element **Glyphs**



attributes	Name	Type	Use	Default	Fixed	Annotation
	BidiLevel			0		Specifies the Unicode algorithm bidirectional nesting level. Even 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.
CaretStops	<u>ST_CaretStops</u>				Identifies the positions within the sequence of Unicode characters at which a text-selection tool can 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	<u>ST_RscRefColor</u>				Describes the brush used to fill the shape of the rendered glyphs.
FontRenderingEmSize	<u>ST_GEZero</u>	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.

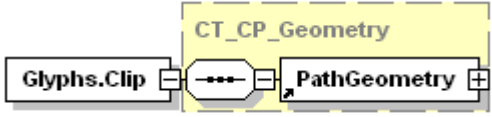
OriginX	<u>ST_Double</u>	required			Specifies the x coordinate of the first glyph in the run, in units of the effective coordinate space. The glyph is placed so that the leading edge of its advance vector and its baseline intersect with the point defined by the OriginX and OriginY attributes.
OriginY	<u>ST_Double</u>	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	<u>ST_Boolean</u>		false		Indicates that a glyph is turned on its side, with the origin being defined as the top center of the unturned glyph.
Indices	<u>ST_Indices</u>				Specifies a series of glyph indices and their attributes used for rendering the glyph run. If the UnicodeString attribute specifies an empty string ("{}") and the Indices attribute is not specified or is also empty, a consumer MUST instantiate an error condition generate an error [M5.2].
UnicodeString	<u>ST_UnicodeString</u>				Contains the string of text rendered by the <Glyphs> element. The text is specified as Unicode code points.
StyleSimulations	<u>ST_StyleSimulations</u>		None		Specifies a style simulation. Valid values are None, ItalicSimulation, BoldSimulation, and BoldItalicSimulation.
RenderTransform	<u>ST_RscRefMatrix</u>				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	ST_RscRefAbbrGeomF				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	ST_ZeroOne		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	ST_RscRef				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	ST_Name				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 §12.1, [and §9.1.7,](#)~~and §12.1.3.~~

2 **19.19 Glyphs.Clip**

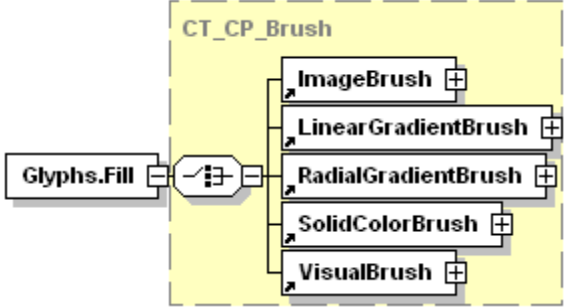
3 element **Glyphs.Clip**

diagram	
annotation	<p>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.</p>

4 For more information, see §14.3.~~3~~[14.3.](#)

5 **19.20 Glyphs.Fill**

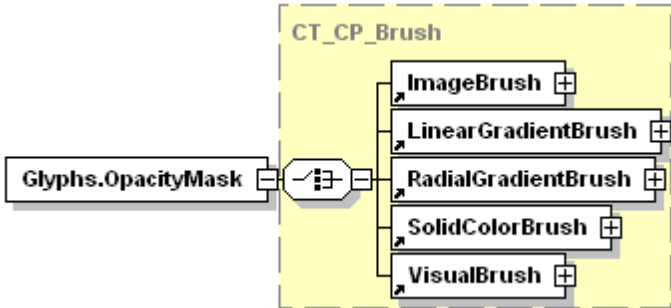
6 element **Glyphs.Fill**

diagram	
annotation	<p>Describes the brush used to fill the shape of the rendered glyphs.</p>

7 For more information, see §12.2.

8 **19.21 Glyphs.OpacityMask**

9 element **Glyphs.OpacityMask**

diagram	
---------	---

annotation	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.
------------	--

1 For more information, see §14.5.3.

2 19.22 Glyphs.RenderTransform

3 element **Glyphs.RenderTransform**

diagram	<p>The diagram shows a box labeled 'Glyphs.RenderTransform' connected by a dashed line to a box labeled 'MatrixTransform'. A yellow dashed box labeled 'CT_CP_Transform' encompasses the 'MatrixTransform' box.</p>
annotation	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.

4 For more information, see §14.4.4~~14.4~~.

5 19.23 GradientStop

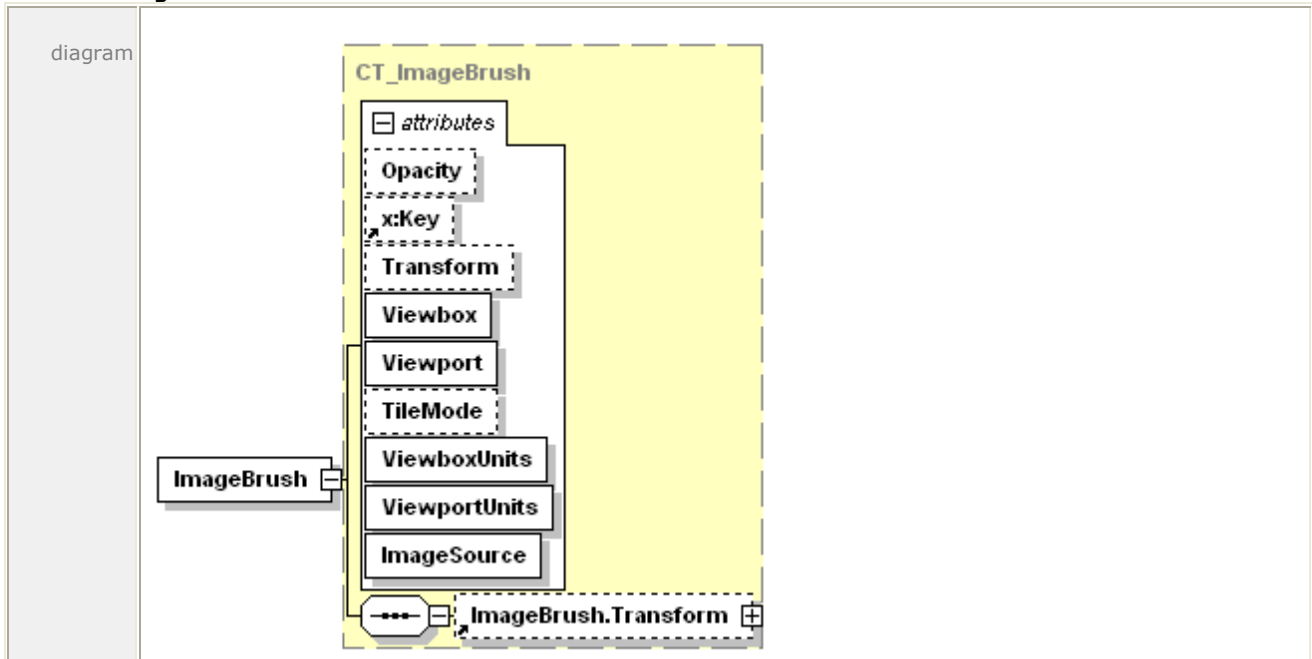
6 element **GradientStop**

diagram	<p>The diagram shows a box labeled 'GradientStop' connected by a dashed line to a box labeled 'attributes'. The 'attributes' box contains two sub-boxes: 'Color' and 'Offset'. A yellow dashed box labeled 'CT_GradientStop' encompasses the 'attributes' box.</p>																		
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><u>ST_Color</u></td> <td>required</td> <td></td> <td></td> <td>Specifies the gradient stop color.</td> </tr> <tr> <td>Offset</td> <td><u>ST_Double</u></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	<u>ST_Color</u>	required			Specifies the gradient stop color.	Offset	<u>ST_Double</u>	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	<u>ST_Color</u>	required			Specifies the gradient stop color.														
Offset	<u>ST_Double</u>	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.																		

7 For more information, see §13.7.

1 **19.24 ImageBrush**

2 element **ImageBrush**



attributes	Name	Type	Use	Default	Fixed	Annotation
	Opacity		ST_ZeroOne		1.0	
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		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 the local effective render transform.
Viewbox		ST_ViewBox	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	ST_ViewBox	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	ST_TileMode		None		Specifies how contents will be tiled in the filled region. Valid values are None, Tile, FlipX, FlipY, and FlipXY.
ViewboxUnits	ST_ViewUnits	required		Absolute	Specifies the relationship of the viewbox coordinates to the containing coordinate space.
ViewportUnits	ST_ViewUnits	required		Absolute	Specifies the relationship of the viewport coordinates to the containing coordinate space.
ImageSource	ST_UriCtxBmp	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.

2 19.25 ImageBrush.Transform


3 element **ImageBrush.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 viewport for the brush is transformed using the local effective render transform.

1 For more information, see §14.4.6~~14.4~~.

2 **19.26 Intent**

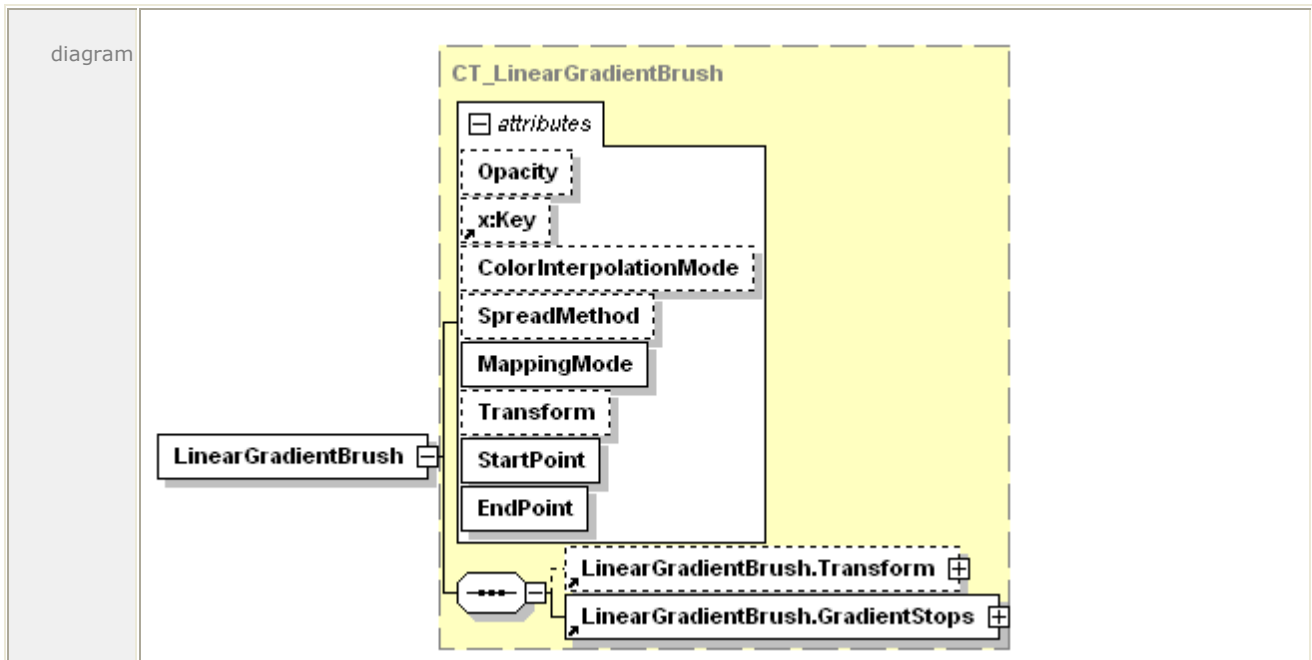
3 element **SignatureDefinitionType/Intent**

diagram	
annotation	A string that represents the intent to which the signing party agrees when signing the document.

4 For more information, see §17.2.2.4.

5 **19.27 LinearGradientBrush**

6 element **LinearGradientBrush**



attributes	Name	Type	Use	Default	Fixed	Annotation
	Opacity	<u>ST_ZeroOne</u>		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	ST_Spread Method		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	ST_Mapping Mode	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	ST_RscRef Matrix				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	ST_Point	required			Specifies the starting point of the linear gradient.
EndPoint	ST_Point	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 §13.5 and §15.

2 **19.28 LinearGradientBrush.GradientStops**

3 element **LinearGradientBrush.GradientStops**

diagram	
annotation	Holds a sequence of GradientStop elements.

4 For more information, see §13.5.2.

5 **19.29 LinearGradientBrush.Transform**

6 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.

7 For more information, see §14.4.8.

8 **19.30 LinkTarget**

9 element **LinkTarget**

diagram													
attributes	<table border="1" style="width: 100%;"> <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>Name</td> <td><u>ST_Name</u></td> <td>required</td> <td></td> <td></td> <td>Contains a string value that identifies the current element as a named, addressable point in the document for the purpose of</td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	Annotation	Name	<u>ST_Name</u>	required			Contains a string value that identifies the current element as a named, addressable point in the document for the purpose of
Name	Type	Use	Default	Fixed	Annotation								
Name	<u>ST_Name</u>	required			Contains a string value that identifies the current element as a named, addressable point in the document for the purpose of								

	hyperlinking.
annotation	Specifies an addressable point on the page.

1 For more information, see §10.2.3.

2 19.31 ListItemStructure

3 element **ListItemStructure**

diagram						
attributes	Name	Type	Use	Default	Fixed	Annotation
	Marker	<u>ST_NameUnique</u>	optional			The named element that represents the marker for this list items, such as a bullet, number, or image.
annotation	Describes a single structural block. These structural blocks are grouped together in a list.					

4 For more information, see §16.1.2.11.

5 19.32 ListStructure

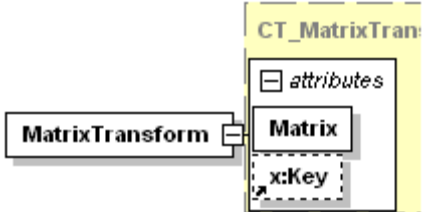
6 element **ListStructure**

diagram						
annotation	Contains a collection of items that are group together in a list.					

1 For more information, see §16.1.2.10.

2 **19.33 MatrixTransform**

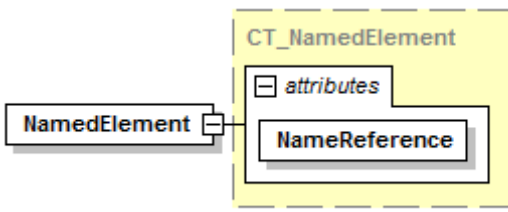
3 element **MatrixTransform**

diagram	 <p>The diagram shows a box labeled 'MatrixTransform' with a small square icon on its right side. A line connects this icon to a larger box labeled 'CT_MatrixTran' which contains a sub-box 'attributes' with a minus sign icon, a box 'Matrix', and a dashed box 'x:Key' with a mouse cursor icon.</p>					
attributes	Name	Type	Use	Default	Fixed	Annotation
	Matrix	ST_Matrix	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.					

4 For more information, see §14.4.1.

5 **19.34 NamedElement**

6 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 'CT_NamedElement' which contains a sub-box 'attributes' with a minus sign icon and a box 'NameReference'.</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.					

1 For more information, see §16.1.2.13.

2 **19.35 OutlineEntry**

3 element **OutlineEntry**

diagram						
attributes	Name	Type	Use	Default	Fixed	Annotation
	OutlineLevel	<u>ST_IntGEOne</u>	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 can 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.					

4 For more information, see §16.1.1.4.

5 **19.36 PageContent**

6 element **PageContent**

<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>Source</td> <td>xs:anyURI</td> <td>required</td> <td></td> <td></td> <td>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].</td> </tr> <tr> <td>Width</td> <td>ST_GEOne</td> <td></td> <td></td> <td></td> <td>TypicalThe width of the pages contained in the page content.</td> </tr> <tr> <td>Height</td> <td>ST_GEOne</td> <td></td> <td></td> <td></td> <td>TypicalThe height of the pages contained in the page content.</td> </tr> </tbody> </table>	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 The width of the pages contained in the page content.	Height	ST_GEOne				Typical The height of the pages contained in the page content.
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 The width of the pages contained in the page content.																				
Height	ST_GEOne				Typical The height of the pages contained in the page content.																				
<p>annotation</p>	<p>Defines a reference from a fixed document to a part that contains a <FixedPage> element.</p>																								

1 For more information, see §10.2.1.

2 19.37 PageContent.LinkTargets

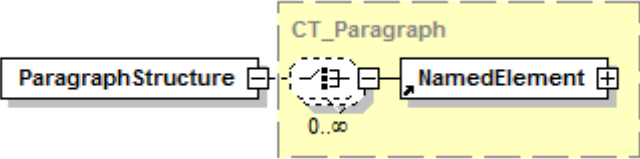
3 element **PageContent.LinkTargets**

<p>diagram</p>						
<p>annotation</p>	<p>Contains a collection of <LinkTarget> elements, each of which is addressable via hyperlink.</p>					

4 For more information, see §10.2.2.

5 19.38 ParagraphStructure

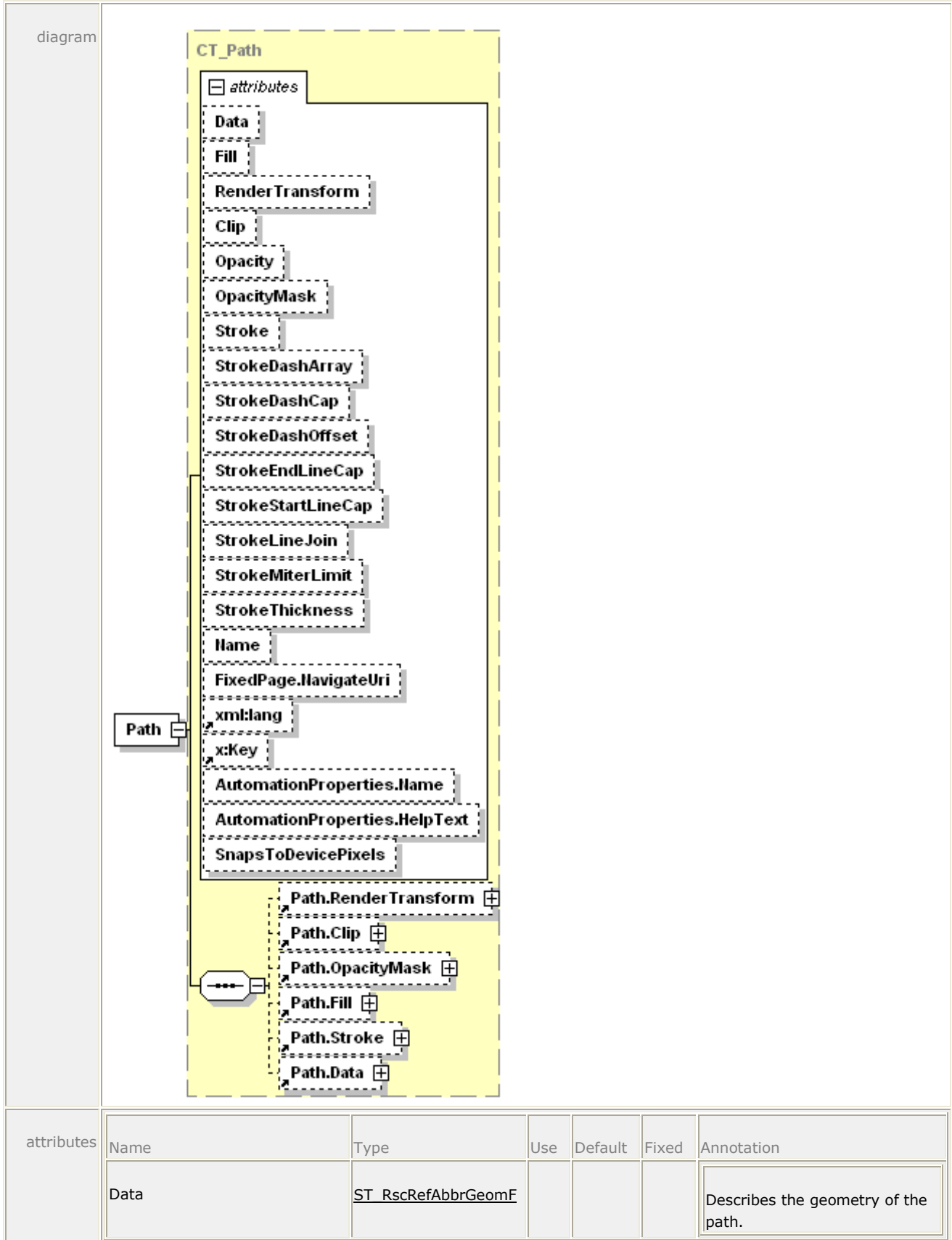
6 element **ParagraphStructure**

diagram	 <p>The diagram shows a class ParagraphStructure with a composition relationship (indicated by a solid line with an open arrowhead) to a class CT_Paragraph. The CT_Paragraph class is highlighted with a yellow dashed border and contains a composition relationship to the NamedElement class. The multiplicity for CT_Paragraph is $0..∞$.</p>
annotation	Contains the named elements that constitute a single paragraph.

- 1 For more information, see §16.1.2.5.

2 **19.39 Path**

- 3 element **Path**



Fill	ST_RscRefColor				Describes the brush used to paint the geometry specified by the Data property of the path.
RenderTransform	ST_RscRefMatrix				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	ST_RscRefAbbrGeomF				Limits the rendered region of the element.
Opacity	ST_ZeroOne		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	ST_RscRef				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	ST_RscRefColor				Specifies the brush used to draw the stroke.
StrokeDashArray	ST_EvenArrayPos				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	ST_DashCap		Flat		Specifies how the ends of each dash are drawn. Valid values are Flat, Round, Square, and

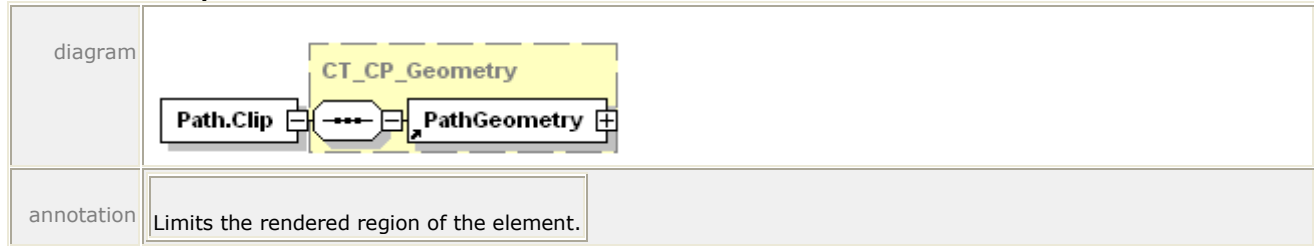
					Triangle.
StrokeDashOffset	<u>ST_Double</u>		0.0		Adjusts the start point for repeating the dash array pattern. If this value is omitted, the dash array aligns with the origin of the stroke. Values are specified as multiples of the stroke thickness.
StrokeEndLineCap	<u>ST_LineCap</u>		Flat		Defines the shape of the end of the last dash in a stroke. Valid values are Flat, Square, Round, and Triangle.
StrokeStartLineCap	<u>ST_LineCap</u>		Flat		Defines the shape of the beginning of the first dash in a stroke. Valid values are Flat, Square, Round, and Triangle.
StrokeLineJoin	<u>ST_LineJoin</u>		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	<u>ST_GEOne</u>		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	<u>ST_GEZero</u>		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 For more information, see §11.1 and §11.2.3.

1 19.40 Path.Clip

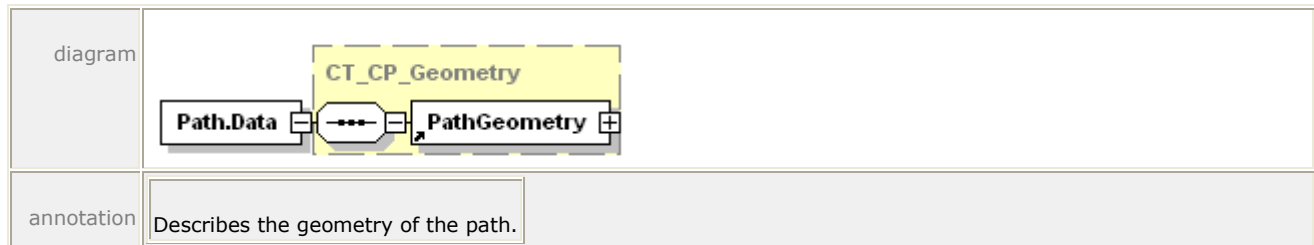
2 element **Path.Clip**



3 For more information, see §14.3.2 ~~14.3~~.

4 19.41 Path.Data

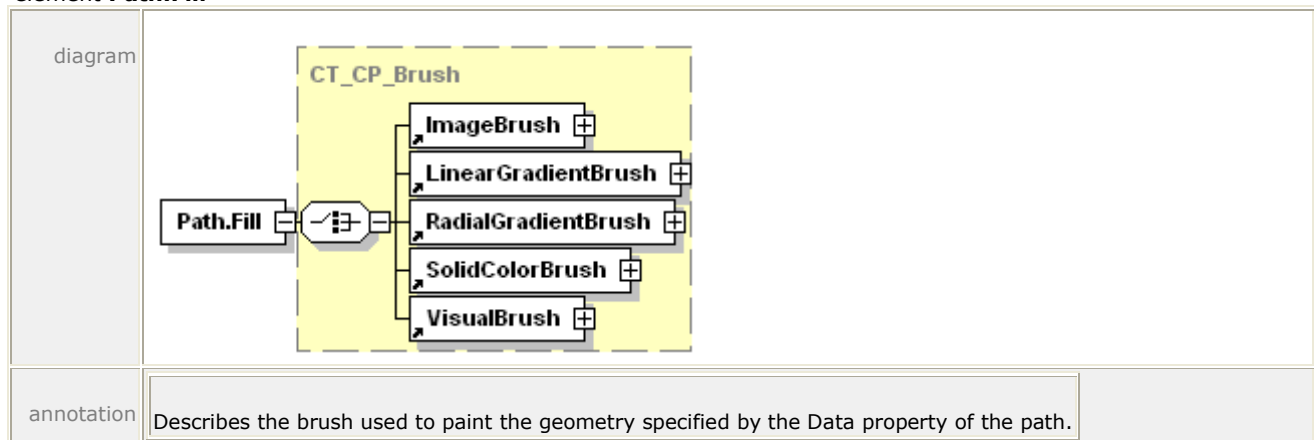
5 element **Path.Data**



6 For more information, see §11.1.1.

7 19.42 Path.Fill

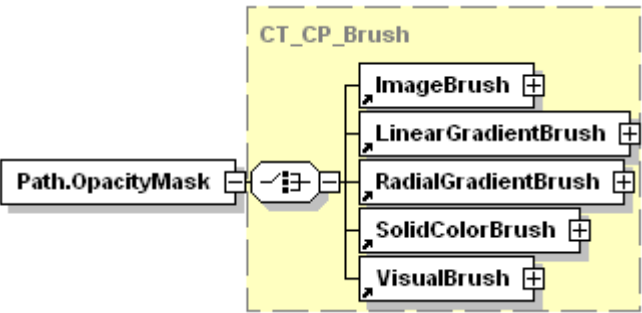
8 element **Path.Fill**



9 For more information, see §11.1.2.

10 19.43 Path.OpacityMask

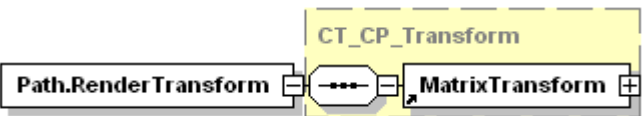
11 element **Path.OpacityMask**

<p>diagram</p>	
<p>annotation</p>	<p>Specifies the 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.</p>

1 For more information, see §14.5.2.

2 19.44 Path.RenderTransform

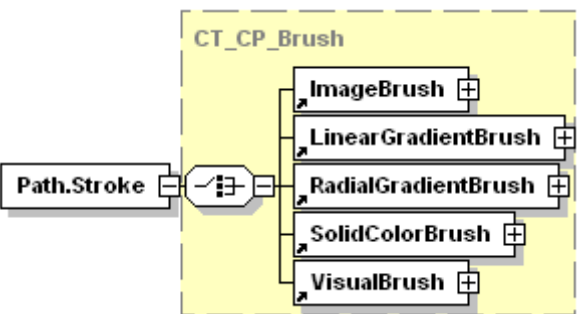
3 element **Path.RenderTransform**

<p>diagram</p>	
<p>annotation</p>	<p>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.</p>

4 For more information, see §14.4.3~~14.4~~.

5 19.45 Path.Stroke

6 element **Path.Stroke**

<p>diagram</p>	
<p>annotation</p>	<p>Specifies the brush used to draw the stroke.</p>

7

1 For more information, see §11.1.3.

2 **19.46 PathFigure**

3 element **PathFigure**

diagram

attributes

Name	Type	Use	Default	Fixed	Annotation
IsClosed	ST_Boolean		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	ST_Point	required			Specifies the starting point for the first segment of the path figure.
IsFilled	ST_Boolean		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.

5 **19.47 PathGeometry**

6 element **PathGeometry**

<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>Figures</td> <td><u>ST_AbbrGeom</u></td> <td></td> <td></td> <td></td> <td>Describes the geometry of the path.</td> </tr> <tr> <td>FillRule</td> <td><u>ST_FillRule</u></td> <td></td> <td>EvenOdd</td> <td></td> <td>Specifies how the intersecting areas of geometric shapes are combined to form a region. Valid values are EvenOdd and NonZero.</td> </tr> <tr> <td>Transform</td> <td><u>ST_RscRefMatrix</u></td> <td></td> <td></td> <td></td> <td>Specifies the local matrix transformation that is applied to all child and descendant elements of the path geometry before it is used for filling, clipping, or stroking.</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 [M4.2].</td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	Annotation	Figures	<u>ST_AbbrGeom</u>				Describes the geometry of the path.	FillRule	<u>ST_FillRule</u>		EvenOdd		Specifies how the intersecting areas of geometric shapes are combined to form a region. Valid values are EvenOdd and NonZero.	Transform	<u>ST_RscRefMatrix</u>				Specifies the local matrix transformation that is applied to all child and descendant elements of the path geometry before it is used for filling, clipping, or stroking.	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 [M4.2].					
Name	Type	Use	Default	Fixed	Annotation																															
Figures	<u>ST_AbbrGeom</u>				Describes the geometry of the path.																															
FillRule	<u>ST_FillRule</u>		EvenOdd		Specifies how the intersecting areas of geometric shapes are combined to form a region. Valid values are EvenOdd and NonZero.																															
Transform	<u>ST_RscRefMatrix</u>				Specifies the local matrix transformation that is applied to all child and descendant elements of the path geometry before it is used for filling, clipping, or stroking.																															
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 [M4.2].																															
<p>annotation</p>	<p>Contains a set of <PathFigure> elements.</p>																																			

1 For more information, see §11.2.1.1.

3 19.48 PathGeometry.Transform

4 element **PathGeometry.Transform**

<p>diagram</p>						
<p>annotation</p>	<p>Specifies the local matrix transformation that is applied to all child and descendant elements of the path</p>					

geometry before it is used for filling, clipping, or stroking.

1 For more information, see §14.4.5.

2 19.49 PolyBezierSegment

3 element **PolyBezierSegment**

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

4

5 19.50 PolyLineSegment

6 element **PolyLineSegment**

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>Points</td> <td>ST_Points</td> <td>required</td> <td></td> <td></td> <td>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 can appear. Coordinate pairs are separated from other coordinate pairs by whitespace.</td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	Annotation	Points	ST_Points	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 can appear. Coordinate pairs are separated from other coordinate pairs by whitespace.					
Name	Type	Use	Default	Fixed	Annotation													
Points	ST_Points	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 can appear. Coordinate pairs are separated from other coordinate pairs by whitespace.													

	IsStroked	<u>ST_Boolean</u>		true		Specifies whether the stroke for this segment of the path is drawn. Can be true or false.
1	annotation	Specifies a set of points between which lines are drawn.				

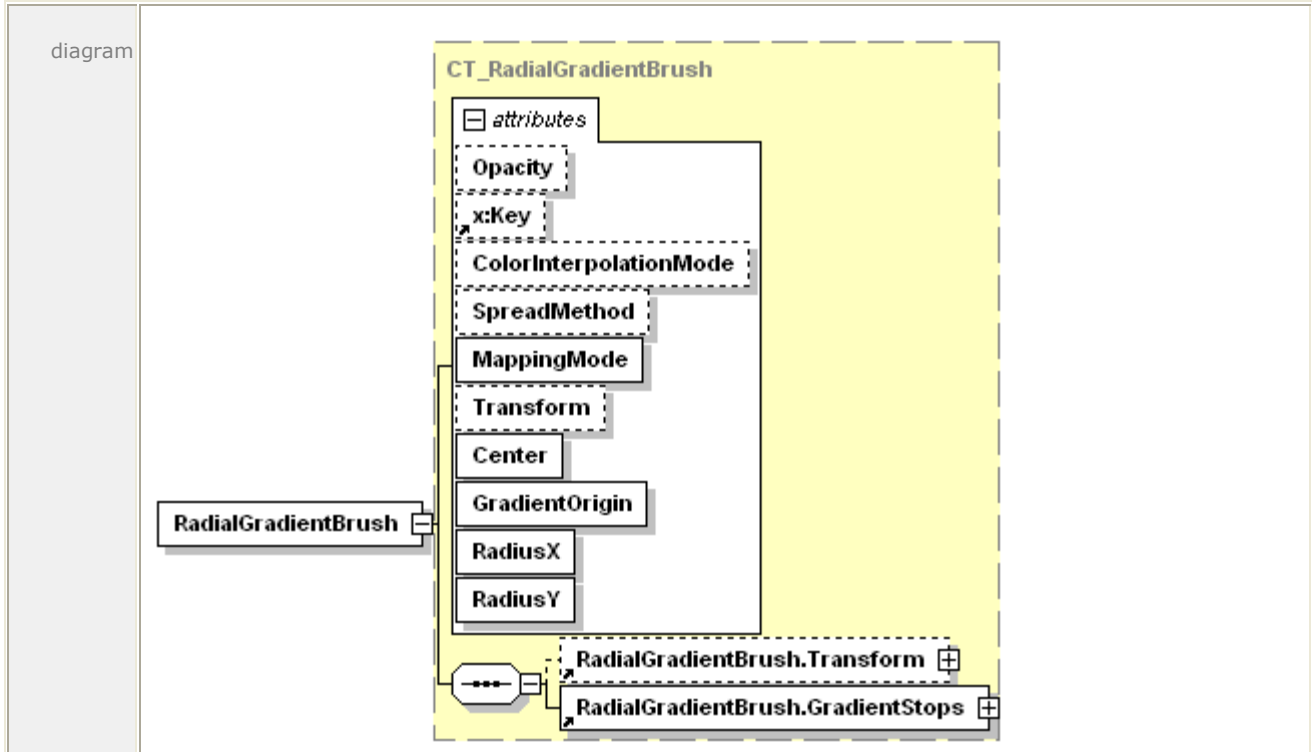
2 **19.51 PolyQuadraticBezierSegment**

3 element **PolyQuadraticBezierSegment**

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

5 **19.52 RadialGradientBrush**

6 element **RadialGradientBrush**




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	ST_ClrIntMode		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	ST_Spread Method		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	ST_MappingMode	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	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 ellipse defined by the center, gradient origin, x radius, and y radius values is transformed using the local effective render transform.
	Center	ST_Point	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	ST_Point	required			Specifies the origin point of the radial gradient.
	RadiusX	ST_GEZero	required			Specifies the radius in the x dimension of the ellipse which defines the radial gradient.
	RadiusY	ST_GEZero	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 §13.6 and §15.

1 **19.53 RadialGradientBrush.GradientStops**


2 element **RadialGradientBrush.GradientStops**

diagram	
annotation	Holds a sequence of <GradientStop> elements.

3 For more information, see §13.6.2.

4 **19.54 RadialGradientBrush.Transform**

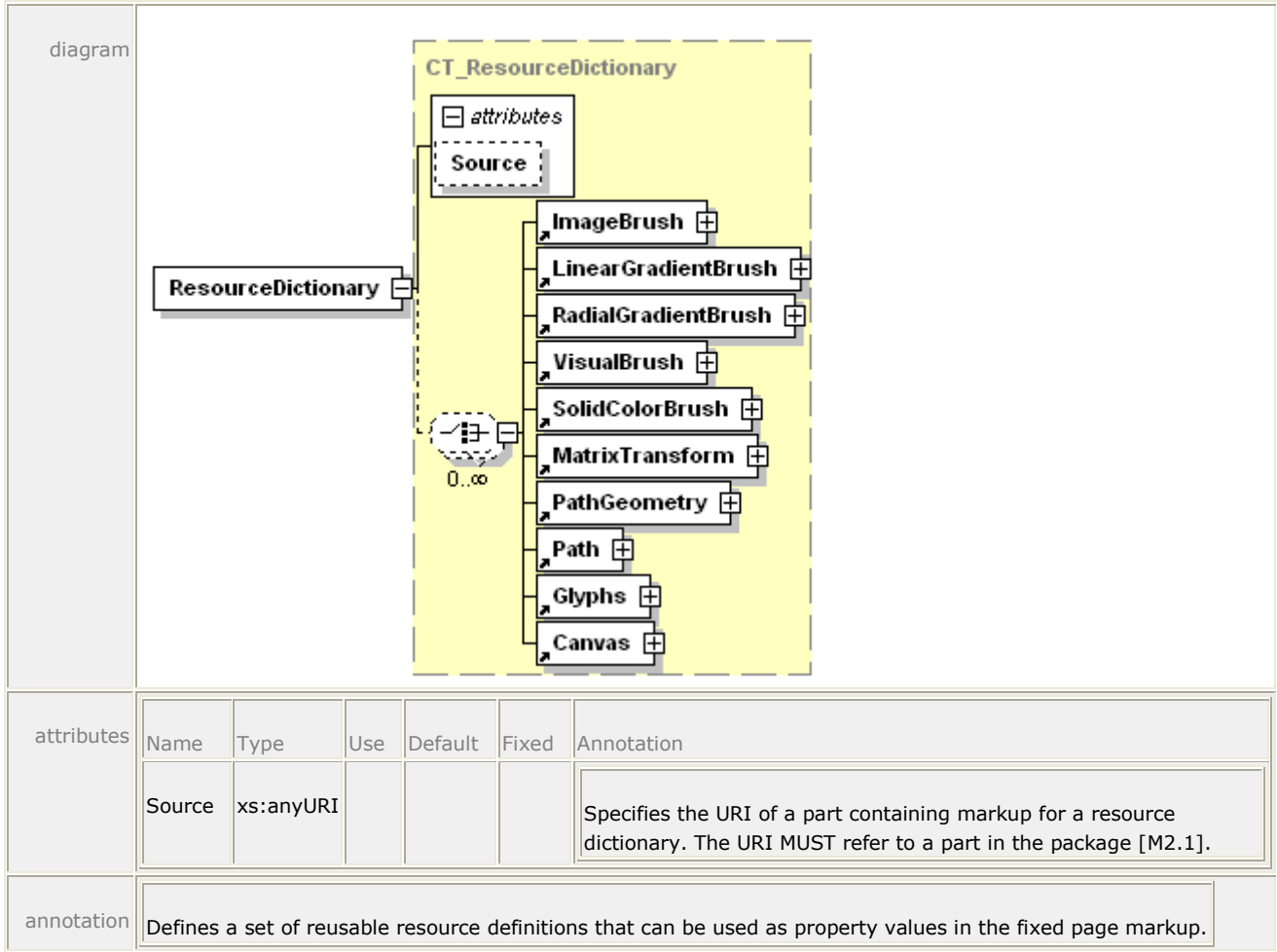
5 element **RadialGradientBrush.Transform**

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 center, gradient origin, x radius, and y radius are transformed using the local effective render transform.

6 For more information, see §14.4.9.

7 **19.55 ResourceDictionary**

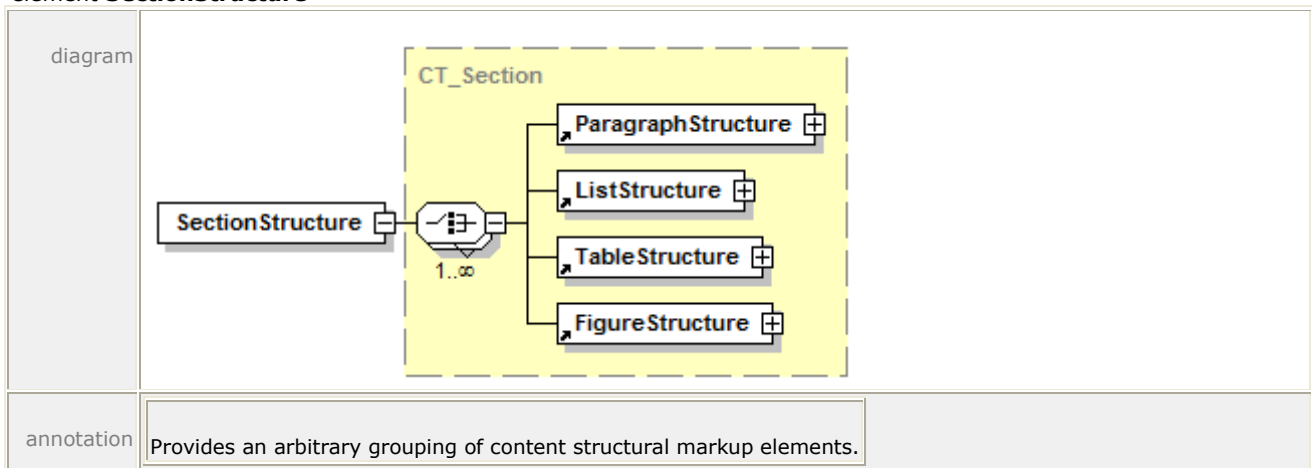
8 element **ResourceDictionary**



1 For more information, see §14.2.3 ~~14.2~~.

2 19.56 SectionStructure


3 element **SectionStructure**



4 For more information, see §16.1.2.4.

1 **19.57 SignBy**

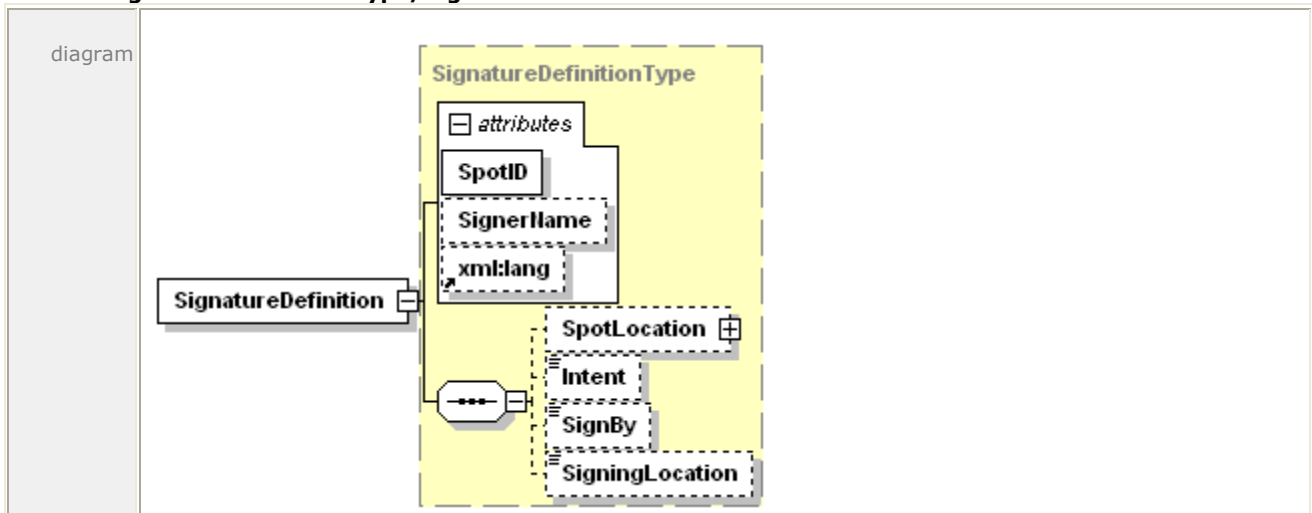
2 element **SignatureDefinitionType/SignBy**

diagram	
annotation	The date and time by which the requested party is to sign the XPS Document.

3 For more information, see §17.2.2.5.

4 **19.58 SignatureDefinition**

5 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.
------------	--------------------------------

6 For more information, see §17.2.2.2.

1 **19.59 SignatureDefinitions**

2 element **SignatureDefinitions**

diagram	
annotation	The root element for the SignatureDefinitions part.

3 For more information, see §17.2.2.1.

4 **19.60 SigningLocation**

5 element **SignatureDefinitionType/SigningLocation**

diagram	
annotation	The legal location where the document is signed.

6 For more information, see §17.2.2.6.

7 **19.61 SolidColorBrush**

8 element **SolidColorBrush**

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>Opacity</td> <td><u>ST_ZeroOne</u></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 [M6.1].</td> </tr> </tbody> </table>	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 [M6.1].
	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 [M6.1].														

	Color	<u>ST_Color</u>	required			Specifies the color for filled elements.
annotation	Fills defined geometric regions with a solid color.					

1 For more information, see §13.1.

2 **19.62 SpotLocation**

3 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.					

4 For more information, see§17.2.2.3.

5 **19.63 Story**

6 element **Story**

diagram						
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.					

1 For more information, see §16.1.1.5.

2 **19.64 StoryBreak**

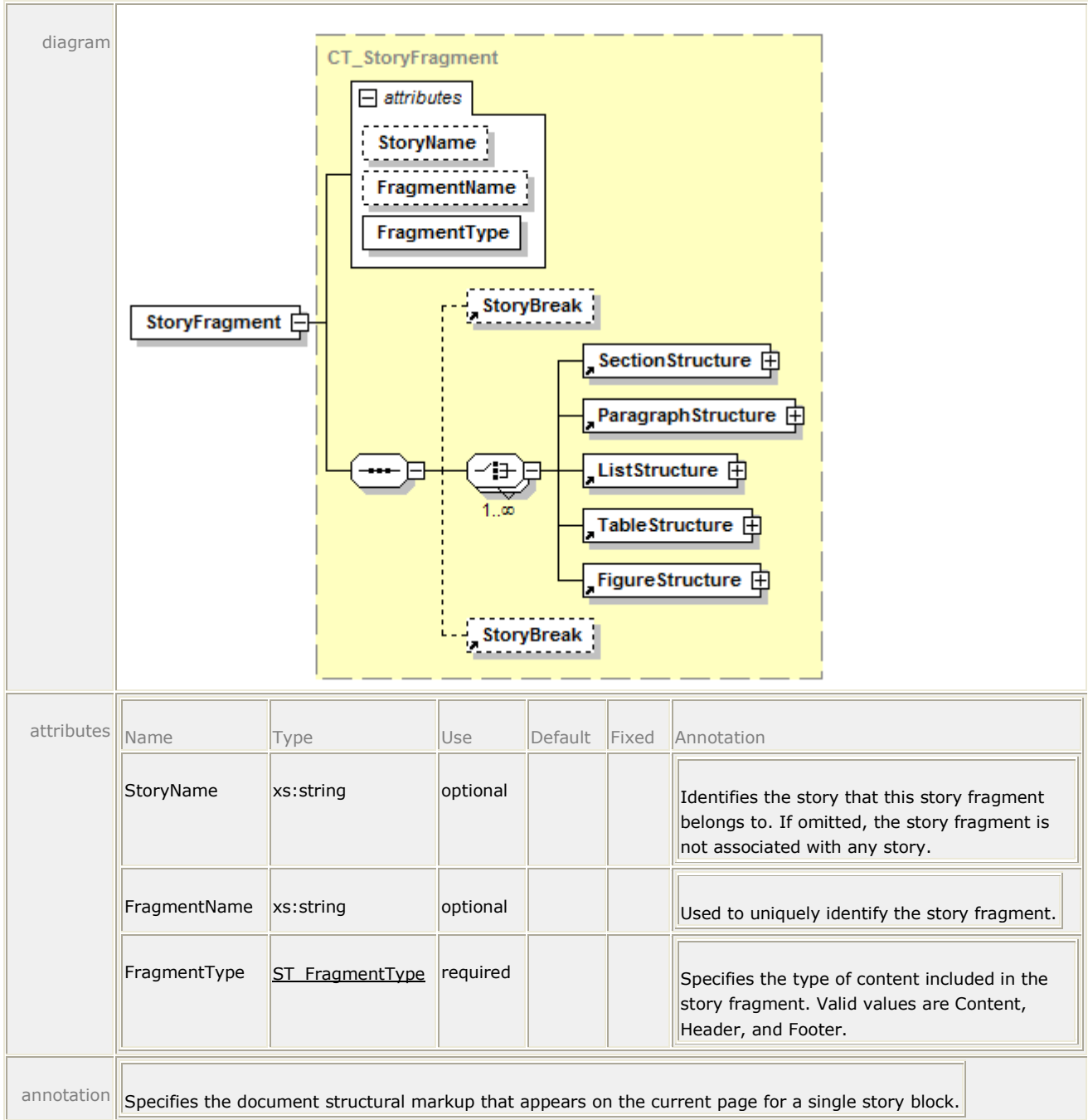
3 element **StoryBreak**

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>.

4 For more information, see §16.1.2.3.

5 **19.65 StoryFragment**

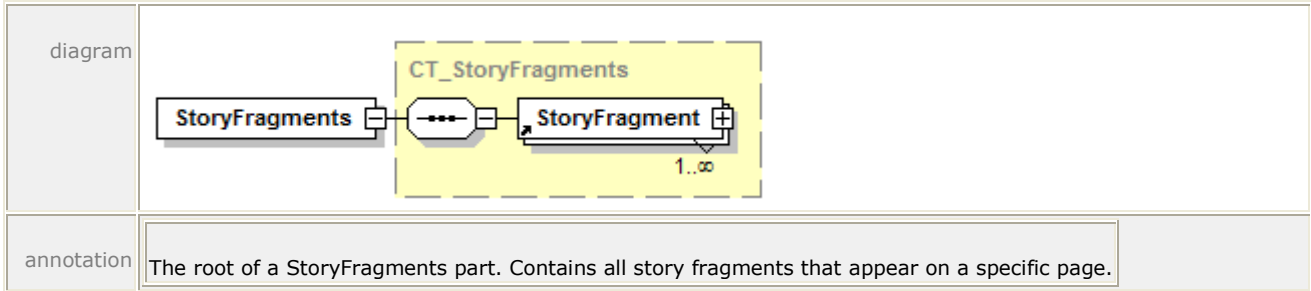
6 element **StoryFragment**



1 For more information, see §16.1.2.2.

2 19.66 StoryFragments

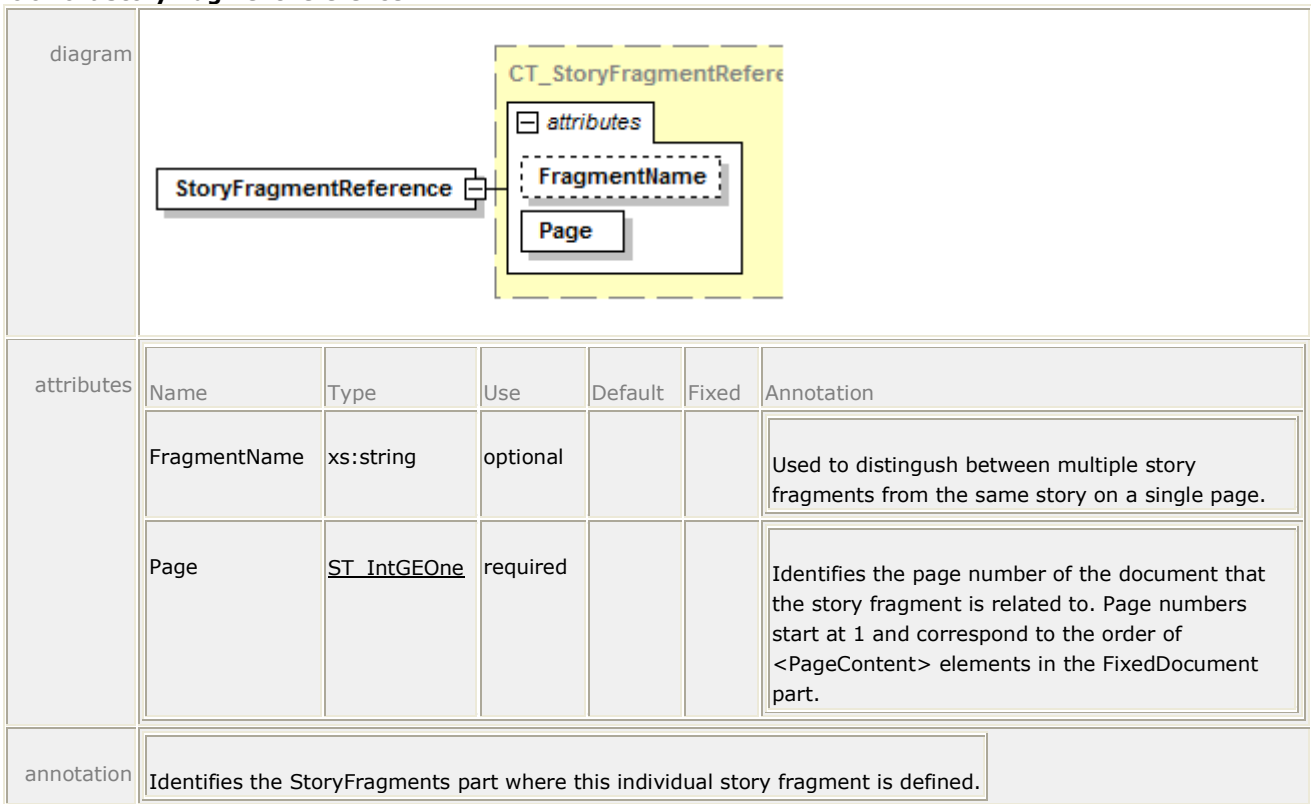
3 element **StoryFragments**



1 For more information, see §16.1.2.1.

2 19.67 StoryFragmentReference

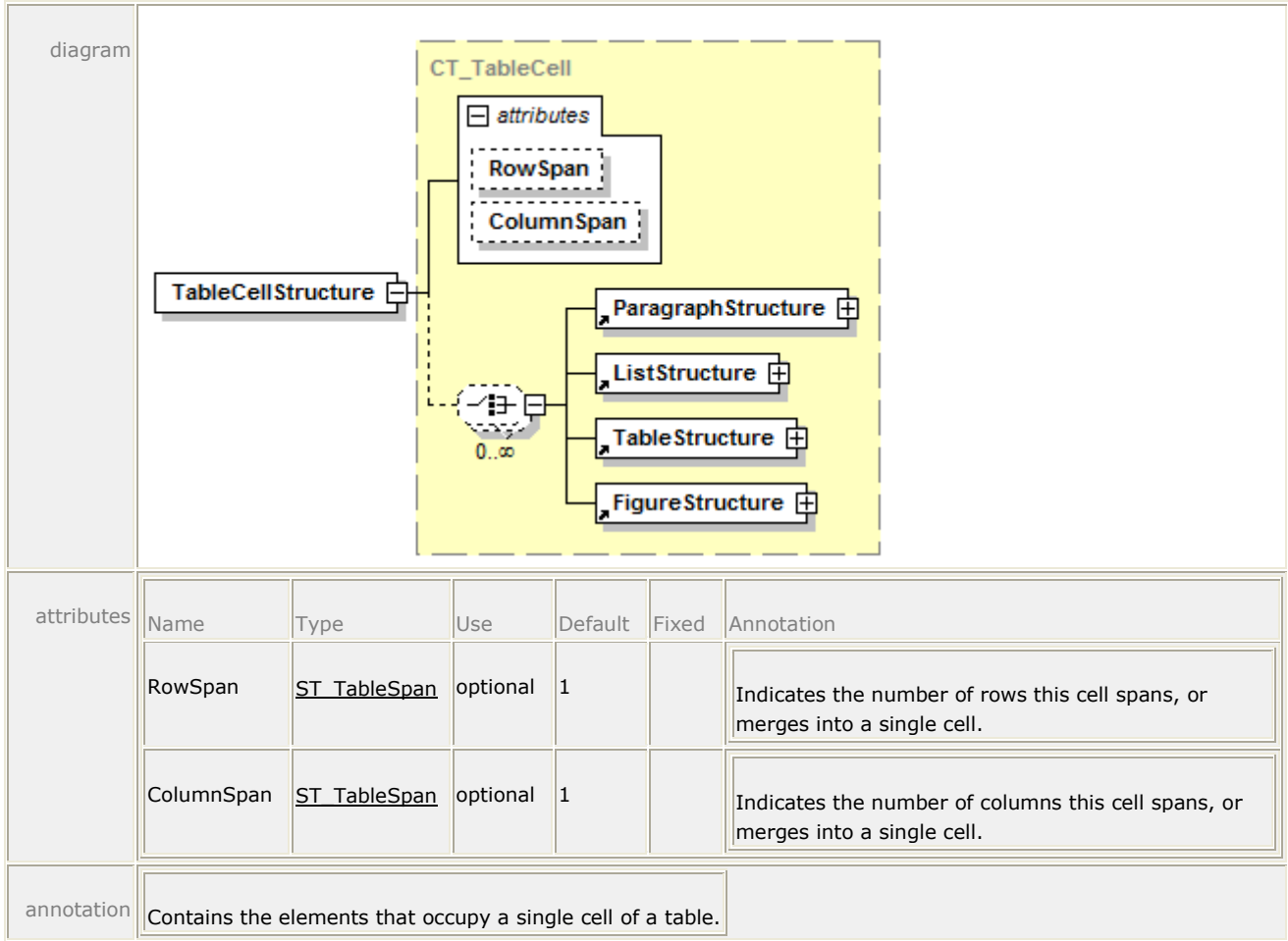
3 element **StoryFragmentReference**



4 For more information, see §16.1.1.6.

5 19.68 TableCellStructure

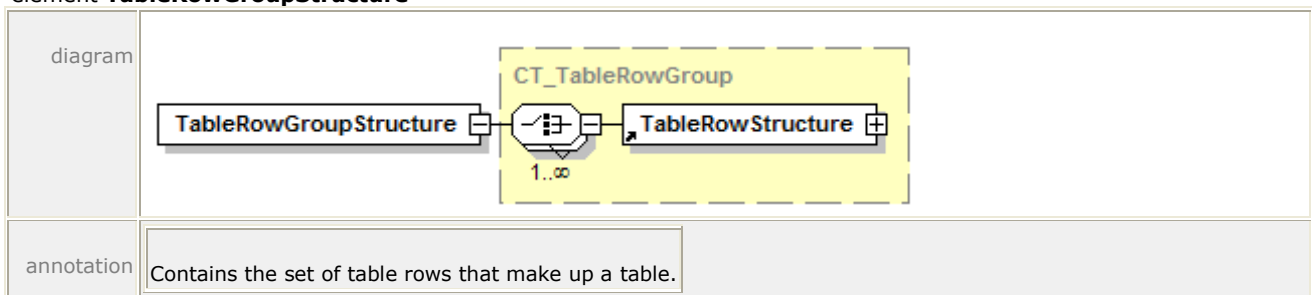
6 element **TableCellStructure**



1 For more information, see §16.1.2.9.

2 19.69 TableRowGroupStructure

3 element **TableRowGroupStructure**



4 For more information, see §16.1.2.7.

5 19.70 TableRowStructure

6 element **TableRowStructure**

diagram	<p>The diagram shows a class CT_TableRow (indicated by a dashed yellow box) containing two classes: TableRowStructure and TableCellStructure. A composition relationship is shown between them, with a multiplicity of 1..∞ at the TableCellStructure end.</p>
annotation	<p>Contains the set of table cells that make up a row of a table.</p>

1 For more information, see §16.1.2.8.

2 19.71 TableStructure

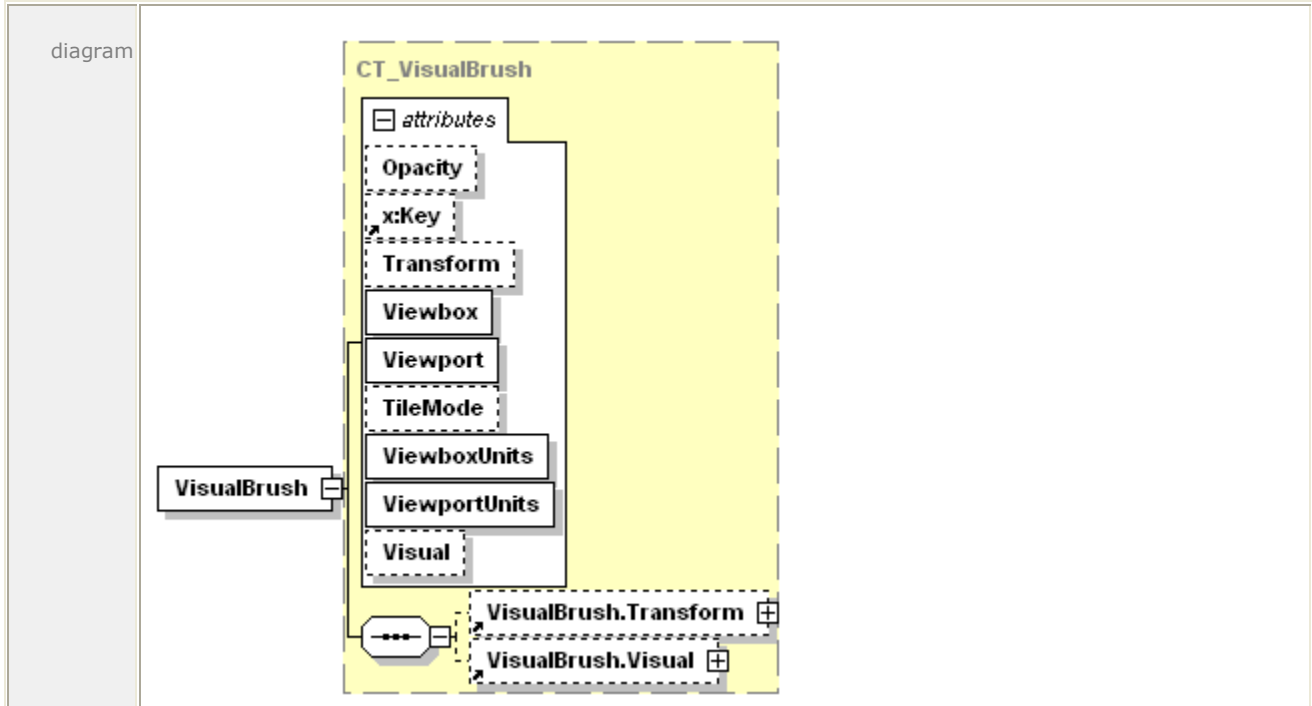
3 element **TableStructure**

diagram	<p>The diagram shows a class CT_Table (indicated by a dashed yellow box) containing two classes: TableStructure and TableRowGroupStructure. A composition relationship is shown between them, with a multiplicity of 1..∞ at the TableRowGroupStructure end.</p>
annotation	<p>Contains a complete definition of a table in the XPS Document.</p>

4 For more information, see §16.1.2.6.

5 19.72 VisualBrush

6 element **VisualBrush**



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	ST_ViewBox	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	ST_ViewBox	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	ST_TileMode		None		Specifies how contents will be tiled in the filled region. Valid values are None, Tile, FlipX, FlipY, and FlipXY.
ViewboxUnits	ST_ViewUnits	required		Absolute	Specifies the relationship of the viewbox coordinates to the containing coordinate space.
ViewportUnits	ST_ViewUnits	required		Absolute	Specifies the relationship of the viewport coordinates to the containing coordinate space.
Visual	ST_RscRef				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 can 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.

2 19.73 VisualBrush.Transform

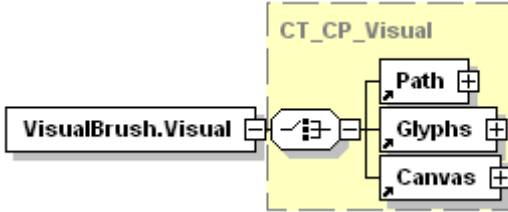
3 element **VisualBrush.Transform**

diagram	<pre> graph LR V[VisualBrush.Transform] --- M[MatrixTransform] M --- CT[CT_CP_Transform] </pre>
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.

1 For more information, see §14.4.7~~14.4~~.

2 **19.74 VisualBrush.Visual**

3 element **VisualBrush.Visual**

<p>diagram</p>	 <p>The diagram shows a class VisualBrush.Visual on the left. A line connects it to a class CT_CP_Visual on the right, which is enclosed in a dashed yellow box. Inside CT_CP_Visual, there are three sub-elements: Path, Glyphs, and Canvas, each with a small square icon and a plus sign next to it.</p>
<p>annotation</p>	<p>Specifies a <Path> element, <Glyphs> element, or <Canvas> element used to draw the brush's source contents.</p>

4
5
6

1 A. Schemas – W3C XML-~~Schema~~

2 A.1 Signature Definitions-~~W3C-Schema~~

3 The schema shown below is also provided in electronic form as a file named
4 SignatureDefinitions.xsd, which is contained in an accompanying zip archive named "XPS WC3
5 Schemas.zip". If discrepancies exist between the representation as published below and the
6 corresponding electronic version, the published version below is the definitive version.

```

7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
<?xml version="1.0" encoding="utf-8"?>
<xs:schema targetNamespace="http://schemas.microsoft.com/xps/2005/06/signature-
definitions" xmlns="http://schemas.microsoft.com/xps/2005/06/signature-definitions"
xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
blockDefault="#all">
  <xs:import namespace="http://www.w3.org/XML/1998/namespace" />
  <xs:complexType name="SignatureDefinitionsType">
    <xs:sequence>
      <xs:element name="SignatureDefinition" type="SignatureDefinitionType"
minOccurs="1" maxOccurs="unbounded">
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  <xs:complexType name="SpotLocationType">
    <xs:attribute name="PageURI" type="xs:anyURI" use="required">
      </xs:attribute>
    <xs:attribute name="StartX" type="xs:double" use="required">
      </xs:attribute>
    <xs:attribute name="StartY" type="xs:double" use="required">
      </xs:attribute>
    </xs:complexType>
  <xs:complexType name="SignatureDefinitionType">
    <xs:sequence>
      <xs:element name="SpotLocation" type="SpotLocationType" minOccurs="0">
        </xs:element>
      <xs:element name="Intent" type="xs:string" minOccurs="0">
        </xs:element>
      <xs:element name="SignBy" type="xs:dateTime" minOccurs="0">
        </xs:element>
      <xs:element name="SigningLocation" type="xs:string" minOccurs="0">
        </xs:element>
      </xs:sequence>
    <xs:attribute name="SpotID" type="xs:ID" use="required">
      </xs:attribute>
    <xs:attribute name="SignerName" type="xs:string">
      </xs:attribute>
    <xs:attribute ref="xml:lang">

```

```
42     </xs:attribute>
43 </xs:complexType>
44
45 <xs:element name="SignatureDefinitions" type="SignatureDefinitionsType">
46     </xs:element>
47 </xs:schema>
```

1 **A.2 XPS Document** ~~W3C Schema~~

2 The schema shown below is also provided in electronic form as a file named XPSDocument.xsd,
3 which is contained in an accompanying zip archive named "XPS WC3 Schemas.zip". If
4 discrepancies exist between the representation as published below and the corresponding
5 electronic version, the published version below is the definitive version

```
6
7 <?xml version="1.0" encoding="utf-8"?>
8 <xs:schema targetNamespace="http://schemas.microsoft.com/xps/2005/06"
9 xmlns="http://schemas.microsoft.com/xps/2005/06"
10 xmlns:mstns="http://tempuri.org/XMLSchema.xsd"
11 xmlns:xs="http://www.w3.org/2001/XMLSchema"
12 xmlns:x="http://schemas.microsoft.com/xps/2005/06/resourcedictionary-key"
13 elementFormDefault="qualified" blockDefault="#all">
14
15     <xs:import namespace=
16     "http://schemas.microsoft.com/xps/2005/06/resourcedictionary-key" />
17     <xs:import namespace="http://www.w3.org/XML/1998/namespace" />
18
19     <!-- Names used for types and groups:
20
21         ST_*         simpleType
22         CT_*         complexType
23         G_*          group
24         AG_*         attributeGroup
25
26     -->
27
28     <!-- Individual real number patterns
29     All patterns using numbers now use <whitespace value="collapse">.
30     As a result, any whitespace in the pattern can be expressed as:
31     mandatory whitespace, one or more: " "
32     optional whitespace, zero or more: " ?"
33
34     For better readability, each pattern using numbers is also described in a comment
35     using
36     one of the following pattern designators.
37
38     The actual patterns are generated by replacement by the schema publication process.
39     -->
40     <!--DEFINE [pint]         "([1-9][0-9]*)" -->
41     <!--DEFINE [uint]        "([0-9]+)" -->
42     <!--DEFINE [dec]         "(\\-?(([0-9]+(\\. [0-9]+)?)|(\\. [0-9]+)))" -->
43     <!--DEFINE [rn]         "((\\-|\\+)?([0-9]+(\\. [0-9]+)?)|(\\. [0-9]+))((e|E)(\\-|\\+)?[0-
44     9]+)?)" -->
45     <!--DEFINE [prn]        "(\\+?(([0-9]+(\\. [0-9]+)?)|(\\. [0-9]+))((e|E)(\\-|\\+)?[0-
46     9]+)?)" -->
47     <!--DEFINE [scs]        "( ? , ?)" -->
48
49     <!-- Complex Types -->
```

```

46 <xs:complexType name="CT_MatrixTransform">
47   <xs:attributeGroup ref="AG_MatrixTransform" />
48 </xs:complexType>
49
50 <xs:complexType name="CT_SolidColorBrush">
51   <xs:attributeGroup ref="AG_Brush" />
52   <xs:attributeGroup ref="AG_SolidColorBrush" />
53 </xs:complexType>
54
55 <xs:complexType name="CT_ImageBrush">
56   <xs:sequence>
57     <xs:element ref="ImageBrush.Transform" minOccurs="0" />
58   </xs:sequence>
59   <xs:attributeGroup ref="AG_Brush" />
60   <xs:attributeGroup ref="AG_TileBrush" />
61   <xs:attributeGroup ref="AG_ImageBrush" />
62 </xs:complexType>
63
64 <xs:complexType name="CT_VisualBrush">
65   <xs:sequence>
66     <xs:element ref="VisualBrush.Transform" minOccurs="0" />
67     <xs:element ref="VisualBrush.Visual" minOccurs="0" />
68   </xs:sequence>
69   <xs:attributeGroup ref="AG_Brush" />
70   <xs:attributeGroup ref="AG_TileBrush" />
71   <xs:attributeGroup ref="AG_VisualBrush" />
72 </xs:complexType>
73
74 <xs:complexType name="CT_LinearGradientBrush">
75   <xs:sequence>
76     <xs:element ref="LinearGradientBrush.Transform" minOccurs="0" />
77     <xs:element ref="LinearGradientBrush.GradientStops" />
78   </xs:sequence>
79   <xs:attributeGroup ref="AG_Brush" />
80   <xs:attributeGroup ref="AG_GradientBrush" />
81   <xs:attributeGroup ref="AG_LinearGradientBrush" />
82 </xs:complexType>
83
84 <xs:complexType name="CT_RadialGradientBrush">
85   <xs:sequence>
86     <xs:element ref="RadialGradientBrush.Transform" minOccurs="0" />
87     <xs:element ref="RadialGradientBrush.GradientStops" />
88   </xs:sequence>
89   <xs:attributeGroup ref="AG_Brush" />
90   <xs:attributeGroup ref="AG_GradientBrush" />
91   <xs:attributeGroup ref="AG_RadialGradientBrush" />
92 </xs:complexType>
93
94 <xs:complexType name="CT_GradientStop">
95   <xs:attributeGroup ref="AG_GradientStop" />
96 </xs:complexType>
97
98 <xs:complexType name="CT_PathGeometry">
99   <xs:sequence>
100    <xs:element ref="PathGeometry.Transform" minOccurs="0" />

```

```

101     <xs:element ref="PathFigure" minOccurs="0" maxOccurs="unbounded" />
102   </xs:sequence>
103   <xs:attributeGroup ref="AG_PathGeometry" />
104 </xs:complexType>
105
106 <xs:complexType name="CT_Glyphs">
107   <xs:sequence>
108     <xs:element ref="Glyphs.RenderTransform" minOccurs="0" />
109     <xs:element ref="Glyphs.Clip" minOccurs="0" />
110     <xs:element ref="Glyphs.OpacityMask" minOccurs="0" />
111     <xs:element ref="Glyphs.Fill" minOccurs="0" />
112   </xs:sequence>
113   <xs:attributeGroup ref="AG_Glyphs" />
114 </xs:complexType>
115
116 <xs:complexType name="CT_Path">
117   <xs:sequence>
118     <xs:element ref="Path.RenderTransform" minOccurs="0" />
119     <xs:element ref="Path.Clip" minOccurs="0" />
120     <xs:element ref="Path.OpacityMask" minOccurs="0" />
121     <xs:element ref="Path.Fill" minOccurs="0" />
122     <xs:element ref="Path.Stroke" minOccurs="0" />
123     <xs:element ref="Path.Data" minOccurs="0" />
124   </xs:sequence>
125   <xs:attributeGroup ref="AG_Path" />
126   <xs:attributeGroup ref="AG_AutomationProvider" />
127   <xs:attributeGroup ref="AG_SnapsToDevicePixels" />
128 </xs:complexType>
129
130 <xs:complexType name="CT_PathFigure">
131   <xs:sequence>
132     <xs:choice maxOccurs="unbounded">
133       <xs:element ref="PolyLineSegment" />
134       <xs:element ref="PolyBezierSegment" />
135       <xs:element ref="ArcSegment" />
136       <xs:element ref="PolyQuadraticBezierSegment" />
137     </xs:choice>
138   </xs:sequence>
139   <xs:attributeGroup ref="AG_PathFigure" />
140 </xs:complexType>
141
142 <xs:complexType name="CT_ArcSegment">
143   <xs:attributeGroup ref="AG_ArcSegment" />
144 </xs:complexType>
145
146 <xs:complexType name="CT_PolyQuadraticBezierSegment">
147   <xs:attributeGroup ref="AG_PolyQuadraticBezierSegment" />
148 </xs:complexType>
149
150 <xs:complexType name="CT_PolyLineSegment">
151   <xs:attributeGroup ref="AG_PolyLineSegment" />
152 </xs:complexType>
153
154 <xs:complexType name="CT_PolyBezierSegment">
155   <xs:attributeGroup ref="AG_PolyBezierSegment" />

```

```

156 </xs:complexType>
157
158 <xs:complexType name="CT_Canvas">
159   <xs:sequence>
160     <xs:element ref="Canvas.Resources" minOccurs="0" />
161     <xs:element ref="Canvas.RenderTransform" minOccurs="0" />
162     <xs:element ref="Canvas.Clip" minOccurs="0" />
163     <xs:element ref="Canvas.OpacityMask" minOccurs="0" />
164     <xs:choice minOccurs="0" maxOccurs="unbounded">
165       <xs:element ref="Path" />
166       <xs:element ref="Glyphs" />
167       <xs:element ref="Canvas" />
168     </xs:choice>
169   </xs:sequence>
170   <xs:attributeGroup ref="AG_Canvas" />
171   <xs:attributeGroup ref="AG_AutomationProvider" />
172 </xs:complexType>
173
174 <xs:complexType name="CT_ResourceDictionary">
175   <xs:choice minOccurs="0" maxOccurs="unbounded">
176     <xs:element ref="ImageBrush" />
177     <xs:element ref="LinearGradientBrush" />
178     <xs:element ref="RadialGradientBrush" />
179     <xs:element ref="VisualBrush" />
180     <xs:element ref="SolidColorBrush" />
181     <xs:element ref="MatrixTransform" />
182     <xs:element ref="PathGeometry" />
183     <xs:element ref="Path" />
184     <xs:element ref="Glyphs" />
185     <xs:element ref="Canvas" />
186   </xs:choice>
187   <xs:attributeGroup ref="AG_ResourceDictionary" />
188 </xs:complexType>
189
190 <xs:complexType name="CT_FixedPage">
191   <xs:sequence>
192     <xs:element ref="FixedPage.Resources" minOccurs="0" />
193     <xs:choice minOccurs="0" maxOccurs="unbounded">
194       <xs:element ref="Path" />
195       <xs:element ref="Glyphs" />
196       <xs:element ref="Canvas" />
197     </xs:choice>
198   </xs:sequence>
199   <xs:attributeGroup ref="AG_FixedPage" />
200 </xs:complexType>
201
202 <xs:complexType name="CT_FixedDocument">
203   <xs:sequence>
204     <xs:element ref="PageContent" maxOccurs="unbounded" />
205   </xs:sequence>
206 </xs:complexType>
207
208 <xs:complexType name="CT_PageContent">
209   <xs:sequence>
210     <xs:element ref="PageContent.LinkTargets" minOccurs="0" />

```

```

211     </xs:sequence>
212     <xs:attributeGroup ref="AG_PageContent" />
213 </xs:complexType>
214
215 <xs:complexType name="CT_FixedDocumentSequence">
216     <xs:sequence>
217         <xs:element ref="DocumentReference" maxOccurs="unbounded" />
218     </xs:sequence>
219 </xs:complexType>
220
221 <xs:complexType name="CT_DocumentReference">
222     <xs:attributeGroup ref="AG_DocumentReference" />
223 </xs:complexType>
224
225 <xs:complexType name="CT_LinkTarget">
226     <xs:attributeGroup ref="AG_LinkTarget" />
227 </xs:complexType>
228
229 <xs:complexType name="CT_CP_LinkTargets">
230     <xs:sequence>
231         <xs:element ref="LinkTarget" maxOccurs="unbounded" />
232     </xs:sequence>
233 </xs:complexType>
234
235 <xs:complexType name="CT_CP_Transform">
236     <xs:sequence>
237         <xs:element ref="MatrixTransform" />
238     </xs:sequence>
239 </xs:complexType>
240
241 <xs:complexType name="CT_CP_Visual">
242     <xs:choice>
243         <xs:element ref="Path" />
244         <xs:element ref="Glyphs" />
245         <xs:element ref="Canvas" />
246     </xs:choice>
247 </xs:complexType>
248
249 <xs:complexType name="CT_CP_GradientStops">
250     <xs:sequence>
251         <xs:element ref="GradientStop" minOccurs="2" maxOccurs="unbounded" />
252     </xs:sequence>
253 </xs:complexType>
254
255 <xs:complexType name="CT_CP_Geometry">
256     <xs:sequence>
257         <xs:element ref="PathGeometry" />
258     </xs:sequence>
259 </xs:complexType>
260
261 <xs:complexType name="CT_CP_Brush">
262     <xs:choice>
263         <xs:element ref="ImageBrush" />
264         <xs:element ref="LinearGradientBrush" />
265         <xs:element ref="RadialGradientBrush" />

```

```

266         <xs:element ref="SolidColorBrush" />
267         <xs:element ref="VisualBrush" />
268     </xs:choice>
269 </xs:complexType>
270
271 <xs:complexType name="CT_CP_Resources">
272     <xs:sequence minOccurs="0">
273         <xs:element ref="ResourceDictionary" />
274     </xs:sequence>
275 </xs:complexType>
276
277 <!-- Root elements -->
278 <xs:element name="MatrixTransform" type="CT_MatrixTransform">
279     </xs:element>
280
281 <xs:element name="SolidColorBrush" type="CT_SolidColorBrush">
282     </xs:element>
283
284 <xs:element name="ImageBrush" type="CT_ImageBrush">
285     </xs:element>
286
287 <xs:element name="VisualBrush" type="CT_VisualBrush">
288     </xs:element>
289
290 <xs:element name="LinearGradientBrush" type="CT_LinearGradientBrush">
291     </xs:element>
292
293 <xs:element name="RadialGradientBrush" type="CT_RadialGradientBrush">
294     </xs:element>
295
296 <xs:element name="Glyphs" type="CT_Glyphs">
297     </xs:element>
298
299 <xs:element name="Path" type="CT_Path">
300     </xs:element>
301
302 <xs:element name="Canvas" type="CT_Canvas">
303     </xs:element>
304
305 <xs:element name="GradientStop" type="CT_GradientStop">
306     </xs:element>
307
308 <xs:element name="ResourceDictionary" type="CT_ResourceDictionary">
309     </xs:element>
310
311 <xs:element name="PathGeometry" type="CT_PathGeometry">
312     </xs:element>
313
314 <xs:element name="PathFigure" type="CT_PathFigure">
315     </xs:element>
316
317 <xs:element name="PolyLineSegment" type="CT_PolyLineSegment">
318     </xs:element>
319
320 <xs:element name="ArcSegment" type="CT_ArcSegment">

```



```
321     </xs:element>
322
323     <xs:element name="PolyBezierSegment" type="CT_PolyBezierSegment">
324     </xs:element>
325
326     <xs:element name="PolyQuadraticBezierSegment" type="CT_PolyQuadraticBezierSegment">
327     </xs:element>
328
329     <xs:element name="FixedPage" type="CT_FixedPage">
330     </xs:element>
331
332     <xs:element name="FixedDocument" type="CT_FixedDocument">
333     </xs:element>
334
335     <xs:element name="PageContent" type="CT_PageContent">
336     </xs:element>
337
338     <xs:element name="FixedDocumentSequence" type="CT_FixedDocumentSequence">
339     </xs:element>
340
341     <xs:element name="DocumentReference" type="CT_DocumentReference">
342     </xs:element>
343
344     <xs:element name="LinkTarget" type="CT_LinkTarget">
345     </xs:element>
346
347     <xs:element name="PageContent.LinkTargets" type="CT_CP_LinkTargets">
348     </xs:element>
349
350     <xs:element name="ImageBrush.Transform" type="CT_CP_Transform">
351     </xs:element>
352
353     <xs:element name="VisualBrush.Transform" type="CT_CP_Transform">
354     </xs:element>
355
356     <xs:element name="LinearGradientBrush.Transform" type="CT_CP_Transform">
357     </xs:element>
358
359     <xs:element name="RadialGradientBrush.Transform" type="CT_CP_Transform">
360     </xs:element>
361
362     <xs:element name="PathGeometry.Transform" type="CT_CP_Transform">
363     </xs:element>
364
365     <xs:element name="Glyphs.RenderTransform" type="CT_CP_Transform">
366     </xs:element>
367
368     <xs:element name="Path.RenderTransform" type="CT_CP_Transform">
369     </xs:element>
370
371     <xs:element name="Canvas.RenderTransform" type="CT_CP_Transform">
372     </xs:element>
373
374     <xs:element name="VisualBrush.Visual" type="CT_CP_Visual">
375     </xs:element>
```

```

376 <xs:element name="LinearGradientBrush.GradientStops" type="CT_CP_GradientStops">
377     </xs:element>
378
379
380 <xs:element name="RadialGradientBrush.GradientStops" type="CT_CP_GradientStops">
381     </xs:element>
382
383 <xs:element name="Glyphs.Clip" type="CT_CP_Geometry">
384     </xs:element>
385
386 <xs:element name="Path.Clip" type="CT_CP_Geometry">
387     </xs:element>
388
389 <xs:element name="Canvas.Clip" type="CT_CP_Geometry">
390     </xs:element>
391
392 <xs:element name="Glyphs.OpacityMask" type="CT_CP_Brush">
393     </xs:element>
394
395 <xs:element name="Path.OpacityMask" type="CT_CP_Brush">
396     </xs:element>
397
398 <xs:element name="Canvas.OpacityMask" type="CT_CP_Brush">
399     </xs:element>
400
401 <xs:element name="Glyphs.Fill" type="CT_CP_Brush">
402     </xs:element>
403
404 <xs:element name="Path.Fill" type="CT_CP_Brush">
405     </xs:element>
406
407 <xs:element name="Path.Data" type="CT_CP_Geometry">
408     </xs:element>
409
410 <xs:element name="Path.Stroke" type="CT_CP_Brush">
411     </xs:element>
412
413 <xs:element name="Canvas.Resources" type="CT_CP_Resources">
414     </xs:element>
415
416 <xs:element name="FixedPage.Resources" type="CT_CP_Resources">
417     </xs:element>
418
419 <xs:attributeGroup name="AG_GradientStop">
420     <xs:attribute name="Color" type="ST_Color" use="required">
421         </xs:attribute>
422     <xs:attribute name="Offset" type="ST_Double" use="required">
423         </xs:attribute>
424 </xs:attributeGroup>
425
426 <xs:attributeGroup name="AG_Brush">
427     <xs:attribute name="Opacity" type="ST_ZeroOne" default="1.0">
428         </xs:attribute>
429     <xs:attribute ref="x:Key" />
430 </xs:attributeGroup>

```

```

431
432     <xs:attributeGroup name="AG_TileBrush">
433         <xs:attribute name="Transform" type="ST_RscRefMatrix">
434             </xs:attribute>
435         <xs:attribute name="Viewbox" type="ST_ViewBox" use="required">
436             </xs:attribute>
437         <xs:attribute name="Viewport" type="ST_ViewBox" use="required">
438             </xs:attribute>
439         <xs:attribute name="TileMode" type="ST_TileMode" default="None">
440             </xs:attribute>
441         <xs:attribute name="ViewboxUnits" type="ST_ViewUnits" use="required"
442 fixed="Absolute">
443             </xs:attribute>
444         <xs:attribute name="ViewportUnits" type="ST_ViewUnits" use="required"
445 fixed="Absolute">
446             </xs:attribute>
447     </xs:attributeGroup>
448
449     <xs:attributeGroup name="AG_VisualBrush">
450         <xs:attribute name="Visual" type="ST_RscRef">
451             </xs:attribute>
452     </xs:attributeGroup>
453
454     <xs:attributeGroup name="AG_GradientBrush">
455         <xs:attribute name="ColorInterpolationMode" type="ST_ClrIntMode"
456 default="SRgbLinearInterpolation">
457             </xs:attribute>
458         <xs:attribute name="SpreadMethod" type="ST_SpreadMethod" default="Pad">
459             </xs:attribute>
460         <xs:attribute name="MappingMode" type="ST_MappingMode" use="required"
461 fixed="Absolute">
462             </xs:attribute>
463     </xs:attributeGroup>
464
465     <xs:attributeGroup name="AG_SolidColorBrush">
466         <xs:attribute name="Color" type="ST_Color" use="required">
467             </xs:attribute>
468     </xs:attributeGroup>
469
470     <xs:attributeGroup name="AG_ImageBrush">
471         <xs:attribute name="ImageSource" type="ST_UriCtxBmp" use="required">
472             </xs:attribute>
473     </xs:attributeGroup>
474
475     <xs:attributeGroup name="AG_LinearGradientBrush">
476         <xs:attribute name="Transform" type="ST_RscRefMatrix">
477             </xs:attribute>
478         <xs:attribute name="StartPoint" type="ST_Point" use="required">
479             </xs:attribute>
480         <xs:attribute name="EndPoint" type="ST_Point" use="required">
481             </xs:attribute>
482     </xs:attributeGroup>
483
484     <xs:attributeGroup name="AG_RadialGradientBrush">
485         <xs:attribute name="Transform" type="ST_RscRefMatrix">

```

```

486         </xs:attribute>
487     <xs:attribute name="Center" type="ST_Point" use="required">
488         </xs:attribute>
489     <xs:attribute name="GradientOrigin" type="ST_Point" use="required">
490         </xs:attribute>
491     <xs:attribute name="RadiusX" type="ST_GEZero" use="required">
492         </xs:attribute>
493     <xs:attribute name="RadiusY" type="ST_GEZero" use="required">
494         </xs:attribute>
495 </xs:attributeGroup>
496
497 <xs:attributeGroup name="AG_PathGeometry">
498     <xs:attribute name="Figures" type="ST_AbbrGeom">
499         </xs:attribute>
500     <xs:attribute name="FillRule" type="ST_FillRule" default="EvenOdd">
501         </xs:attribute>
502     <xs:attribute name="Transform" type="ST_RscRefMatrix">
503         </xs:attribute>
504     <xs:attribute ref="x:Key" />
505 </xs:attributeGroup>
506
507 <xs:attributeGroup name="AG_ResourceDictionary">
508     <xs:attribute name="Source" type="xs:anyURI">
509         </xs:attribute>
510 </xs:attributeGroup>
511
512 <xs:attributeGroup name="AG_PolyLineSegment">
513     <xs:attribute name="Points" type="ST_Points" use="required">
514         </xs:attribute>
515     <xs:attribute name="IsStroked" type="ST_Boolean" default="true">
516         </xs:attribute>
517 </xs:attributeGroup>
518
519 <xs:attributeGroup name="AG_ArcSegment">
520     <xs:attribute name="Point" type="ST_Point" use="required">
521         </xs:attribute>
522     <xs:attribute name="Size" type="ST_PointGE0" use="required">
523         </xs:attribute>
524     <xs:attribute name="RotationAngle" type="ST_Double" use="required">
525         </xs:attribute>
526     <xs:attribute name="IsLargeArc" type="ST_Boolean" use="required">
527         </xs:attribute>
528     <xs:attribute name="SweepDirection" type="ST_SweepDirection" use="required">
529         </xs:attribute>
530     <xs:attribute name="IsStroked" type="ST_Boolean" default="true">
531         </xs:attribute>
532 </xs:attributeGroup>
533
534 <xs:attributeGroup name="AG_PolyBezierSegment">
535     <xs:attribute name="Points" type="ST_Points" use="required">
536         </xs:attribute>
537     <xs:attribute name="IsStroked" type="ST_Boolean" default="true">
538         </xs:attribute>
539 </xs:attributeGroup>
540

```

```

541 <xs:attributeGroup name="AG_PolyQuadraticBezierSegment">
542   <xs:attribute name="Points" type="ST_Points" use="required">
543     </xs:attribute>
544   <xs:attribute name="IsStroked" type="ST_Boolean" default="true">
545     </xs:attribute>
546 </xs:attributeGroup>
547
548 <xs:attributeGroup name="AG_Glyphs">
549   <xs:attribute name="BidiLevel" default="0">
550     <xs:simpleType>
551       <xs:restriction base="xs:integer">
552         <xs:minInclusive value="0" />
553         <xs:maxInclusive value="61" />
554       </xs:restriction>
555     </xs:simpleType>
556   </xs:attribute>
557   <xs:attribute name="CaretStops" type="ST_CaretStops">
558     </xs:attribute>
559   <xs:attribute name="DeviceFontName" type="ST_UnicodeString">
560     </xs:attribute>
561   <xs:attribute name="Fill" type="ST_RscRefColor">
562     </xs:attribute>
563   <xs:attribute name="FontRenderingEmSize" type="ST_GEZero" use="required">
564     </xs:attribute>
565   <xs:attribute name="FontUri" type="xs:anyURI" use="required">
566     </xs:attribute>
567   <xs:attribute name="OriginX" type="ST_Double" use="required">
568     </xs:attribute>
569   <xs:attribute name="OriginY" type="ST_Double" use="required">
570     </xs:attribute>
571   <xs:attribute name="IsSideways" type="ST_Boolean" default="false">
572     </xs:attribute>
573   <xs:attribute name="Indices" type="ST_Indices">
574     </xs:attribute>
575   <xs:attribute name="UnicodeString" type="ST_UnicodeString">
576     </xs:attribute>
577   <xs:attribute name="StyleSimulations" type="ST_StyleSimulations"
578 default="None">
579     </xs:attribute>
580   <xs:attribute name="RenderTransform" type="ST_RscRefMatrix">
581     </xs:attribute>
582   <xs:attribute name="Clip" type="ST_RscRefAbbrGeomF">
583     </xs:attribute>
584   <xs:attribute name="Opacity" type="ST_ZeroOne" default="1.0">
585     </xs:attribute>
586   <xs:attribute name="OpacityMask" type="ST_RscRef">
587     </xs:attribute>
588   <xs:attribute name="Name" type="ST_Name">
589     </xs:attribute>
590   <xs:attribute name="FixedPage.NavigateUri" type="xs:anyURI">
591     </xs:attribute>
592   <xs:attribute ref="xml:lang">
593     </xs:attribute>
594   <xs:attribute ref="x:Key" />
595 </xs:attributeGroup>

```

```

596
597 <xs:attributeGroup name="AG_Path">
598   <xs:attribute name="Data" type="ST_RscRefAbbrGeomF">
599     </xs:attribute>
600   <xs:attribute name="Fill" type="ST_RscRefColor">
601     </xs:attribute>
602   <xs:attribute name="RenderTransform" type="ST_RscRefMatrix">
603     </xs:attribute>
604   <xs:attribute name="Clip" type="ST_RscRefAbbrGeomF">
605     </xs:attribute>
606   <xs:attribute name="Opacity" type="ST_ZeroOne" default="1.0">
607     </xs:attribute>
608   <xs:attribute name="OpacityMask" type="ST_RscRef">
609     </xs:attribute>
610   <xs:attribute name="Stroke" type="ST_RscRefColor">
611     </xs:attribute>
612   <xs:attribute name="StrokeDashArray" type="ST_EvenArrayPos">
613     </xs:attribute>
614   <xs:attribute name="StrokeDashCap" type="ST_DashCap" default="Flat">
615     </xs:attribute>
616   <xs:attribute name="StrokeDashOffset" type="ST_Double" default="0.0">
617     </xs:attribute>
618   <xs:attribute name="StrokeEndLineCap" type="ST_LineCap" default="Flat">
619     </xs:attribute>
620   <xs:attribute name="StrokeStartLineCap" type="ST_LineCap" default="Flat">
621     </xs:attribute>
622   <xs:attribute name="StrokeLineJoin" type="ST_LineJoin" default="Miter">
623     </xs:attribute>
624   <xs:attribute name="StrokeMiterLimit" type="ST_GEOne" default="10.0">
625     </xs:attribute>
626   <xs:attribute name="StrokeThickness" type="ST_GEZero" default="1.0">
627     </xs:attribute>
628   <xs:attribute name="Name" type="ST_Name">
629     </xs:attribute>
630   <xs:attribute name="FixedPage.NavigateUri" type="xs:anyURI">
631     </xs:attribute>
632   <xs:attribute ref="xml:lang">
633     </xs:attribute>
634   <xs:attribute ref="x:Key" />
635 </xs:attributeGroup>
636
637 <xs:attributeGroup name="AG_PathFigure">
638   <xs:attribute name="IsClosed" type="ST_Boolean" default="false">
639     </xs:attribute>
640   <xs:attribute name="StartPoint" type="ST_Point" use="required">
641     </xs:attribute>
642   <xs:attribute name="IsFilled" type="ST_Boolean" default="true">
643     </xs:attribute>
644 </xs:attributeGroup>
645
646 <xs:attributeGroup name="AG_Canvas">
647   <xs:attribute name="RenderTransform" type="ST_RscRefMatrix">
648     </xs:attribute>
649   <xs:attribute name="Clip" type="ST_RscRefAbbrGeomF">
650     </xs:attribute>

```

```

651     <xs:attribute name="Opacity" type="ST_ZeroOne" default="1.0">
652         </xs:attribute>
653     <xs:attribute name="OpacityMask" type="ST_RscRef">
654         </xs:attribute>
655     <xs:attribute name="Name" type="ST_Name">
656         </xs:attribute>
657     <xs:attribute name="RenderOptions.EdgeMode" type="ST_EdgeMode">
658         </xs:attribute>
659     <xs:attribute name="FixedPage.NavigateUri" type="xs:anyURI">
660         </xs:attribute>
661     <xs:attribute ref="xml:lang">
662         </xs:attribute>
663     <xs:attribute ref="x:Key" />
664 </xs:attributeGroup>
665
666 <xs:attributeGroup name="AG_PageContent">
667     <xs:attribute name="Source" type="xs:anyURI" use="required">
668         </xs:attribute>
669     <xs:attribute name="Width" type="ST_GEOne">
670         </xs:attribute>
671     <xs:attribute name="Height" type="ST_GEOne">
672         </xs:attribute>
673 </xs:attributeGroup>
674
675 <xs:attributeGroup name="AG_LinkTarget">
676     <xs:attribute name="Name" type="ST_NUName" use="required">
677         </xs:attribute>
678 </xs:attributeGroup>
679
680 <xs:attributeGroup name="AG_DocumentReference">
681     <xs:attribute name="Source" type="xs:anyURI" use="required">
682         </xs:attribute>
683 </xs:attributeGroup>
684
685 <xs:attributeGroup name="AG_MatrixTransform">
686     <xs:attribute name="Matrix" type="ST_Matrix" use="required">
687         </xs:attribute>
688     <xs:attribute ref="x:Key" />
689 </xs:attributeGroup>
690
691 <xs:attributeGroup name="AG_FixedPage">
692     <xs:attribute name="Width" type="ST_GEOne" use="required">
693         </xs:attribute>
694     <xs:attribute name="Height" type="ST_GEOne" use="required">
695         </xs:attribute>
696     <xs:attribute name="ContentBox" type="ST_ContentBox">
697         </xs:attribute>
698     <xs:attribute name="BleedBox" type="ST_BleedBox">
699         </xs:attribute>
700     <xs:attribute ref="xml:lang" use="required">
701         </xs:attribute>
702     <xs:attribute name="Name" type="ST_Name">
703         </xs:attribute>
704 </xs:attributeGroup>
705

```

```

706 <xs:attributeGroup name="AG_AutomationProvider">
707   <xs:attribute name="AutomationProperties.Name" type="xs:string">
708     </xs:attribute>
709   <xs:attribute name="AutomationProperties.HelpText" type="xs:string">
710     </xs:attribute>
711 </xs:attributeGroup>
712
713 <xs:attributeGroup name="AG_SnapsToDevicePixels">
714   <xs:attribute name="SnapsToDevicePixels" type="ST_Boolean">
715     </xs:attribute>
716 </xs:attributeGroup>
717
718 <!-- Simple data types -->
719 <!-- A unique Name (ID with pattern restriction according to XPS spec) -->
720 <xs:simpleType name="ST_Name">
721   <xs:restriction base="xs:ID">
722     <xs:pattern
723 value="(\p{Lu}|\p{Ll}|\p{Lt}|\p{Lo}|\p{N1}|_)(\p{Lu}|\p{Ll}|\p{Lt}|\p{Lo}|\p{N1}|\p{Mn}
724 |\p{Mc}|\p{Nd}|_)*" />
725   </xs:restriction>
726 </xs:simpleType>
727
728 <!-- A non-unique Name (ID with pattern restriction according to XPS spec) -->
729 <xs:simpleType name="ST_NUName">
730   <xs:restriction base="xs:string">
731     <xs:pattern
732 value="(\p{Lu}|\p{Ll}|\p{Lt}|\p{Lo}|\p{N1}|_)(\p{Lu}|\p{Ll}|\p{Lt}|\p{Lo}|\p{N1}|\p{Mn}
733 |\p{Mc}|\p{Nd}|_)*" />
734   </xs:restriction>
735 </xs:simpleType>
736
737 <!-- Boolean with true and false only (no 0 or 1) -->
738 <xs:simpleType name="ST_Boolean">
739   <xs:restriction base="xs:boolean">
740     <xs:pattern value="true|false" />
741   </xs:restriction>
742 </xs:simpleType>
743
744 <!-- real number from 0.0 to 1.0 inclusive -->
745 <xs:simpleType name="ST_ZeroOne">
746   <xs:restriction base="ST_Double">
747     <xs:minInclusive value="0.0" />
748     <xs:maxInclusive value="1.0" />
749   </xs:restriction>
750 </xs:simpleType>
751
752 <!-- positive real number -->
753 <xs:simpleType name="ST_GEZero">
754   <xs:restriction base="ST_Double">
755     <xs:minInclusive value="0.0" />
756   </xs:restriction>
757 </xs:simpleType>
758
759 <!-- positive real number, equal or greater than one -->
760 <xs:simpleType name="ST_GEOne">

```



```

761     <xs:restriction base="ST_Double">
762         <xs:minInclusive value="1.0" />
763     </xs:restriction>
764 </xs:simpleType>
765
766 <!-- Double -->
767 <xs:simpleType name="ST_Double">
768     <xs:restriction base="xs:double">
769         <xs:whiteSpace value="collapse" />
770 <!--
771         <xs:pattern value="[rn]" />
772 -->
773         <xs:pattern value="((\-|\+)?(([\0-9]+(\.[\0-9]+)?)|(\.[\0-9]+))((e|E)(\-
774 |\+)?[\0-9]+)?)" />
775     </xs:restriction>
776 </xs:simpleType>
777
778 <!-- Point: 2 numbers, separated by , and arbitrary whitespace -->
779 <xs:simpleType name="ST_Point">
780     <xs:restriction base="xs:string">
781         <xs:whiteSpace value="collapse" />
782 <!--
783         <xs:pattern value="[rn][scs][rn]" />
784 -->
785         <xs:pattern value="((\-|\+)?(([\0-9]+(\.[\0-9]+)?)|(\.[\0-9]+))((e|E)(\-
786 |\+)?[\0-9]+)?) ( ?, ?)((\-|\+)?(([\0-9]+(\.[\0-9]+)?)|(\.[\0-9]+))((e|E)(\-|\+)?[\0-9]+)?)"
787 />
788     </xs:restriction>
789 </xs:simpleType>
790
791 <!-- PointGE0: 2 non-negative numbers, separated by , and arbitrary whitespace -->
792 <xs:simpleType name="ST_PointGE0">
793     <xs:restriction base="xs:string">
794         <xs:whiteSpace value="collapse" />
795 <!--
796         <xs:pattern value="[prn][scs][prn]" />
797 -->
798         <xs:pattern value="(\+?(([\0-9]+(\.[\0-9]+)?)|(\.[\0-9]+))((e|E)(\-|\+)?[\0-
799 9]+)?) ( ?, ?)(\+?(([\0-9]+(\.[\0-9]+)?)|(\.[\0-9]+))((e|E)(\-|\+)?[\0-9]+)?)" />
800     </xs:restriction>
801 </xs:simpleType>
802
803 <!-- Points: List of ST_Point, separated by arbitrary whitespace -->
804 <xs:simpleType name="ST_Points">
805     <xs:restriction base="xs:string">
806         <xs:whiteSpace value="collapse" />
807 <!--
808         <xs:pattern value="[rn][scs][rn]( [rn][scs][rn])*" />
809 -->
810         <xs:pattern value="((\-|\+)?(([\0-9]+(\.[\0-9]+)?)|(\.[\0-9]+))((e|E)(\-
811 |\+)?[\0-9]+)?) ( ?, ?)((\-|\+)?(([\0-9]+(\.[\0-9]+)?)|(\.[\0-9]+))((e|E)(\-|\+)?[\0-9]+)?) (
812 ((\-|\+)?(([\0-9]+(\.[\0-9]+)?)|(\.[\0-9]+))((e|E)(\-|\+)?[\0-9]+)?) ( ?, ?)((\-|\+)?(([\0-
813 9]+(\.[\0-9]+)?)|(\.[\0-9]+))((e|E)(\-|\+)?[\0-9]+)?)*)" />
814     </xs:restriction>
815 </xs:simpleType>

```

```

816
817     <!-- EvenArray: List with even number of entries of non-negative numbers. -->
818     <xs:simpleType name="ST_EvenArrayPos">
819         <xs:restriction base="xs:string">
820             <xs:whiteSpace value="collapse" />
821     <!--
822         <xs:pattern value="[prn] [prn]( [prn] [prn])*"/>
823     -->
824         <xs:pattern value="(\+?(([0-9]+(\.[0-9]+)?)|(\.[0-9]+))((e|E)(\-|\+)?[0-
825 9]+)?)(\+?(([0-9]+(\.[0-9]+)?)|(\.[0-9]+))((e|E)(\-|\+)?[0-9]+)?)(\+?(([0-9]+(\.[0-
826 9]+)?)|(\.[0-9]+))((e|E)(\-|\+)?[0-9]+)?)(\+?(([0-9]+(\.[0-9]+)?)|(\.[0-9]+))((e|E)(\-
827 |\+)?[0-9]+)?))*" />
828         </xs:restriction>
829     </xs:simpleType>
830
831     <!-- Array: List of numbers. -->
832     <xs:simpleType name="ST_Array">
833         <xs:restriction base="xs:string">
834             <xs:whiteSpace value="collapse" />
835     <!--
836         <xs:pattern value="([rn] ?)*"/>
837     -->
838         <xs:pattern value="(((\-|\+)?(([0-9]+(\.[0-9]+)?)|(\.[0-9]+))((e|E)(\-
839 |\+)?[0-9]+)?) ?)*" />
840         </xs:restriction>
841     </xs:simpleType>
842
843     <!-- ViewBox: 4 numbers, separated by , and arbitrary whitespace. Second number
844 pair must be non-negative -->
845     <xs:simpleType name="ST_ViewBox">
846         <xs:restriction base="xs:string">
847             <xs:whiteSpace value="collapse" />
848     <!--
849         <xs:pattern value="[rn][scs][rn][scs][prn][scs][prn]"/>
850     -->
851         <xs:pattern value="((\-|\+)?(([0-9]+(\.[0-9]+)?)|(\.[0-9]+))((e|E)(\-
852 |\+)?[0-9]+)?)( ?, ?)((\-|\+)?(([0-9]+(\.[0-9]+)?)|(\.[0-9]+))((e|E)(\-|\+)?[0-9]+)?)(
853 ?, ?)(\+?(([0-9]+(\.[0-9]+)?)|(\.[0-9]+))((e|E)(\-|\+)?[0-9]+)?)( ?, ?)(\+?(([0-
854 9]+(\.[0-9]+)?)|(\.[0-9]+))((e|E)(\-|\+)?[0-9]+)?)" />
855         </xs:restriction>
856     </xs:simpleType>
857
858     <!-- ContentBox: 4 non-negative numbers, separated by commas and arbitrary
859 whitespace -->
860     <xs:simpleType name="ST_ContentBox">
861         <xs:restriction base="xs:string">
862             <xs:whiteSpace value="collapse" />
863     <!--
864         <xs:pattern value="[prn][scs][prn][scs][prn][scs][prn]"/>
865     -->
866         <xs:pattern value="(\+?(([0-9]+(\.[0-9]+)?)|(\.[0-9]+))((e|E)(\-|\+)?[0-
867 9]+)?)( ?, ?)(\+?(([0-9]+(\.[0-9]+)?)|(\.[0-9]+))((e|E)(\-|\+)?[0-9]+)?)( ?,
868 ?)(\+?(([0-9]+(\.[0-9]+)?)|(\.[0-9]+))((e|E)(\-|\+)?[0-9]+)?)( ?, ?)(\+?(([0-9]+(\.[0-
869 9]+)?)|(\.[0-9]+))((e|E)(\-|\+)?[0-9]+)?)" />
870         </xs:restriction>

```

```

871     </xs:simpleType>
872
873     <!-- BleedBox: 4 numbers, separated by , and arbitrary whitespace. Second number
874 pair must be non-negative -->
875     <xs:simpleType name="ST_BleedBox">
876     <xs:restriction base="xs:string">
877         <xs:whiteSpace value="collapse" />
878 <!--
879         <xs:pattern value="[rn][scs][rn][scs][prn][scs][prn]" />
880 -->
881         <xs:pattern value="((\-|\+)?(([\0-9]+(\.[\0-9]+)?)|(\.[\0-9]+))((e|E)(\-
882 |\+)?[0-9]+)?)(?, ?)((\-|\+)?(([\0-9]+(\.[\0-9]+)?)|(\.[\0-9]+))((e|E)(\-|\+)?[0-9]+)?)(
883 ?, ?)(\+?(([\0-9]+(\.[\0-9]+)?)|(\.[\0-9]+))((e|E)(\-|\+)?[0-9]+)?)(?, ?)(\+?(([\0-
884 9]+(\.[\0-9]+)?)|(\.[\0-9]+))((e|E)(\-|\+)?[0-9]+)?)" />
885     </xs:restriction>
886 </xs:simpleType>
887
888     <!-- Bare Matrix form: 6 numbers separated by , and arbitrary whitespace -->
889     <xs:simpleType name="ST_Matrix">
890     <xs:restriction base="xs:string">
891         <xs:whiteSpace value="collapse" />
892 <!--
893         <xs:pattern value="[rn][scs][rn][scs][rn][scs][rn][scs][rn][scs][rn]" />
894 -->
895         <xs:pattern value="((\-|\+)?(([\0-9]+(\.[\0-9]+)?)|(\.[\0-9]+))((e|E)(\-
896 |\+)?[0-9]+)?)(?, ?)((\-|\+)?(([\0-9]+(\.[\0-9]+)?)|(\.[\0-9]+))((e|E)(\-|\+)?[0-9]+)?)(
897 ?, ?)((\-|\+)?(([\0-9]+(\.[\0-9]+)?)|(\.[\0-9]+))((e|E)(\-|\+)?[0-9]+)?)(?, ?)((\-
898 |\+)?(([\0-9]+(\.[\0-9]+)?)|(\.[\0-9]+))((e|E)(\-|\+)?[0-9]+)?)(?, ?)((\-|\+)?(([\0-
899 9]+(\.[\0-9]+)?)|(\.[\0-9]+))((e|E)(\-|\+)?[0-9]+)?)(?, ?)((\-|\+)?(([\0-9]+(\.[\0-
900 9]+)?)|(\.[\0-9]+))((e|E)(\-|\+)?[0-9]+)?)" />
901     </xs:restriction>
902 </xs:simpleType>
903
904     <!-- Color: 6 or 8 hex digits -->
905     <xs:simpleType name="ST_Color">
906     <xs:restriction base="xs:string">
907         <!-- The pattern restriction does not check for sRGB gamut -->
908         <!-- The pattern restriction does not check for color profile URI validity
909 -->
910         <xs:whiteSpace value="collapse" />
911 <!--
912         <xs:pattern value="(#[\0-9a-fA-F]{2})?[\0-9a-fA-F]{6}|\
913             (sc# ?[dec][scs][dec][scs][dec]([scs][dec])?)|\
914             (ContextColor +[\S]+ ?[dec]([scs][dec]){3,8})" />
915 -->
916         <xs:pattern value="(#[\0-9a-fA-F]{2})?[\0-9a-fA-F]{6}|(sc# ?(\-?(([\0-
917 9]+(\.[\0-9]+)?)|(\.[\0-9]+))) ( ?, ?)(\-?(([\0-9]+(\.[\0-9]+)?)|(\.[\0-9]+))) ( ?, ?)(\-
918 ?(([\0-9]+(\.[\0-9]+)?)|(\.[\0-9]+))) ( ?, ?)(\-?(([\0-9]+(\.[\0-9]+)?)|(\.[\0-
919 9]+)))?)|(ContextColor +[\S]+ ?(\-?(([\0-9]+(\.[\0-9]+)?)|(\.[\0-9]+))) ( ?, ?)(\-?(([\0-
920 9]+(\.[\0-9]+)?)|(\.[\0-9]+)))}{3,8})" />
921     </xs:restriction>
922 </xs:simpleType>
923
924     <!-- Indices grammar for Glyphs.CaretStops -->
925     <xs:simpleType name="ST_CaretStops">

```

```

926     <xs:restriction base="xs:string">
927         <xs:whiteSpace value="collapse" />
928         <xs:pattern value="[0-9A-Fa-f]*" />
929     </xs:restriction>
930 </xs:simpleType>
931
932 <!-- Indices grammar for Glyphs.Indices -->
933 <xs:simpleType name="ST_Indices">
934     <xs:restriction base="xs:string">
935         <xs:whiteSpace value="collapse" />
936 <!--
937         <xs:pattern value="(\
938             ((\[pint](:[pint])?\))?[uint])?\
939             (,[prn]?([rn]?([rn])?)?)?\
940             )\
941             (;\
942             ((\[pint](:[pint])?\))?[uint])?\
943             (,[prn]?([rn]?([rn])?)?)?\
944             )" />
945 -->
946         <xs:pattern value="(((\((([1-9][0-9]*)(:([1-9][0-9]*)?)?\))?)?([0-
947 9]+))?(,(\+?((([0-9]+(\.[0-9]+)?)|(\.[0-9]+))((e|E)(\-|\+)?[0-9]+)?)?,( (\-|\+)?((([0-
948 9]+(\.[0-9]+)?)|(\.[0-9]+))((e|E)(\-|\+)?[0-9]+)?)?,( (\-|\+)?((([0-9]+(\.[0-
949 9]+)?)|(\.[0-9]+))((e|E)(\-|\+)?[0-9]+)?))?)?);(((\((([1-9][0-9]*)(:([1-9][0-
950 9]*)?)?\))?)?([0-9]+))?(,(\+?((([0-9]+(\.[0-9]+)?)|(\.[0-9]+))((e|E)(\-|\+)?[0-
951 9]+)?)?,( (\-|\+)?((([0-9]+(\.[0-9]+)?)|(\.[0-9]+))((e|E)(\-|\+)?[0-9]+)?)?,( (\-
952 |\+)?((([0-9]+(\.[0-9]+)?)|(\.[0-9]+))((e|E)(\-|\+)?[0-9]+)?))?)?)?)" />
953     </xs:restriction>
954 </xs:simpleType>
955
956 <!-- UnicodeString grammar -->
957 <xs:simpleType name="ST_UnicodeString">
958     <xs:restriction base="xs:string">
959         <xs:pattern value="(([\^{\}|\{\}\})).*" />
960     </xs:restriction>
961 </xs:simpleType>
962
963 <!-- Abbreviated Geometry grammar for Path.Data , clip and Geometries -->
964 <xs:simpleType name="ST_AbbrGeomF">
965     <xs:restriction base="xs:string">
966         <xs:whiteSpace value="collapse" />
967 <!--
968         <xs:pattern value="(F ?(0|1))?\
969             ( ?(M|m)( ?[rn][scs][rn]))\
970             (\
971                 ( ?(M|m)( ?[rn][scs][rn]))|\
972                 ( ?(L|l)( ?[rn][scs][rn])( [rn][scs][rn])*)|\
973                 ( ?(H|h|V|v)( ?[rn])( [rn])*)|\
974                 ( ?(Q|q|S|s)( ?[rn][scs][rn] [rn][scs][rn]))((
975 [rn][scs][rn]){2})*|\
976                 ( ?(C|c)( ?[rn][scs][rn]( [rn][scs][rn]){2}))((
977 [rn][scs][rn]){3})*|\
978                 ( ?(A|a)( ?[rn][scs][rn] [rn] [0-1] [0-1]
979 [rn][scs][rn]))\

```

```

980 ( [rn][scs][rn] [rn] [0-1] [0-1]
981 [rn][scs][rn]))*\
982 ( ?(Z|z))\
983 )"*/>
984 -->
985 <xs:pattern value="(F ?(0|1))?( ?(M|m)( ?((\-|\+)?((\.[0-9]+\.[0-
986 9]+\.?)|(\.[0-9]+))((e|E)(\-|\+)?[0-9]+)?)( ?, ?)((\-|\+)?((\.[0-9]+\.[0-9]+\.)|(\.[0-
987 9]+\.)((e|E)(\-|\+)?[0-9]+)?)))( ?(M|m)( ?((\-|\+)?((\.[0-9]+\.[0-9]+\.)|(\.[0-
988 9]+\.)((e|E)(\-|\+)?[0-9]+)?)( ?, ?)((\-|\+)?((\.[0-9]+\.[0-9]+\.)|(\.[0-9]+\.)((e|E)(\-|\+)?[0-
989 9]+\.)?)))( ?(L|l)( ?((\-|\+)?((\.[0-9]+\.[0-9]+\.)|(\.[0-9]+\.)((e|E)(\-|\+)?[0-
990 9]+\.)?)( ?, ?)((\-|\+)?((\.[0-9]+\.[0-9]+\.)|(\.[0-9]+\.)((e|E)(\-|\+)?[0-9]+)?)))( ((\-|\+)?((\.[0-9]+\.[0-9]+\.)|(\.[0-9]+\.)((e|E)(\-|\+)?[0-9]+)?)( ?, ?)((\-|\+)?((\.[0-9]+\.[0-9]+\.)|(\.[0-9]+\.)((e|E)(\-|\+)?[0-9]+)?)))*|( ?(H|h|V|v)( ?((\-|\+)?((\.[0-9]+\.[0-9]+\.)|(\.[0-9]+\.)((e|E)(\-|\+)?[0-9]+)?)))( ((\-|\+)?((\.[0-9]+\.[0-9]+\.)|(\.[0-9]+\.)((e|E)(\-|\+)?[0-9]+)?)))*|( ?(Q|q|S|s)( ?((\-|\+)?((\.[0-9]+\.[0-9]+\.)|(\.[0-9]+\.)((e|E)(\-|\+)?[0-9]+)?)( ?, ?)((\-|\+)?((\.[0-9]+\.[0-9]+\.)|(\.[0-9]+\.)((e|E)(\-|\+)?[0-9]+)?))((\-|\+)?((\.[0-9]+\.[0-9]+\.)|(\.[0-9]+\.)((e|E)(\-|\+)?[0-9]+)?))((\-|\+)?((\.[0-9]+\.[0-9]+\.)|(\.[0-9]+\.)((e|E)(\-|\+)?[0-9]+)?)))*|( ?(C|c)( ?((\-|\+)?((\.[0-9]+\.[0-9]+\.)|(\.[0-9]+\.)((e|E)(\-|\+)?[0-9]+)?))((\-|\+)?((\.[0-9]+\.[0-9]+\.)|(\.[0-9]+\.)((e|E)(\-|\+)?[0-9]+)?))((\-|\+)?((\.[0-9]+\.[0-9]+\.)|(\.[0-9]+\.)((e|E)(\-|\+)?[0-9]+)?)))*|( ?(A|a)( ?((\-|\+)?((\.[0-9]+\.[0-9]+\.)|(\.[0-9]+\.)((e|E)(\-|\+)?[0-9]+)?))((\-|\+)?((\.[0-9]+\.[0-9]+\.)|(\.[0-9]+\.)((e|E)(\-|\+)?[0-9]+)?))((\-|\+)?((\.[0-9]+\.[0-9]+\.)|(\.[0-9]+\.)((e|E)(\-|\+)?[0-9]+)?)))*|( ?(Z|z)))*" />
991 992 993 994 995 996 997 998 999
1000 </xs:restriction>
1001 </xs:simpleType>
1002
1003 <!-- Abbreviated Geometry grammar for PatGeometry.Figures -->
1004 <xs:simpleType name="ST_AbbrGeom">
1005 <xs:restriction base="xs:string">
1006 <xs:whiteSpace value="collapse" />
1007 <!--
1008 <xs:pattern value="( ?(M|m)( ?[rn][scs][rn]))\
1009 (\
1010 ( ?(M|m)( ?[rn][scs][rn]))*\
1011 ( ?(L|l)( ?[rn][scs][rn])( [rn][scs][rn]))*\
1012 ( ?(H|h|V|v)( ?[rn])( [rn]))*\
1013 ( ?(Q|q|S|s)( ?[rn][scs][rn] [rn][scs][rn]))((
1014 [rn][scs][rn]){2})*|\
1015 ( ?(C|c)( ?[rn][scs][rn]( [rn][scs][rn]){2}))((
1016 [rn][scs][rn]){3})*|\
1017 ( ?(A|a)( ?[rn][scs][rn] [rn] [0-1] [0-1]
1018 [rn][scs][rn]))\
1019 ( [rn][scs][rn] [rn] [0-1] [0-1]
1020 [rn][scs][rn]))*\

```

```

1035         ( ?(Z|z))\
1036         )*/>
1037     -->
1038         <xs:pattern value="( ?(M|m)( ?((\-|\+)?(([\0-9]+(\.[\0-9]+)?)|(\.[\0-
1039     9+))((e|E)(\-|\+)?[\0-9]+)?)( ?, ?)((\-|\+)?(([\0-9]+(\.[\0-9]+)?)|(\.[\0-9]+))((e|E)(\-
1040     |\+)?[\0-9]+?)))(( ?(M|m)( ?((\-|\+)?(([\0-9]+(\.[\0-9]+)?)|(\.[\0-9]+))((e|E)(\-|\+)?[\0-
1041     9+)?))( ?, ?)((\-|\+)?(([\0-9]+(\.[\0-9]+)?)|(\.[\0-9]+))((e|E)(\-|\+)?[\0-9]+?)))|
1042     ?(L|l)( ?((\-|\+)?(([\0-9]+(\.[\0-9]+)?)|(\.[\0-9]+))((e|E)(\-|\+)?[\0-9]+)?)( ?, ?)((\-
1043     |\+)?(([\0-9]+(\.[\0-9]+)?)|(\.[\0-9]+))((e|E)(\-|\+)?[\0-9]+?)))( (\-|\+)?(([\0-9]+(\.[\0-
1044     9+)?)|(\.[\0-9]+))((e|E)(\-|\+)?[\0-9]+?))( ?, ?)((\-|\+)?(([\0-9]+(\.[\0-9]+)?)|(\.[\0-
1045     9+))((e|E)(\-|\+)?[\0-9]+?)))*|( ?(H|h|V|v)( ?((\-|\+)?(([\0-9]+(\.[\0-9]+)?)|(\.[\0-
1046     9+))((e|E)(\-|\+)?[\0-9]+?)))( (\-|\+)?(([\0-9]+(\.[\0-9]+)?)|(\.[\0-9]+))((e|E)(\-
1047     |\+)?[\0-9]+?)))*|( ?(Q|q|S|s)( ?((\-|\+)?(([\0-9]+(\.[\0-9]+)?)|(\.[\0-9]+))((e|E)(\-
1048     |\+)?[\0-9]+?))( ?, ?)((\-|\+)?(([\0-9]+(\.[\0-9]+)?)|(\.[\0-9]+))((e|E)(\-|\+)?[\0-9]+?))
1049     ((\-|\+)?(([\0-9]+(\.[\0-9]+)?)|(\.[\0-9]+))((e|E)(\-|\+)?[\0-9]+?))( ?, ?)((\-|\+)?(([\0-
1050     9+)(\.[\0-9]+)?)|(\.[\0-9]+))((e|E)(\-|\+)?[\0-9]+?)))( (\-|\+)?(([\0-9]+(\.[\0-
1051     9+)?)|(\.[\0-9]+))((e|E)(\-|\+)?[\0-9]+?))( ?, ?)((\-|\+)?(([\0-9]+(\.[\0-9]+)?)|(\.[\0-
1052     9+))((e|E)(\-|\+)?[\0-9]+?))){2})*|( ?(C|c)( ?((\-|\+)?(([\0-9]+(\.[\0-9]+)?)|(\.[\0-
1053     9+))((e|E)(\-|\+)?[\0-9]+?))( ?, ?)((\-|\+)?(([\0-9]+(\.[\0-9]+)?)|(\.[\0-9]+))((e|E)(\-
1054     |\+)?[\0-9]+?))( (\-|\+)?(([\0-9]+(\.[\0-9]+)?)|(\.[\0-9]+))((e|E)(\-|\+)?[\0-9]+?))( ?,
1055     ?)((\-|\+)?(([\0-9]+(\.[\0-9]+)?)|(\.[\0-9]+))((e|E)(\-|\+)?[\0-9]+?))){2}(( (\-
1056     |\+)?(([\0-9]+(\.[\0-9]+)?)|(\.[\0-9]+))((e|E)(\-|\+)?[\0-9]+?))( ?, ?)((\-|\+)?(([\0-
1057     9+)(\.[\0-9]+)?)|(\.[\0-9]+))((e|E)(\-|\+)?[\0-9]+?))){3})*|( ?(A|a)( ?((\-|\+)?(([\0-
1058     9+)(\.[\0-9]+)?)|(\.[\0-9]+))((e|E)(\-|\+)?[\0-9]+?))( ?, ?)((\-|\+)?(([\0-9]+(\.[\0-
1059     9+)?)|(\.[\0-9]+))((e|E)(\-|\+)?[\0-9]+?)) (\-|\+)?(([\0-9]+(\.[\0-9]+)?)|(\.[\0-
1060     9+))((e|E)(\-|\+)?[\0-9]+?)) [0-1] [0-1] ((\-|\+)?(([\0-9]+(\.[\0-9]+)?)|(\.[\0-
1061     9+))((e|E)(\-|\+)?[\0-9]+?))( ?, ?)((\-|\+)?(([\0-9]+(\.[\0-9]+)?)|(\.[\0-9]+))((e|E)(\-
1062     |\+)?[\0-9]+?)))( (\-|\+)?(([\0-9]+(\.[\0-9]+)?)|(\.[\0-9]+))((e|E)(\-|\+)?[\0-9]+?))( ?,
1063     ?)((\-|\+)?(([\0-9]+(\.[\0-9]+)?)|(\.[\0-9]+))((e|E)(\-|\+)?[\0-9]+?)) (\-|\+)?(([\0-
1064     9+)(\.[\0-9]+)?)|(\.[\0-9]+))((e|E)(\-|\+)?[\0-9]+?)) [0-1] [0-1] ((\-|\+)?(([\0-9]+(\.[\0-
1065     9+)?)|(\.[\0-9]+))((e|E)(\-|\+)?[\0-9]+?))( ?, ?)((\-|\+)?(([\0-9]+(\.[\0-9]+)?)|(\.[\0-
1066     9+))((e|E)(\-|\+)?[\0-9]+?)))*|( ?(Z|z)))" />
1067     </xs:restriction>
1068 </xs:simpleType>
1069
1070     <!-- Image reference via Uri -->
1071     <xs:simpleType name="ST_UriImage">
1072         <xs:restriction base="xs:anyURI">
1073             <xs:pattern value="([\^{}.*]?" />
1074         </xs:restriction>
1075     </xs:simpleType>
1076
1077     <!-- Image reference via ColorConvertedBitmap -->
1078     <xs:simpleType name="ST_CtxBmpImage">
1079         <xs:restriction base="xs:string">
1080             <xs:pattern value="\{ColorConvertedBitmap[\s]+[\S]+[\s]+[\S]+\}[\s]*" />
1081         </xs:restriction>
1082     </xs:simpleType>
1083
1084     <!-- Image reference via Uri or ColorConvertedBitmap -->
1085     <xs:simpleType name="ST_UriCtxBmp">
1086         <xs:union memberTypes="ST_UriImage ST_CtxBmpImage" />
1087     </xs:simpleType>
1088
1089     <!-- Resource reference -->

```

```

1090 <xs:simpleType name="ST_RscRef">
1091   <xs:restriction base="xs:string">
1092     <xs:pattern value="\{StaticResource[\s]+[\S]+\}[\s]*" />
1093   </xs:restriction>
1094 </xs:simpleType>
1095
1096 <!-- Resource reference OR Color -->
1097 <xs:simpleType name="ST_RscRefColor">
1098   <xs:union memberTypes="ST_Color ST_RscRef" />
1099 </xs:simpleType>
1100
1101 <!-- Resource reference OR Compact Matrix-->
1102 <xs:simpleType name="ST_RscRefMatrix">
1103   <xs:union memberTypes="ST_Matrix ST_RscRef" />
1104 </xs:simpleType>
1105
1106 <!-- Resource reference OR AbbrGeomF-->
1107 <xs:simpleType name="ST_RscRefAbbrGeomF">
1108   <xs:union memberTypes="ST_AbbrGeomF ST_RscRef" />
1109 </xs:simpleType>
1110
1111 <!-- Sweep Direction enumeration -->
1112 <xs:simpleType name="ST_SweepDirection">
1113   <xs:restriction base="xs:string">
1114     <xs:enumeration value="Clockwise" />
1115     <xs:enumeration value="Counterclockwise" />
1116   </xs:restriction>
1117 </xs:simpleType>
1118
1119 <!-- Dash Cap enumeration -->
1120 <xs:simpleType name="ST_DashCap">
1121   <xs:restriction base="xs:string">
1122     <xs:enumeration value="Flat" />
1123     <xs:enumeration value="Round" />
1124     <xs:enumeration value="Square" />
1125     <xs:enumeration value="Triangle" />
1126   </xs:restriction>
1127 </xs:simpleType>
1128
1129 <!-- Line Cap enumeration -->
1130 <xs:simpleType name="ST_LineCap">
1131   <xs:restriction base="xs:string">
1132     <xs:enumeration value="Flat" />
1133     <xs:enumeration value="Round" />
1134     <xs:enumeration value="Square" />
1135     <xs:enumeration value="Triangle" />
1136   </xs:restriction>
1137 </xs:simpleType>
1138
1139 <!-- Line Join enumeration -->
1140 <xs:simpleType name="ST_LineJoin">
1141   <xs:restriction base="xs:string">
1142     <xs:enumeration value="Miter" />
1143     <xs:enumeration value="Bevel" />
1144     <xs:enumeration value="Round" />

```

```

1145     </xs:restriction>
1146 </xs:simpleType>
1147
1148 <!-- Tile Mode enumeration -->
1149 <xs:simpleType name="ST_TileMode">
1150   <xs:restriction base="xs:string">
1151     <xs:enumeration value="None" />
1152     <xs:enumeration value="Tile" />
1153     <xs:enumeration value="FlipX" />
1154     <xs:enumeration value="FlipY" />
1155     <xs:enumeration value="FlipXY" />
1156   </xs:restriction>
1157 </xs:simpleType>
1158
1159 <!-- Color Interpolation Mode enumeration -->
1160 <xs:simpleType name="ST_ClrIntMode">
1161   <xs:restriction base="xs:string">
1162     <xs:enumeration value="ScRgbLinearInterpolation" />
1163     <xs:enumeration value="SRgbLinearInterpolation" />
1164   </xs:restriction>
1165 </xs:simpleType>
1166
1167 <!-- SpreadMethod Mode enumeration -->
1168 <xs:simpleType name="ST_SpreadMethod">
1169   <xs:restriction base="xs:string">
1170     <xs:enumeration value="Pad" />
1171     <xs:enumeration value="Reflect" />
1172     <xs:enumeration value="Repeat" />
1173   </xs:restriction>
1174 </xs:simpleType>
1175
1176 <!-- FillRule Mode enumeration -->
1177 <xs:simpleType name="ST_FillRule">
1178   <xs:restriction base="xs:string">
1179     <xs:enumeration value="EvenOdd" />
1180     <xs:enumeration value="NonZero" />
1181   </xs:restriction>
1182 </xs:simpleType>
1183
1184 <!-- Edge Mode enumeration -->
1185 <xs:simpleType name="ST_EdgeMode">
1186   <xs:restriction base="xs:string">
1187     <xs:enumeration value="Aliased" />
1188   </xs:restriction>
1189 </xs:simpleType>
1190
1191 <!-- Style Simulation Enumeration -->
1192 <xs:simpleType name="ST_StyleSimulations">
1193   <xs:restriction base="xs:string">
1194     <xs:enumeration value="None" />
1195     <xs:enumeration value="ItalicSimulation" />
1196     <xs:enumeration value="BoldSimulation" />
1197     <xs:enumeration value="BoldItalicSimulation" />
1198   </xs:restriction>
1199 </xs:simpleType>

```



```
1200
1201 <!-- ViewUnits Enumeration -->
1202 <xs:simpleType name="ST_ViewUnits">
1203   <xs:restriction base="xs:string">
1204     <xs:enumeration value="Absolute" />
1205   </xs:restriction>
1206 </xs:simpleType>
1207
1208 <!-- MappingMode Enumeration -->
1209 <xs:simpleType name="ST_MappingMode">
1210   <xs:restriction base="xs:string">
1211     <xs:enumeration value="Absolute" />
1212   </xs:restriction>
1213 </xs:simpleType>
1214 </xs:schema>
```


1 **A.3 Resource Dictionary Key-~~W3C~~Schema**

2 The schema shown below is also provided in electronic form as a file named
3 ResourceDictionaryKey.xsd, which is contained in an accompanying zip archive named "XPS
4 WC3 Schemas.zip". If discrepancies exist between the representation as published below and
5 the corresponding electronic version, the published version below is the definitive version.

```
6  
1 <?xml version="1.0" encoding="utf-8"?>  
2 <xs:schema  
3 targetNamespace="http://schemas.microsoft.com/xps/2005/06/resourcedictionary-key"  
4 xmlns="http://schemas.microsoft.com/xps/2005/06/resourcedictionary-key"  
5 xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"  
6 blockDefault="#all">  
7  
8   <xs:attribute name="Key">  
9     <xs:simpleType>  
10       <xs:restriction base="xs:string">  
11         <!-- A Key (pattern restriction according to XPS spec) -->  
12         <xs:pattern  
13 value="(\p{Lu}|\p{Ll}|\p{Lt}|\p{Lo}|\p{Nl}|_)(\p{Lu}|\p{Ll}|\p{Lt}|\p{Lo}|\p{Nl}|\p{Mn}  
14 |\p{Mc}|\p{Nd}|_)*" />  
15       </xs:restriction>  
16     </xs:simpleType>  
17   </xs:attribute>  
18  
19 </xs:schema>
```


1 **A.4 Document Structure** ~~W3C Schema~~

2 The schema shown below is also provided in electronic form as a file named
 3 DocumentStructure.xsd, which is contained in an accompanying zip archive named "XPS WC3
 4 Schemas.zip". If discrepancies exist between the representation as published below and the
 5 corresponding electronic version, the published version below is the definitive version.

```

6
7 <?xml version="1.0" encoding="UTF-8"?><xs:schema
8 targetNamespace="http://schemas.microsoft.com/xps/2005/06/documentstructure"
9 xmlns="http://schemas.microsoft.com/xps/2005/06/documentstructure"
10 xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
11 blockDefault="#all">
12
13     <xs:import namespace="http://www.w3.org/XML/1998/namespace" />
14
15     <!-- =====DocumentStructure Part===== -->
16     <!-- Complex Types -->
17     <xs:complexType name="CT_DocumentStructure">
18         <xs:sequence>
19             <xs:element ref="DocumentStructure.Outline" minOccurs="0" />
20             <xs:element ref="Story" minOccurs="0" maxOccurs="unbounded" />
21         </xs:sequence>
22     </xs:complexType>
23     <xs:complexType name="CT_CP_Outline">
24         <xs:sequence>
25             <xs:element ref="DocumentOutline" />
26         </xs:sequence>
27     </xs:complexType>
28     <xs:complexType name="CT_DocumentOutline">
29         <xs:sequence>
30             <xs:element ref="OutlineEntry" maxOccurs="unbounded" />
31         </xs:sequence>
32         <xs:attributeGroup ref="AG_DocumentOutline" />
33     </xs:complexType>
34     <xs:complexType name="CT_OutlineEntry">
35         <xs:attributeGroup ref="AG_OutlineEntry" />
36     </xs:complexType>
37     <xs:complexType name="CT_Story">
38         <xs:sequence>
39             <xs:element ref="StoryFragmentReference" maxOccurs="unbounded" />
40         </xs:sequence>
41         <xs:attributeGroup ref="AG_Story" />
42     </xs:complexType>
43     <xs:complexType name="CT_StoryFragmentReference">
44         <xs:attributeGroup ref="AG_StoryFragmentReference" />
45     </xs:complexType>
46     <!-- Simple Types -->
47     <!-- A Name (ID with pattern restriction according to XPS spec) -->
48     <xs:simpleType name="ST_Name">
49         <xs:restriction base="xs:string">

```

```

44         <xs:pattern
45 value="(\p{Lu}|\p{Ll}|\p{Lo}|\p{Lt}|\p{Nl})(\p{Lu}|\p{Ll}|\p{Lo}|\p{Lt}|\p{Nl}|\p{Mn}|\
46 p{Mc}|\p{Nd}|\p{Lm}|_)*" />
47         </xs:restriction>
48     </xs:simpleType>
49     <!-- A Unique Name (ID with pattern restriction according to XPS spec) -->
50     <xs:simpleType name="ST_NameUnique">
51         <xs:restriction base="xs:ID">
52             <xs:pattern
53 value="(\p{Lu}|\p{Ll}|\p{Lo}|\p{Lt}|\p{Nl})(\p{Lu}|\p{Ll}|\p{Lo}|\p{Lt}|\p{Nl}|\p{Mn}|\
54 p{Mc}|\p{Nd}|\p{Lm}|_)*" />
55             </xs:restriction>
56         </xs:simpleType>
57     <!-- integer greater than or equal to 1 inclusive -->
58     <xs:simpleType name="ST_IntGEOne">
59         <xs:restriction base="xs:int">
60             <xs:minInclusive value="1" />
61         </xs:restriction>
62     </xs:simpleType>
63     <!-- Elements -->
64     <xs:element name="DocumentStructure" type="CT_DocumentStructure">
65         </xs:element>
66     <xs:element name="DocumentStructure.Outline" type="CT_CP_Outline">
67         </xs:element>
68     <xs:element name="DocumentOutline" type="CT_DocumentOutline">
69         </xs:element>
70     <xs:element name="OutlineEntry" type="CT_OutlineEntry">
71         </xs:element>
72     <xs:element name="Story" type="CT_Story">
73         </xs:element>
74     <xs:element name="StoryFragmentReference" type="CT_StoryFragmentReference">
75         </xs:element>
76     <!-- Attribute Groups -->
77     <xs:attributeGroup name="AG_DocumentOutline">
78         <xs:attribute ref="xml:lang" use="required">
79             </xs:attribute>
80     </xs:attributeGroup>
81     <xs:attributeGroup name="AG_OutlineEntry">
82         <xs:attribute name="OutlineLevel" type="ST_IntGEOne" use="optional"
83 default="1">
84             </xs:attribute>
85         <xs:attribute name="OutlineTarget" type="xs:anyURI" use="required">
86             </xs:attribute>
87         <xs:attribute name="Description" type="xs:string" use="required">
88             </xs:attribute>
89         <xs:attribute ref="xml:lang" use="optional">
90             </xs:attribute>
91     </xs:attributeGroup>
92     <xs:attributeGroup name="AG_Story">
93         <xs:attribute name="StoryName" type="xs:string" use="required">
94             </xs:attribute>
95     </xs:attributeGroup>
96     <xs:attributeGroup name="AG_StoryFragmentReference">
97         <xs:attribute name="FragmentName" type="xs:string" use="optional">
98             </xs:attribute>

```

```

99         <xs:attribute name="Page" type="ST_IntGEOne" use="required">
100             </xs:attribute>
101     </xs:attributeGroup>
102     <!-- =====StoryFragments Part===== -->
103     <!-- Complex Types -->
104     <xs:complexType name="CT_StoryFragments">
105         <xs:sequence>
106             <xs:element ref="StoryFragment" maxOccurs="unbounded" />
107         </xs:sequence>
108     </xs:complexType>
109     <xs:complexType name="CT_StoryFragment">
110         <xs:sequence>
111             <xs:element ref="StoryBreak" minOccurs="0" />
112             <xs:choice maxOccurs="unbounded">
113                 <xs:element ref="SectionStructure" />
114                 <xs:element ref="ParagraphStructure" />
115                 <xs:element ref="ListStructure" />
116                 <xs:element ref="TableStructure" />
117                 <xs:element ref="FigureStructure" />
118             </xs:choice>
119             <xs:element ref="StoryBreak" minOccurs="0" />
120         </xs:sequence>
121         <xs:attributeGroup ref="AG_StoryFragment" />
122     </xs:complexType>
123     <xs:complexType name="CT_Break">
124     </xs:complexType>
125     <xs:complexType name="CT_Section">
126         <xs:choice maxOccurs="unbounded">
127             <xs:element ref="ParagraphStructure" />
128             <xs:element ref="ListStructure" />
129             <xs:element ref="TableStructure" />
130             <xs:element ref="FigureStructure" />
131         </xs:choice>
132     </xs:complexType>
133     <xs:complexType name="CT_Paragraph">
134         <xs:choice minOccurs="0" maxOccurs="unbounded">
135             <xs:element ref="NamedElement" />
136         </xs:choice>
137     </xs:complexType>
138     <xs:complexType name="CT_Table">
139         <xs:choice maxOccurs="unbounded">
140             <xs:element ref="TableRowGroupStructure" />
141         </xs:choice>
142     </xs:complexType>
143     <xs:complexType name="CT_TableRowGroup">
144         <xs:choice maxOccurs="unbounded">
145             <xs:element ref="TableRowStructure" />
146         </xs:choice>
147     </xs:complexType>
148     <xs:complexType name="CT_TableRow">
149         <xs:choice maxOccurs="unbounded">
150             <xs:element ref="TableCellStructure" />
151         </xs:choice>
152     </xs:complexType>
153     <xs:complexType name="CT_TableCell">

```

```

154     <xs:choice minOccurs="0" maxOccurs="unbounded">
155         <xs:element ref="ParagraphStructure" />
156         <xs:element ref="ListStructure" />
157         <xs:element ref="TableStructure" />
158         <xs:element ref="FigureStructure" />
159     </xs:choice>
160     <xs:attributeGroup ref="AG_TableCell" />
161 </xs:complexType>
162 <xs:complexType name="CT_List">
163     <xs:choice maxOccurs="unbounded">
164         <xs:element ref="ListItemStructure" />
165     </xs:choice>
166 </xs:complexType>
167 <xs:complexType name="CT_ListItem">
168     <xs:choice minOccurs="0" maxOccurs="unbounded">
169         <xs:element ref="ParagraphStructure" />
170         <xs:element ref="ListStructure" />
171         <xs:element ref="TableStructure" />
172         <xs:element ref="FigureStructure" />
173     </xs:choice>
174     <xs:attributeGroup ref="AG_ListItem" />
175 </xs:complexType>
176 <xs:complexType name="CT_Figure">
177     <xs:choice minOccurs="0" maxOccurs="unbounded">
178         <xs:element ref="NamedElement" />
179     </xs:choice>
180 </xs:complexType>
181 <xs:complexType name="CT_NamedElement">
182     <xs:attributeGroup ref="AG_NamedElement" />
183 </xs:complexType>
184 <!-- Simple Types -->
185 <!-- FragmentType enumeration -->
186 <xs:simpleType name="ST_FragmentType">
187     <xs:restriction base="xs:string">
188         <xs:enumeration value="Content" />
189         <xs:enumeration value="Header" />
190         <xs:enumeration value="Footer" />
191     </xs:restriction>
192 </xs:simpleType>
193 <xs:simpleType name="ST_Location">
194     <xs:restriction base="xs:string">
195         <xs:pattern value="([0-9][0-9]*)(\,[0-9][0-9]*)*" />
196     </xs:restriction>
197 </xs:simpleType>
198 <xs:simpleType name="ST_TableSpan">
199     <xs:restriction base="xs:int">
200         <xs:minInclusive value="1" />
201     </xs:restriction>
202 </xs:simpleType>
203 <xs:simpleType name="ST_ElementIndex">
204     <xs:restriction base="xs:int">
205         <xs:minInclusive value="0" />
206     </xs:restriction>
207 </xs:simpleType>
208 <!-- Elements -->

```



```

209     <xs:element name="StoryFragments" type="CT_StoryFragments">
210         </xs:element>
211     <xs:element name="StoryFragment" type="CT_StoryFragment">
212         </xs:element>
213     <xs:element name="StoryBreak" type="CT_Break">
214         </xs:element>
215     <xs:element name="SectionStructure" type="CT_Section">
216         </xs:element>
217     <xs:element name="ParagraphStructure" type="CT_Paragraph">
218         </xs:element>
219     <xs:element name="TableStructure" type="CT_Table">
220         </xs:element>
221     <xs:element name="TableRowGroupStructure" type="CT_TableRowGroup">
222         </xs:element>
223     <xs:element name="TableRowStructure" type="CT_TableRow">
224         </xs:element>
225     <xs:element name="TableCellStructure" type="CT_TableCell">
226         </xs:element>
227     <xs:element name="ListStructure" type="CT_List">
228         </xs:element>
229     <xs:element name="ListItemStructure" type="CT_ListItem">
230         </xs:element>
231     <xs:element name="FigureStructure" type="CT_Figure">
232         </xs:element>
233     <xs:element name="NamedElement" type="CT_NamedElement">
234         </xs:element>
235     <!-- Attribute Groups -->
236     <xs:attributeGroup name="AG_StoryFragment">
237         <xs:attribute name="StoryName" type="xs:string" use="optional">
238             </xs:attribute>
239         <xs:attribute name="FragmentName" type="xs:string" use="optional">
240             </xs:attribute>
241         <xs:attribute name="FragmentType" type="ST_FragmentType" use="required">
242             </xs:attribute>
243     </xs:attributeGroup>
244     <xs:attributeGroup name="AG_TableCell">
245         <xs:attribute name="RowSpan" type="ST_TableSpan" use="optional"
246 default="1">
247             </xs:attribute>
248         <xs:attribute name="ColumnSpan" type="ST_TableSpan" use="optional"
249 default="1">
250             </xs:attribute>
251     </xs:attributeGroup>
252     <xs:attributeGroup name="AG_ListItem">
253         <xs:attribute name="Marker" type="ST_NameUnique" use="optional">
254             </xs:attribute>
255     </xs:attributeGroup>
256     <xs:attributeGroup name="AG_NamedElement">
257         <xs:attribute name="NameReference" type="ST_Name" use="required">
258             </xs:attribute>
259     </xs:attributeGroup>
260 </xs:schema>

```


1 **A.5 Discard Control** ~~W3C Schema~~

2 The schema shown below is also provided in electronic form as a file named DiscardControl.xsd,
3 which is contained in an accompanying zip archive named "XPS WC3 Schemas.zip". If
4 discrepancies exist between the representation as published below and the corresponding
5 electronic version, the published version below is the definitive version.

6

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <xs:schema targetNamespace="http://schemas.microsoft.com/xps/2005/06/discard-control"
3 xmlns="http://schemas.microsoft.com/xps/2005/06/discard-control"
4 xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
5 blockDefault="#all">
6
7     <xs:complexType name="CT_DiscardControl">
8         <xs:sequence>
9             <xs:element ref="Discard" minOccurs="0" maxOccurs="unbounded" />
10        </xs:sequence>
11    </xs:complexType>
12
13    <xs:complexType name="CT_Discard">
14        <xs:attribute name="SentinelPage" type="xs:anyURI" use="required">
15        </xs:attribute>
16        <xs:attribute name="Target" type="xs:anyURI" use="required">
17        </xs:attribute>
18    </xs:complexType>
19
20    <xs:element name="DiscardControl" type="CT_DiscardControl">
21    </xs:element>
22
23    <xs:element name="Discard" type="CT_Discard">
24    </xs:element>
25
26 </xs:schema>
```

1 **A.6 3D-Graphic Content**~~W3C Schema~~

2 The schema shown below is also provided in electronic form as a file named OpenXPS3D.xsd,
3 which is contained in an accompanying zip archive named "XPS WC3 Schemas.zip". If
4 discrepancies exist between the representation as published below and the corresponding
5 electronic version, the published version below is the definitive version.

```

6
1 <?xml version="1.0" encoding="utf-8"?>
2 <!-- OpenXPS 3D Graphics content Schema prepared by Nathan Crews - Autodesk -->
3 <xs:schema xmlns="http://schemas.openxps.org/openxps/3d/2008"
4   xmlns:xps1="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   targetNamespace="http://schemas.openxps.org/openxps/3d/2008"
9   elementFormDefault="qualified" blockDefault="#all">
10 <!-- Import XPS and related XML Schemas -->
11 <xs:import namespace="http://schemas.microsoft.com/xps/2005/06"/>
12 <xs:import namespace=
13   "http://schemas.microsoft.com/xps/2005/06/resourcedictionary-key"/>
14 <xs:import namespace="http://www.w3.org/XML/1998/namespace"/>
15 <!-- Names used for types and groups follows XPS 1.0 pattern:
16     ST_*         simpleType
17     CT_*         complexType
18     G_*          group
19     AG_*         attributeGroup   -->
20 <!-- Complex Types -->
21 <xs:complexType name="CT_Brush3D">
22   <xs:sequence>
23     <xs:element ref="xps1:ImageBrush.Transform" minOccurs="0"/>
24   </xs:sequence>
25   <xs:attributeGroup ref="AG_Brush3D"/>
26 </xs:complexType>
27 <!-- Root elements -->
28 <xs:element name="Brush3D">
29   <xs:complexType>
30     <xs:complexContent>
31       <xs:extension base="CT_Brush3D"/>
32     </xs:complexContent>
33   </xs:complexType>
34 </xs:element>
35 <!-- Attribute Groups -->
36 <xs:attributeGroup name="AG_Brush3D">
37   <xs:attribute name="Source3D" type="ST_UriImage3D" use="required"/>
38   <xs:attribute ref="x:Key"/>
39   <xs:attribute name="Transform" type="xps1:ST_RscRefMatrix"/>
40   <xs:attribute name="Viewbox" type="xps1:ST_ViewBox" use="required"/>
41   <xs:attribute name="Viewport" type="xps1:ST_ViewBox" use="required"/>
42   <xs:attribute name="ViewboxUnits" type="xps1:ST_ViewUnits" use="required"
43     fixed="Absolute"/>
44   <xs:attribute name="ViewportUnits" type="xps1:ST_ViewUnits" use="required"
45     fixed="Absolute"/>

```

```
46 </xs:attributeGroup>
47 <!-- Simple Types -->
48 <xs:simpleType name="ST_UriImage3D">
49   <xs:restriction base="xs:anyURI">
50     <xs:pattern value="([^\{].*)?"/>
51   </xs:restriction>
52 </xs:simpleType>
53 </xs:schema>
```


1 **B. Schemas – RELAX NG**

2 **This annex is informative**

3 **A-7B.1 Signature Definitions**

4 The schema shown below is also provided in electronic form as a file named
5 SignatureDefinitions.rnc, which is contained in an accompanying zip archive named "XPS RELAX
6 NG Schemas.zip". If discrepancies exist between the representation as published below and the
7 corresponding electronic version, the published version below is the definitive version.

8 <<Schema goes here>>

1 **A.8B.2 XPS Document**

2 [The schema shown below is also provided in electronic form as a file named XPSDocument.rnc,](#)
3 [which is contained in an accompanying zip archive named "XPS RELAX NG Schemas.zip". If](#)
4 [discrepancies exist between the representation as published below and the corresponding](#)
5 [electronic version, the published version below is the definitive version.](#)

6 <<Schema goes here>>

1 **A.9B.3 Resource Dictionary Key**

2 [The schema shown below is also provided in electronic form as a file named](#)
3 [ResourceDictionaryKey.rnc, which is contained in an accompanying zip archive named "XPS](#)
4 [RELAX NG Schemas.zip". If discrepancies exist between the representation as published below](#)
5 [and the corresponding electronic version, the published version below is the definitive version.](#)

6 <<Schema goes here>>

1 **~~A.10~~B.4 Document Structure**

2 [The schema shown below is also provided in electronic form as a file named](#)
3 [DocumentStructure.rnc, which is contained in an accompanying zip archive named "XPS RELAX](#)
4 [NG Schemas.zip". If discrepancies exist between the representation as published below and the](#)
5 [corresponding electronic version, the published version below is the definitive version.](#)

6 <<Schema goes here>>

1 **~~A.11~~B.5 Discard Control**

2 [The schema shown below is also provided in electronic form as a file named DiscardControl.xsd,](#)
3 [which is contained in an accompanying zip archive named "XPS RELAX NG Schemas.zip". If](#)
4 [discrepancies exist between the representation as published below and the corresponding](#)
5 [electronic version, the published version below is the definitive version.](#)

6 <<Schema goes here>>

1 **A.12B.6 3D-Graphic Content**

2 [The schema shown below is also provided in electronic form as a file named OpenXPS3D.xsd,](#)
3 [which is contained in an accompanying zip archive named "XPS RELAX NG Schemas.zip".](#) [If](#)
4 [discrepancies exist between the representation as published below and the corresponding](#)
5 [electronic version, the published version below is the definitive version.](#)

1 <<Schema goes here>>

2 **End of informative text.**

1 **B.C. Abbreviated Geometry Syntax Algorithm**

2 A path geometry specified using the abbreviated geometry syntax (see §11.2.3) is equivalent
 3 to a path specified using a path geometry. The following algorithm describes how the
 4 abbreviated path syntax can be transformed into a path geometry containing path figures that,
 5 in turn, contain various segments.

6 This algorithm assumes that the presented string is well-formed according to the markup
 7 schema. Whitespace skipping is assumed without being explicitly spelled out in the algorithm.

```

8
9   Let CURRENTPOINT = 0,0
10  Create a new PathGeometry PG
11  PG.FillRule = EvenOdd
12  Let CURRENTPATHFIGURE = undefined
13
14  Read input character CH
15
16  if ( CH == 'F' )
17  {
18      Read input character CH
19      if ( CH == '0' )
20      {
21          PG.FillRule = EvenOdd
22      }
23      else
24      {
25          PG.FillRule = NonZero
26      }
27  }
28  else
29  {
30      GOTO label_first
31  }
32
33  label_repeat:
34  Read input character CH
35
36  label_first:
37
38  if ( CH == 'm' )
39  {
40      Read relative coordinate pair DX,DY
41      Let CURRENTPOINT.X = CURRENTPOINT.X + DX
42      Let CURRENTPOINT.Y = CURRENTPOINT.Y + DY
43      Create a new PathFigure CURRENTPATHFIGURE and add to PG
44      Let attribute CURRENTPATHFIGURE.StartPoint = CURRENTPOINT
45  }
46  else if ( CH == 'M' )
47  {
48      Read coordinate pair X,Y
49      Let CURRENTPOINT.X = X
50      Let CURRENTPOINT.Y = Y
51      Create a new PathFigure CURRENTPATHFIGURE and add to PG
52      Let attribute CURRENTPATHFIGURE.StartPoint = CURRENTPOINT
53  }
54  else if ( CH == 'l' )
55  {
56      Create new PolyLineSegment S
57      Add S to CURRENTPATHFIGURE
58  label_1:

```

```

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

```

```

1          Add CURRENTPOINT.X,CURRENTPOINT.Y to end of S.Points attribute list
2          if ( next character is not a letter )
3          {
4              GOTO label_6
5          }
6      }
7      else if ( CH == 'c' )
8      {
9          Create new PolyBezierSegment S
10         Add S to CURRENTPATHFIGURE
11     label_7:
12         Read relative coordinate pair DX,DY
13         Let POINT.X = CURRENTPOINT.X + DX
14         Let POINT.Y = CURRENTPOINT.Y + DY
15         Add POINT.X, POINT.Y to end of S.Points attribute list
16         Read coordinate pair DX,DY
17         Let POINT.X = CURRENTPOINT.X + DX
18         Let POINT.Y = CURRENTPOINT.Y + DY
19         Add POINT.X, POINT.Y to end of S.Points attribute list
20         Read coordinate pair DX,DY
21         Let CURRENTPOINT.X = CURRENTPOINT.X + DX
22         Let CURRENTPOINT.Y = CURRENTPOINT.Y + DY
23         Add CURRENTPOINT.X,CURRENTPOINT.Y to end of S.Points attribute list
24         if ( next character is not a letter )
25         {
26             GOTO label_7
27         }
28     }
29     else if ( CH == 'C' )
30     {
31         Create new PolyBezierSegment S
32         Add S to CURRENTPATHFIGURE
33     label_8:
34         Read coordinate pair X,Y
35         Let POINT.X = X
36         Let POINT.Y = Y
37         Add POINT.X, POINT.Y to end of S.Points attribute list
38         Read coordinate pair X,Y
39         Let POINT.X = X
40         Let POINT.Y = Y
41         Add POINT.X, POINT.Y to end of S.Points attribute list
42         Read coordinate pair X,Y
43         Let CURRENTPOINT.X = X
44         Let CURRENTPOINT.Y = Y
45         Add CURRENTPOINT.X,CURRENTPOINT.Y to end of S.Points attribute list
46         if ( next character is not a letter )
47         {
48             GOTO label_8
49         }
50     }
51     else if ( CH == 'q' )
52     {
53         Create new PolyQuadraticBezierSegment S
54         Add S to CURRENTPATHFIGURE
55     label_9:
56         Read relative coordinate pair DX,DY
57         Let POINT.X = CURRENTPOINT.X + DX
58         Let POINT.Y = CURRENTPOINT.Y + DY
59         Add POINT.X, POINT.Y to end of S.Points attribute list
60         Read relative coordinate pair DX,DY
61         Let CURRENTPOINT.X = CURRENTPOINT.X + DX
62         Let CURRENTPOINT.Y = CURRENTPOINT.Y + DY
63         Add CURRENTPOINT.X,CURRENTPOINT.Y to end of S.Points attribute list
64         if ( next character is not a letter )
65         {
66             GOTO label_9
67         }
68     }

```

```

1      else ( if CH == 'Q' )
2      {
3          Create new PolyQuadraticBezierSegment S
4          Add S to CURRENTPATHFIGURE
5      label_10:
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_10
17         }
18     }
19     else if ( CH == 's' )
20     {
21         Create new PolyBezierSegment S
22         Add S to CURRENTPATHFIGURE
23     label_11:
24         if ( S.Points is non-empty )
25         {
26             Let LASTCTRLPOINT = Point before last point in S.Points
27             Let POINT.X = 2 * CURRENTPOINT.X - LASTCTRLPOINT.X
28             Let POINT.Y = 2 * CURRENTPOINT.Y - LASTCTRLPOINT.Y
29         }
30         else if ( segment before CURRENTPATHSEGMENT is a PolyBezierSegment )
31         {
32             Let LASTCTRLPOINT = Point before last point in previous PolyBezierSegment
33             Let POINT.X = 2 * CURRENTPOINT.X - LASTCTRLPOINT.X
34             Let POINT.Y = 2 * CURRENTPOINT.Y - LASTCTRLPOINT.Y
35         }
36         else
37         {
38             Let POINT = CURRENTPOINT
39         }
40
41         Add POINT.X, POINT.Y to end of S.Points attribute list
42         Read relative coordinate pair DX,DY
43         Let POINT.X = CURRENTPOINT.X + DX
44         Let POINT.Y = CURRENTPOINT.Y + DY
45         Add POINT.X, POINT.Y to end of S.Points attribute list
46         Read relative coordinate pair DX,DY
47         Let CURRENTPOINT.X = CURRENTPOINT.X + DX
48         Let CURRENTPOINT.Y = CURRENTPOINT.Y + DY
49         Add CURRENTPOINT.X,CURRENTPOINT.Y to end of S.Points attribute list
50         if ( next character is not a letter )
51         {
52             GOTO label_11
53         }
54     }
55     else if ( CH == 'S' )
56     {
57         Create new PolyBezierSegment S
58         Add S to CURRENTPATHFIGURE
59     label_12:
60         if ( S.Points is non-empty )
61         {
62             Let LASTCTRLPOINT = Point before last point in S.Points
63             Let POINT.X = 2 * CURRENTPOINT.X - LASTCTRLPOINT.X
64             Let POINT.Y = 2 * CURRENTPOINT.Y - LASTCTRLPOINT.Y
65         }
66         else if ( segment before CURRENTPATHSEGMENT is a PolyBezierSegment )
67         {
68             Let LASTCTRLPOINT = S.Point before last point in previous PolyBezierSegment
69             Let POINT.X = 2 * CURRENTPOINT.X - LASTCTRLPOINT.X

```

```

1           Let POINT.Y = 2 * CURRENTPOINT.Y - LASTCTRLPOINT.Y
2       }
3   else
4       {
5           Let POINT = CURRENTPOINT
6       }
7
8       Add POINT.X, POINT.Y to end of S.Points attribute list
9       Read coordinate pair X,Y
10      Let POINT.X = X
11      Let POINT.Y = Y
12      Add POINT.X, POINT.Y to end of S.Points attribute list
13      Read coordinate pair X,Y
14      Let CURRENTPOINT.X = X
15      Let CURRENTPOINT.Y = Y
16      Add CURRENTPOINT.X,CURRENTPOINT.Y to end of S.Points attribute list
17      if ( next character is not a letter )
18          {
19              GOTO label_12
20          }
21      }
22  else if ( CH == 'a' )
23      {
24  label_13:
25      Create new ArcSegment S
26      Add S to CURRENTPATHFIGURE
27      Read Radius Pair RX,RY
28      Read Rotation ROT
29      Read integer FLAG1
30      Read integer FLAG2
31      Read relative coordinate pair DX,DY
32      Let CURRENTPOINT.X = CURRENTPOINT.X + DX
33      Let CURRENTPOINT.Y = CURRENTPOINT.Y + DY
34      Let S.Point = CURRENTPOINT.X,CURRENTPOINT.Y
35      Let S.IsLargeArc = (FLAG1 == 1 ? true : false)
36      Let S.SweepDirection = (FLAG2 == 1 ? Clockwise : Counterclockwise)
37      Let S.RotationAngle = ROT
38      Let S.Size = RX, RY
39      if ( next character is not a letter )
40          {
41              GOTO label_13
42          }
43      }
44  else if ( CH == 'A' )
45      {
46  label_14:
47      Create new ArcSegment S
48      Add S to CURRENTPATHFIGURE
49      Read Radius Pair RX,RY
50      Read Rotation ROT
51      Read integer FLAG1
52      Read integer FLAG2
53      Read coordinate pair X,Y
54      Let CURRENTPOINT.X = X
55      Let CURRENTPOINT.Y = Y
56      Let S.Point = CURRENTPOINT.X,CURRENTPOINT.Y
57      Let S.IsLargeArc = (FLAG1 == 1 ? true : false)
58      Let S.SweepDirection = (FLAG2 == 1 ? Clockwise : Counterclockwise)
59      Let S.RotationAngle = ROT
60      Let S.Size = RX, RY
61      if ( next character is not a letter )
62          {
63              GOTO label_14
64          }
65      }
66  else if ( CH == 'z' or CH == 'Z' )
67      {
68      Let attribute CURRENTPATHFIGURE.IsClosed = true
69      Let CURRENTPOINT = First point of first segment of CURRENTPATHFIGURE

```

```

1           Let CURRENTPATHFIGURE = undefined
2       }
3   /* This case can not occur, because the input is assumed to be well-formed according to the markup
4   schema
5       else
6       {
7           ERROR: Invalid input character
8       }
9   */
10
11   if ( End of input reached )
12   {
13       Terminate algorithm, return PG
14   }
15   else
16   {
17       GOTO label_repeat
18   }

```

19

20 ~~C. scRGB Gamut Boundary Definition~~

```

21 <?xml version="1.0" encoding="utf-8" ?>
22 <cdm:ColorDeviceModel
23   ID="http://schemas.microsoft.com/windows/2005/02/color/wcsRGB.cdmp"
24   xmlns:cdm="http://schemas.microsoft.com/windows/2005/02/color/"
25   ColorDeviceModel"
26   xmlns:wcs="http://schemas.microsoft.com/windows/2005/02/color/"
27   WcsCommonProfileTypes"
28   xmlns:xs="http://www.w3.org/2001/XMLSchema-instance">
29
30   <cdm:ProfileName>
31     <wcs:Text xml:lang="en-US">wcsRGB virtual device model profile</wcs:Text>
32   </cdm:ProfileName>
33   <cdm:Description>
34     <wcs:Text xml:lang="en-US">wcsRGB virtual s2.13 device model system
35 profile</wcs:Text>
36   </cdm:Description>
37   <cdm:Author>
38     <wcs:Text xml:lang="en-US">Microsoft Corporation</wcs:Text>
39   </cdm:Author>
40
41   <cdm:MeasurementConditions>
42     <cdm:ColorSpace>CIEXYZ</cdm:ColorSpace>
43     <cdm:WhitePointName>D65</cdm:WhitePointName>
44   </cdm:MeasurementConditions>
45   <cdm:SelfLuminous>true</cdm:SelfLuminous>
46   <cdm:MaxColorant>4.0</cdm:MaxColorant>
47   <cdm:MinColorant>4.0</cdm:MinColorant>
48
49   <cdm:RGBVirtualDevice>
50     <cdm:MeasurementData TimeStamp="2005-02-09T22:00:00">
51       <cdm:WhitePrimary X="95.05" Y="100.00" Z="108.90" />
52       <cdm:RedPrimary X="41.24" Y="21.26" Z="1.93" />
53       <cdm:GreenPrimary X="35.76" Y="71.52" Z="11.92" />

```

```

1  <del><cdm:BluePrimary X="18.05" Y="7.22" Z="95.05" />
2  <del><cdm:BlackPrimary X="0" Y="0" Z="0" />
3  <del><cdm:Gamma value="1.0" />
4  <del><cdm:GamutBoundarySamples>
5  <del><cdm:RGB R="0.166433" G="0.016114" B="0.027918" />
6  <del><cdm:RGB R="0.257134" G="0.026542" B="0.041950" />
7  <del><cdm:RGB R="0.373396" G="0.039395" B="0.060310" />
8  <del><cdm:RGB R="0.516590" G="0.054207" B="0.083665" />
9  <del><cdm:RGB R="0.161885" G="0.013701" B="0.015224" />
10 <del><cdm:RGB R="0.254873" G="0.024242" B="0.022314" />
11 <del><cdm:RGB R="0.370591" G="0.036207" B="0.031873" />
12 <del><cdm:RGB R="0.645317" G="0.053422" B="0.061566" />
13 <del><cdm:RGB R="0.246052" G="0.020570" B="0.009804" />
14 <del><cdm:RGB R="0.366951" G="0.033595" B="0.013350" />
15 <del><cdm:RGB R="0.513086" G="0.047777" B="0.018342" />
16 <del><cdm:RGB R="0.348159" G="0.027092" B="0.002697" />
17 <del><cdm:RGB R="0.501415" G="0.043030" B="0.003038" />
18 <del><cdm:RGB R="0.204158" G="0.007054" B="0.001921" />
19 <del><cdm:RGB R="0.102843" G="0.005460" B="0.002574" />
20 <del><cdm:RGB R="0.161680" G="0.005828" B="0.003751" />
21 <del><cdm:RGB R="0.241218" G="0.005609" B="0.005263" />
22 <del><cdm:RGB R="0.345298" G="0.004589" B="0.007162" />
23 <del><cdm:RGB R="1.381166" G="0.758736" B="0.088389" />
24 <del><cdm:RGB R="0.747224" G="0.490147" B="0.065045" />
25 <del><cdm:RGB R="0.892023" G="0.585517" B="0.076908" />
26 <del><cdm:RGB R="1.054700" G="0.693668" B="0.087994" />
27 <del><cdm:RGB R="1.237403" G="0.815408" B="0.099715" />
28 <del><cdm:RGB R="0.429786" G="0.463528" B="0.056073" />
29 <del><cdm:RGB R="0.109329" G="0.215829" B="0.023359" />
30 <del><cdm:RGB R="0.170974" G="0.342970" B="0.037798" />
31 <del><cdm:RGB R="0.032781" G="0.182702" B="0.017604" />
32 <del><cdm:RGB R="0.040703" G="0.236387" B="0.023182" />
33 <del><cdm:RGB R="0.052010" G="0.299807" B="0.029289" />
34 <del><cdm:RGB R="0.015232" G="0.150695" B="0.012907" />
35 <del><cdm:RGB R="0.020689" G="0.198705" B="0.017162" />
36 <del><cdm:RGB R="0.086126" G="0.133080" B="0.007658" />
37 <del><cdm:RGB R="0.178409" G="0.235796" B="0.126003" />
38 <del><cdm:RGB R="0.000000" G="0.000000" B="0.000000" />
39 <del><cdm:RGB R="0.150496" G="0.178902" B="0.156523" />
40 <del><cdm:RGB R="0.193313" G="0.234293" B="0.204426" />
41 <del><cdm:RGB R="0.202798" G="0.230449" B="0.290886" />
42 <del><cdm:RGB R="0.211558" G="0.225780" B="0.384316" />
43 <del><cdm:RGB R="0.249008" G="0.286162" B="0.470316" />
44 <del><cdm:RGB R="1.024267" G="0.995126" B="0.956518" />
45 <del><cdm:RGB R="0.211156" G="0.218574" B="0.473918" />
46 <del><cdm:RGB R="0.209022" G="0.210461" B="0.567755" />
47 <del><cdm:RGB R="0.368209" G="0.814614" B="1.020972" />
48 <del><cdm:RGB R="0.194449" G="0.199547" B="0.648518" />
49 <del><cdm:RGB R="0.099507" G="0.602350" B="1.001310" />
50 <del><cdm:RGB R="0.447496" G="0.790515" B="1.022822" />
51 <del><cdm:RGB R="0.123046" G="0.108061" B="0.470343" />
52 <del><cdm:RGB R="0.150585" G="0.144010" B="0.596962" />
53 <del><cdm:RGB R="0.185831" G="0.575754" B="1.007060" />
54 <del><cdm:RGB R="0.331216" G="0.668936" B="1.023494" />
55 <del><cdm:RGB R="0.410196" G="0.645807" B="1.014063" />

```

1 ~~-----<cdm:RGB R="0.574938" G="0.752116" B="1.021593" />~~

2 ~~-----<cdm:RGB R="0.483140" G="0.624458" B="1.005310" />~~

3 ~~-----<cdm:RGB R="0.635447" G="0.734762" B="1.010061" />~~

4 ~~-----<cdm:RGB R="0.052205" G="0.035138" B="0.436365" />~~

5 ~~-----<cdm:RGB R="1.024267" G="0.995126" B="0.956518" />~~

6 ~~-----<cdm:RGB R="0.028844" G="0.005277" B="0.123163" />~~

7 ~~-----<cdm:RGB R="0.049636" G="0.010904" B="0.228673" />~~

8 ~~-----<cdm:RGB R="0.080157" G="0.012435" B="0.135751" />~~

9 ~~-----<cdm:RGB R="0.127487" G="0.020154" B="0.216960" />~~

10 ~~-----<cdm:RGB R="0.190206" G="0.029844" B="0.323182" />~~

11 ~~-----<cdm:RGB R="0.097580" G="0.013313" B="0.083401" />~~

12 ~~-----<cdm:RGB R="0.155931" G="0.021750" B="0.133422" />~~

13 ~~-----<cdm:RGB R="0.230038" G="0.031267" B="0.196426" />~~

14 ~~-----<cdm:RGB R="0.326426" G="0.043430" B="0.278289" />~~

15 ~~-----<cdm:RGB R="0.426090" G="0.050043" B="0.360668" />~~

16 ~~-----<cdm:RGB R="0.098869" G="0.010753" B="0.048113" />~~

17 ~~-----<cdm:RGB R="0.164545" G="0.019812" B="0.079230" />~~

18 ~~-----<cdm:RGB R="0.249557" G="0.030741" B="0.119834" />~~

19 ~~-----<cdm:RGB R="0.357564" G="0.043900" B="0.171719" />~~

20 ~~-----<cdm:RGB R="0.495437" G="0.060672" B="0.237960" />~~

21 ~~-----<cdm:RGB R="0.619457" G="0.065550" B="0.301872" />~~

22 ~~-----<cdm:RGB R="0.000000" G="0.000000" B="0.000000" />~~

23 ~~-----<cdm:RGB R="0.169101" G="0.018572" B="0.047904" />~~

24 ~~-----<cdm:RGB R="0.255247" G="0.028467" B="0.071995" />~~

25 ~~-----<cdm:RGB R="0.368932" G="0.041653" B="0.103698" />~~

26 ~~-----<cdm:RGB R="0.508606" G="0.056762" B="0.143412" />~~

27 ~~-----<cdm:RGB R="0.637110" G="0.061275" B="0.186481" />~~

28 ~~-----<cdm:RGB R="1.249149" G="0.024436" B="0.002194" />~~

29 ~~-----<cdm:RGB R="0.000475" G="0.001355" B="0.043728" />~~

30 ~~-----<cdm:RGB R="1.257972" G="0.026125" B="0.049916" />~~

31 ~~-----<cdm:RGB R="1.264195" G="0.027432" B="0.104713" />~~

32 ~~-----<cdm:RGB R="0.003291" G="0.009926" B="0.365422" />~~

33 ~~-----<cdm:RGB R="1.287764" G="0.033804" B="0.414704" />~~

34 ~~-----<cdm:RGB R="0.004536" G="0.013745" B="0.512430" />~~

35 ~~-----<cdm:RGB R="1.297183" G="0.037024" B="0.573653" />~~

36 ~~-----<cdm:RGB R="1.307374" G="0.040929" B="0.763754" />~~

37 ~~-----<cdm:RGB R="0.007818" G="0.023805" B="0.902837" />~~

38 ~~-----<cdm:RGB R="0.001325" G="0.002612" B="0.000199" />~~

39 ~~-----<cdm:RGB R="0.007337" G="0.015263" B="0.001058" />~~

40 ~~-----<cdm:RGB R="0.005430" G="0.000728" B="0.906022" />~~

41 ~~-----<cdm:RGB R="0.020327" G="0.043021" B="0.002887" />~~

42 ~~-----<cdm:RGB R="0.042091" G="0.089818" B="0.005932" />~~

43 ~~-----<cdm:RGB R="0.117963" G="0.253649" B="0.016500" />~~

44 ~~-----<cdm:RGB R="0.141327" G="0.262090" B="0.943868" />~~

45 ~~-----<cdm:RGB R="0.174730" G="0.376455" B="0.024386" />~~

46 ~~-----<cdm:RGB R="0.204641" G="0.390245" B="0.954021" />~~

47 ~~-----<cdm:RGB R="0.245665" G="0.530007" B="0.034229" />~~

48 ~~-----<cdm:RGB R="0.282073" G="0.548898" B="0.962351" />~~

49 ~~-----<cdm:RGB R="1.150086" G="0.568927" B="1.001779" />~~

50 ~~-----<cdm:RGB R="0.374737" G="0.740509" B="0.968358" />~~

51 ~~-----<cdm:RGB R="1.083883" G="0.766726" B="1.001175" />~~

52 ~~-----<cdm:RGB R="1.002593" G="0.985026" B="0.073045" />~~

53 ~~-----<cdm:RGB R="0.434478" G="0.938894" B="0.060390" />~~

54 ~~-----<cdm:RGB R="0.000055" G="0.000136" B="0.003135" />~~

55 ~~-----<cdm:RGB R="0.000203" G="0.000555" B="0.016188" />~~

1 ~~————— <cdm:RGB R="0.000896" G="0.002619" B="0.089419" />~~

2 ~~————— <cdm:RGB R="0.001492" G="0.004425" B="0.156418" />~~

3 ~~————— <cdm:RGB R="0.002283" G="0.006839" B="0.247555" />~~

4 ~~————— <cdm:RGB R="0.006038" G="0.018353" B="0.690852" />~~

5 ~~————— <cdm:RGB R="0.007818" G="0.023805" B="0.902837" />~~

6 ~~————— <cdm:RGB R="0.483692" G="0.967403" B="0.971726" />~~

7 ~~————— <cdm:RGB R="0.003728" G="0.000194" B="0.002807" />~~

8 ~~————— <cdm:RGB R="0.004140" G="0.000555" B="0.015782" />~~

9 ~~————— <cdm:RGB R="0.004738" G="0.001328" B="0.043372" />~~

10 ~~————— <cdm:RGB R="0.005489" G="0.002582" B="0.089155" />~~

11 ~~————— <cdm:RGB R="0.006397" G="0.004383" B="0.156254" />~~

12 ~~————— <cdm:RGB R="0.007479" G="0.006795" B="0.247483" />~~

13 ~~————— <cdm:RGB R="0.008758" G="0.009881" B="0.365433" />~~

14 ~~————— <cdm:RGB R="0.010255" G="0.013700" B="0.512516" />~~

15 ~~————— <cdm:RGB R="0.011993" G="0.018308" B="0.691003" />~~

16 ~~————— <cdm:RGB R="0.013995" G="0.023760" B="0.903048" />~~

17 ~~————— <cdm:RGB R="0.034050" G="0.018254" B="0.692249" />~~

18 ~~————— <cdm:RGB R="0.036746" G="0.023689" B="0.904397" />~~

19 ~~————— <cdm:RGB R="0.080545" G="0.023830" B="0.908226" />~~

20 ~~————— <cdm:RGB R="0.149752" G="0.024452" B="0.915123" />~~

21 ~~————— <cdm:RGB R="1.002593" G="0.985026" B="0.073045" />~~

22 ~~————— <cdm:RGB R="1.249149" G="0.024436" B="0.002194" />~~

23 ~~————— <cdm:RGB R="1.252780" G="0.025117" B="0.015342" />~~

24 ~~————— <cdm:RGB R="1.257972" G="0.026125" B="0.049916" />~~

25 ~~————— <cdm:RGB R="1.271263" G="0.029093" B="0.182298" />~~

26 ~~————— <cdm:RGB R="1.279122" G="0.031189" B="0.284927" />~~

27 ~~————— <cdm:RGB R="1.002258" G="0.999362" B="0.998757" />~~

28 ~~————— <cdm:RGB R="0.007818" G="0.023805" B="0.902837" />~~

29 ~~————— <cdm:RGB R="0.001325" G="0.002612" B="0.000199" />~~

30 ~~————— <cdm:RGB R="0.004903" G="0.018821" B="0.903364" />~~

31 ~~————— <cdm:RGB R="0.005430" G="0.000728" B="0.906022" />~~

32 ~~————— <cdm:RGB R="0.024203" G="0.033431" B="0.912191" />~~

33 ~~————— <cdm:RGB R="0.042091" G="0.089818" B="0.005932" />~~

34 ~~————— <cdm:RGB R="0.052335" G="0.086586" B="0.921578" />~~

35 ~~————— <cdm:RGB R="0.074175" G="0.159024" B="0.010406" />~~

36 ~~————— <cdm:RGB R="0.090966" G="0.161809" B="0.932665" />~~

37 ~~————— <cdm:RGB R="0.117963" G="0.253649" B="0.016500" />~~

38 ~~————— <cdm:RGB R="0.174730" G="0.376455" B="0.024386" />~~

39 ~~————— <cdm:RGB R="0.204641" G="0.390245" B="0.954021" />~~

40 ~~————— <cdm:RGB R="0.331890" G="0.716723" B="0.046182" />~~

41 ~~————— <cdm:RGB R="0.374737" G="0.740509" B="0.968358" />~~

42 ~~————— <cdm:RGB R="0.434478" G="0.938894" B="0.060390" />~~

43 ~~————— <cdm:RGB R="0.003567" G="0.000129" B="0.000031" />~~

44 ~~————— <cdm:RGB R="0.013995" G="0.023760" B="0.903048" />~~

45 ~~————— <cdm:RGB R="0.011145" G="0.018782" B="0.903675" />~~

46 ~~————— <cdm:RGB R="0.001010" G="0.000696" B="0.906548" />~~

47 ~~————— <cdm:RGB R="0.017488" G="0.033476" B="0.912853" />~~

48 ~~————— <cdm:RGB R="0.045318" G="0.086662" B="0.922243" />~~

49 ~~————— <cdm:RGB R="0.083641" G="0.161924" B="0.933260" />~~

50 ~~————— <cdm:RGB R="0.133701" G="0.262246" B="0.944373" />~~

51 ~~————— <cdm:RGB R="0.196722" G="0.390440" B="0.954439" />~~

52 ~~————— <cdm:RGB R="0.020512" G="0.000542" B="0.000142" />~~

53 ~~————— <cdm:RGB R="0.034096" G="0.018719" B="0.905266" />~~

54 ~~————— <cdm:RGB R="0.024587" G="0.000632" B="0.908671" />~~

55 ~~————— <cdm:RGB R="0.006983" G="0.033602" B="0.915280" />~~

1 ~~----- <cdm:RGB R=" 0.019836" G="0.086907" B="0.924618" />~~

2 ~~----- <cdm:RGB R=" 0.057114" G="0.162314" B="0.935374" />~~

3 ~~----- <cdm:RGB R=" 0.106133" G="0.262787" B="0.946166" />~~

4 ~~----- <cdm:RGB R=" 0.168134" G="0.391122" B="0.955918" />~~

5 ~~----- <cdm:RGB R="0.080545" G=" 0.023830" B="0.908226" />~~

6 ~~----- <cdm:RGB R="0.078174" G=" 0.018848" B="0.909282" />~~

7 ~~----- <cdm:RGB R="0.069557" G=" 0.000692" B="0.913080" />~~

8 ~~----- <cdm:RGB R="0.053304" G="0.033705" B="0.919825" />~~

9 ~~----- <cdm:RGB R="0.028089" G="0.087255" B="0.928902" />~~

10 ~~----- <cdm:RGB R=" 0.007469" G="0.162946" B="0.939149" />~~

11 ~~----- <cdm:RGB R=" 0.054727" G="0.263702" B="0.949360" />~~

12 ~~----- <cdm:RGB R="0.149752" G=" 0.024452" B="0.915123" />~~

13 ~~----- <cdm:RGB R="0.147668" G=" 0.019423" B="0.916222" />~~

14 ~~----- <cdm:RGB R="0.140009" G=" 0.001093" B="0.920043" />~~

15 ~~----- <cdm:RGB R="0.125292" G="0.033611" B="0.926547" />~~

16 ~~----- <cdm:RGB R="0.102016" G="0.087566" B="0.935050" />~~

17 ~~----- <cdm:RGB R="0.068642" G="0.163702" B="0.944526" />~~

18 ~~----- <cdm:RGB R="0.023701" G="0.264902" B="0.953919" />~~

19 ~~----- <cdm:RGB R="0.248626" G=" 0.025756" B="0.924685" />~~

20 ~~----- <cdm:RGB R="0.246801" G=" 0.020644" B="0.925712" />~~

21 ~~----- <cdm:RGB R="0.240032" G=" 0.002034" B="0.929243" />~~

22 ~~----- <cdm:RGB R="0.226828" G="0.033128" B="0.935165" />~~

23 ~~----- <cdm:RGB R="0.205577" G="0.087658" B="0.942824" />~~

24 ~~----- <cdm:RGB R="0.174609" G="0.164417" B="0.951306" />~~

25 ~~----- <cdm:RGB R="0.132331" G="0.266234" B="0.959693" />~~

26 ~~----- <cdm:RGB R="0.381162" G=" 0.027864" B="0.936087" />~~

27 ~~----- <cdm:RGB R="0.379554" G=" 0.022644" B="0.936982" />~~

28 ~~----- <cdm:RGB R="0.313886" G="0.164903" B="0.959216" />~~

29 ~~----- <cdm:RGB R="0.274412" G="0.267520" B="0.966469" />~~

30 ~~----- <cdm:RGB R="0.551062" G=" 0.030854" B="0.948537" />~~

31 ~~----- <cdm:RGB R="0.404402" G="0.399877" B="0.979322" />~~

32 ~~----- <cdm:RGB R="0.761768" G=" 0.034778" B="0.961424" />~~

33 ~~----- <cdm:RGB R="1.016507" G=" 0.039680" B="0.974324" />~~

34 ~~----- <cdm:RGB R="1.247925" G=" 0.024207" B="0.006804" />~~

35 ~~----- <cdm:RGB R="1.318331" G=" 0.045598" B="0.986957" />~~

36 ~~----- <cdm:RGB R="1.002593" G="0.985026" B=" 0.073045" />~~

37 ~~----- <cdm:RGB R=" 0.017084" G="0.004071" B="0.105192" />~~

38 ~~----- <cdm:RGB R="0.000276" G=" 0.000778" B="0.024704" />~~

39 ~~----- <cdm:RGB R="0.001189" G=" 0.003497" B="0.124447" />~~

40 ~~----- <cdm:RGB R="0.003835" G=" 0.011408" B="0.425034" />~~

41 ~~----- <cdm:RGB R="0.006525" G=" 0.019382" B="0.730759" />~~

42 ~~----- <cdm:RGB R="0.991838" G="0.854433" B="0.996305" />~~

43 ~~----- <cdm:RGB R=" 0.085296" G="0.980282" B="0.983420" />~~

44 ~~----- <cdm:RGB R="0.997109" G="1.001108" B="1.001729" />~~

45 ~~----- <cdm:RGB R="0.000038" G=" 0.000093" B="0.002143" />~~

46 ~~----- <cdm:RGB R="0.000087" G=" 0.000230" B="0.006229" />~~

47 ~~----- <cdm:RGB R="0.000627" G=" 0.001818" B="0.062146" />~~

48 ~~----- <cdm:RGB R="0.005522" G=" 0.016416" B="0.616931" />~~

49 ~~----- <cdm:RGB R="0.006525" G=" 0.019382" B="0.730759" />~~

50 ~~----- <cdm:RGB R="0.001838" G=" 0.000115" B="0.001922" />~~

51 ~~----- <cdm:RGB R="0.001968" G=" 0.000231" B="0.005969" />~~

52 ~~----- <cdm:RGB R="0.002147" G=" 0.000441" B="0.013224" />~~

53 ~~----- <cdm:RGB R="0.002370" G=" 0.000761" B="0.024478" />~~

54 ~~----- <cdm:RGB R="0.002634" G=" 0.001207" B="0.040491" />~~

55 ~~----- <cdm:RGB R="0.002944" G=" 0.001796" B="0.062002" />~~

```

1  <<cdm:RGB R="0.003302" G="0.002545" B="0.089732" />
2  <<cdm:RGB R="0.003715" G="0.003473" B="0.124383" />
3  <<cdm:RGB R="0.004188" G="0.004596" B="0.166643" />
4  <<cdm:RGB R="0.004728" G="0.005932" B="0.217190" />
5  <<cdm:RGB R="0.005341" G="0.007497" B="0.276684" />
6  <<cdm:RGB R="0.006813" G="0.011384" B="0.425118" />
7  <<cdm:RGB R="0.008657" G="0.016392" B="0.617056" />
8  <<cdm:RGB R="0.009736" G="0.019358" B="0.730905" />
9  <<cdm:RGB R="0.080359" G="0.980412" B="0.983525" />
10 <<cdm:RGB R="0.005604" G="0.000209" B="0.001935" />
11 <<cdm:RGB R="0.005786" G="0.000307" B="0.006004" />
12 <<cdm:RGB R="0.006041" G="0.000495" B="0.013270" />
13 <<cdm:RGB R="0.006363" G="0.000795" B="0.024529" />
14 <<cdm:RGB R="0.006745" G="0.001225" B="0.040547" />
15 <<cdm:RGB R="0.007184" G="0.001801" B="0.062069" />
16 <<cdm:RGB R="0.007679" G="0.002542" B="0.089815" />
17 <<cdm:RGB R="0.008231" G="0.003463" B="0.124487" />
18 <<cdm:RGB R="0.009523" G="0.005912" B="0.217344" />
19 <<cdm:RGB R="0.010273" G="0.007474" B="0.276866" />
20 <<cdm:RGB R="0.012011" G="0.011356" B="0.425354" />
21 <<cdm:RGB R="0.013011" G="0.013710" B="0.515596" />
22 <<cdm:RGB R="0.014108" G="0.016361" B="0.617344" />
23 <<cdm:RGB R="0.015308" G="0.019327" B="0.731220" />
24 <<cdm:RGB R="0.021793" G="0.013690" B="0.516185" />
25 <<cdm:RGB R="0.023073" G="0.016337" B="0.617959" />
26 <<cdm:RGB R="0.024454" G="0.019298" B="0.731861" />
27 <<cdm:RGB R="0.025943" G="0.022591" B="0.858499" />
28 <<cdm:RGB R="0.037804" G="0.019298" B="0.732989" />
29 <<cdm:RGB R="0.039522" G="0.022583" B="0.859653" />
30 <<cdm:RGB R="0.057988" G="0.022628" B="0.861439" />
31 <<cdm:RGB R="0.012314" G="0.982157" B="0.984935" />
32 <<cdm:RGB R="0.020932" G="0.982985" B="0.985609" />
33 <<cdm:RGB R="0.110705" G="0.985131" B="0.987376" />
34 <<cdm:RGB R="0.234831" G="0.987909" B="0.989706" />
35 <<cdm:RGB R="0.311005" G="0.989512" B="0.991076" />
36 <<cdm:RGB R="0.494043" G="0.993078" B="0.994197" />
37 <<cdm:RGB R="0.721366" G="0.997001" B="0.997773" />
38 <<cdm:RGB R="0.852925" G="0.999044" B="0.999710" />
39 <<cdm:RGB R="0.997109" G="1.001108" B="1.001729" />
40 <<cdm:RGB R="0.007645" G="0.022679" B="0.857328" />
41 <<cdm:RGB R="0.001742" G="0.000069" B="0.000016" />
42 <<cdm:RGB R="0.010928" G="0.022656" B="0.857495" />
43 <<cdm:RGB R="0.005453" G="0.000169" B="0.000042" />
44 <<cdm:RGB R="0.016619" G="0.022623" B="0.857831" />
45 <<cdm:RGB R="0.071870" G="0.980634" B="0.983705" />
46 <<cdm:RGB R="0.012251" G="0.000330" B="0.000086" />
47 <<cdm:RGB R="0.058141" G="0.980991" B="0.983991" />
48 <<cdm:RGB R="0.038488" G="0.981494" B="0.984397" />
49 <<cdm:RGB R="0.038332" G="0.000891" B="0.000240" />
50 <<cdm:RGB R="0.081991" G="0.022759" B="0.863942" />
51 <<cdm:RGB R="0.112193" G="0.023008" B="0.867198" />
52 <<cdm:RGB R="0.061770" G="0.983975" B="0.986421" />
53 <<cdm:RGB R="0.168229" G="0.986444" B="0.988470" />
54 <<cdm:RGB R="0.397243" G="0.991242" B="0.992576" />
55 <<cdm:RGB R="0.601915" G="0.995006" B="0.995934" />

```

```
1 | —————<cdm:RGB R="0.721366" G="0.997001" B="0.997773" />  
2 | —————<cdm:RGB R="0.849328" G="0.852650" B="0.994033" />  
3 | —————<cdm:RGB R="0.000000" G="0.000000" B="1.000000" />  
4 | —————</cdm:GamutBoundarySamples>  
5 | —————</cdm:MeasurementData>  
6 |  
7 | —————</cdm:RGBVirtualDevice>  
8 |  
9 | —————</cdm:ColorDeviceModel>  
10 |
```

1 D. Standard Namespaces and Content Types

2 The following tables list the namespaces and content types used in XPS packages and XPS
3 Documents.

4 D.1 XML Namespace URIs

5 *Table D-1. Package-wide namespaces*

Description	Namespace URI
Content Types	http://schemas.openxmlformats.org/package/2006/content-types
Core Properties	http://schemas.openxmlformats.org/package/2006/metadata/core-properties
Digital Signatures	http://schemas.openxmlformats.org/package/2006/digital-signature
Relationships	http://schemas.openxmlformats.org/package/2006/relationships
Markup Compatibility	http://schemas.openxmlformats.org/markup-compatibility/2006

6 *Table D-2. XPS Document namespaces*

Description	Namespace URI
DiscardControl	http://schemas.microsoft.com/xps/2005/06/discard-control
Document Structure	http://schemas.microsoft.com/xps/2005/06/documentstructure
FixedDocument	http://schemas.microsoft.com/xps/2005/06
FixedDocumentSequence	http://schemas.microsoft.com/xps/2005/06
FixedPage	http://schemas.microsoft.com/xps/2005/06
Print Schema Framework	http://schemas.microsoft.com/windows/2003/08/printing/printschemaframework
Print Schema Keywords	http://schemas.microsoft.com/windows/2003/08/printing/printschemakeywords
Resource Dictionary (Key attribute)	http://schemas.microsoft.com/xps/2005/06/resource-dictionary-key
Signature Definitions	http://schemas.microsoft.com/xps/2005/06/signature-definitions
Story Fragments	http://schemas.microsoft.com/xps/2005/06/documentstructure
targetNamespace	http://schemas.openxps.org/openxps/3d/2008
3D Graphics Content	http://schemas.openxps.org/openxps/3d/2008

D.2 Content Types

The content types in the tables below MUST NOT include parameters. ~~If a~~ consumer ~~encounters~~ ~~MUST treat~~ the presence of parameters on these content types ~~as an error~~ when the affected part is accessed [it MUST instantiate an error condition](#) [M12.7].

Table D-3. Package-wide content types

Description	Content type
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

Table D-4. XPS Document content types

Description	Content type
FixedDocument	application/vnd.ms-package.xps-fixeddocument+xml
FixedDocumentSequence	application/vnd.ms-package.xps-fixeddocumentssequence+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
X3D image unicode	model/x3d+xml
X3D image binary encoding	model/x3d+binary

1 **D.3 Relationship Types**

2 *Table D-5. Package-wide relationship types*

Description	Relationship type
Core Properties	http://schemas.openxmlformats.org/package/2006/relationships/metadata/core-properties
Digital Signature	http://schemas.openxmlformats.org/package/2006/relationships/digital-signature/signature
Digital Signature Certificate	http://schemas.openxmlformats.org/package/2006/relationships/digital-signature/certificate
Digital Signature Origin	http://schemas.openxmlformats.org/package/2006/relationships/digital-signature/origin
Thumbnail	http://schemas.openxmlformats.org/package/2006/relationships/metadata/thumbnail

3 *Table D-6. XPS Document relationship types*

Description	Relationship type
Digital Signature Definitions	http://schemas.microsoft.com/xps/2005/06/signature-definitions
DiscardControl	http://schemas.microsoft.com/xps/2005/06/discard-control
DocumentStructure	http://schemas.microsoft.com/xps/2005/06/documentstructure
PrintTicket	http://schemas.microsoft.com/xps/2005/06/printticket [pt7]
Required Resource	http://schemas.microsoft.com/xps/2005/06/required-resource
Restricted Font	http://schemas.microsoft.com/xps/2005/06/restricted-font
StartPart	http://schemas.microsoft.com/xps/2005/06/fixerepresentation
StoryFragments	http://schemas.microsoft.com/xps/2005/06/storyfragments

E. Conformance Requirements

This annex is informative

This annex ~~summarizes all~~restates the conformance requirements for producers and consumers implementing the XML Paper Specification. This restatement does not include conformance requirements from normative references such as ECMA-376:2006. ~~It is intended as a convenience; the text in the referenced clause or subclause is considered normative in all cases.~~

In this annex, conformance requirements are divided into three tables per clause, respectively containing the conformance requirements that producers and consumers must follow, those that they should follow, and those that are optional. Each conformance requirement is given a unique ID comprised of a letter (M – MUST; S – SHOULD; O – OPTIONAL), a requirements group number, and a unique ID within that clause. ~~(§9 is the first clause that contains conformance requirements, so that clause has a requirements group number of 1. Each subsequent clause that contains conformance requirements has the next requirements group number, in ascending order.)~~ Producers and consumers can use these IDs to report error conditions. If a requirement is removed from this Standard, its ID will not be reused for any newly added requirement.

~~Additionally, each table identifies who is burdened with enforcing or supporting the requirement—the producer of content for that format or the consumer of content in that format—marked with an “x” in the appropriate column. The consumer could simply be required to validate that the producer correctly enforced a requirement in the generation of an XPS Document; these cases are marked with a “v” instead of an “x”. In certain cases, a requirement only applies to certain producers or consumers; these are marked with a superscripted letter referenced at the end of the table.~~

~~Consumers must support the usage of conformance rules marked as “OPTIONAL” and “SHOULD” only for producers if the consumer accesses the referenced feature. If a consumer or producer does not access the referenced feature it must ignore the manifestation of the rule without error.~~

~~In addition to the conformance requirements identified below, producers and consumers must meet the conformance requirements described in the OPC.~~

~~Numerous entries in the Producer and Consumer columns of the tables in this annex have superscripts, the meaning of which is, as follows:~~

Superscript	Meaning
A	Only applies to producers or consumers implementing digital signature features.
C	Only applies if Core Properties are used.
D	Only applies to producers that generate document structure or consumers that use it.
E	Only applies to consumers that are also producers
F	Only necessary if the referenced feature is used.

H	Only applies to producers that generate hyperlinks or consumers that implement hyperlinks.
K	Only applies to consumers that consume the XPS Document package head-first.
P	Only applies to printing consumers.
R	Only applies to producers or consumers that use DiscardControl parts.
S	Only applies to consumers that implement selection.
T	Only applies if thumbnails are used by the consumer.
U	Only required for a producer if the part is generated or for a consumer when accessing the part.

1 **E.1 Implementation Conformance**

2 **E.1.1 MUST Conformance Requirements**

3 *Table E-1. Implementation MUST conformance requirements*

<u>ID</u>	<u>Rule</u>	<u>Reference</u>
M0.1	A conforming consumer MUST interpret and process the contents of XPS Document instances in a manner conforming to this Standard.	2.2
M0.2	A conforming consumer is NOT REQUIRED to interpret or process all of the content in an XPS Document instance.	2.2
M0.3	A conforming consumer MUST NOT instantiate an error condition in response to XPS Document content conforming to this Standard.	2.2
M0.4	When "OPTIONAL" or "RECOMMENDED" features contained within XPS Document instances are accessed by a consumer, the consumer MUST interpret and process those features in a manner conforming to this Standard.	2.2
M0.5	Any XPS Document instances a conforming producer creates MUST conform to this Standard.	2.2
M0.6	A conforming producer MUST NOT introduce any non-conforming XPS Document content when modifying an XPS Document instance	2.2
M0.7	When a conforming producer chooses to use an "OPTIONAL" or "RECOMMENDED" feature in an XPS Document instance, then the producer MUST create or modify that feature in a manner conforming to this Standard	2.2

1 **E.1.2 SHOULD Conformance Requirements**

2 *Table E-1. Implementation SHOULD conformance requirements*

<u>ID</u>	<u>Rule</u>	<u>Reference</u>
S0.1	A conformant consumer SHOULD instantiate an error condition when XPS Document content not conforming to this Standard is encountered	2.2

3 **~~E.1~~E.2 XPS Document Format**

4 **~~E.1~~E.2.1 MUST Conformance Requirements**

5 *Table E-1. XPS Document format MUST conformance requirements*

ID	Rule	Reference
M1.1	XPS Documents MUST observe all conformance requirements of the OPC Standard, except where specifically noted otherwise in this Standard.	8, 9.2
M1.2	The XPS Document format MUST use a ZIP archive for its physical model.	8.2

6 **E.2.2 SHOULD Conformance Requirements**

7 *Table E-1. XPS Document format SHOULD conformance requirements*

<u>ID</u>	<u>Rule</u>	<u>Reference</u>
S1.1	XPS Documents SHOULD observe all recommendations of the OPC Standard, except where indicated otherwise .	8

8 **~~E.2~~E.3 Parts and Relationships**

9 **~~E.2~~E.3.1 MUST Conformance Requirements**

10 *Table E-2. Parts and Relationships MUST conformance requirements*

ID	Rule	Reference
M2.1	All content to be rendered MUST be contained in the XPS Document.	9.1, 9.1.1, 9.1.5

ID	Rule	Reference
		9.1.7
M2.2	Each part contained in an XPS Document MUST use only the appropriate content type.	9.1, 9.1.7.3
M2.3	An XPS Document MUST contain exactly one FixedDocumentSequence part per fixed payload.	9.1, 9.1.2
M2.4	An XPS Document MUST contain at least one FixedDocument part per fixed payload.	9.1
M2.5	An XPS Document MUST contain at least one FixedPage part per fixed payload.	9.1
M2.6	A <Glyphs> element in FixedPage markup MUST reference a Font part that exists in the XPS Document.	9.1
M2.7	An <ImageBrush> element in FixedPage markup MUST reference an Image part that exists in the XPS Document.	9.1
M2.8	If FixedPage markup references a Remote Resource Dictionary part, it MUST be included in the XPS Document	9.1
M2.9	This requirement was removed prior to Edition 1 of this Standard.	
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.	9.1.1, 9.1.5, 9.1.7, 15.2.3
M2.11	This requirement was removed prior to Edition 1 of this Standard.	
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.	9.1.1, 9.1.7.2, 9.1.7.4
	For preview and print embedding, a producer MUST add a Restricted Font relationship to the FixedDocument part of the document containing the font.	
	If a producer embeds a font with the print and preview restriction bit set, it MUST also add a Restricted Font relationship from the FixedDocument part that includes the FixedPage referencing the font to the restricted font.	
	When editing content, producers MUST instantiate an error condition when encountering any font with the print and preview restriction bit set for which no Restricted Font relationship has been added to the FixedDocument part.	
	When invoking editing functionality, a consumer that is also a producer 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	

ID	Rule	Reference
	and display only consumers MUST consider an XPS Document valid, even if the producer failed to properly set the Restricted Font relationship.	
M2.13	Exactly one StartPart relationship is REQUIRED.	9.1.1
M2.14	The StartPart relationship MUST point from the package to the FixedDocumentSequence part that is the primary fixed payload root.	9.1.1, 9.1.1
M2.15	The order of <DocumentReference> elements in a FixedDocumentSequence part MUST be preserved by consumers that are also producers.	9.1.2
M2.16	The order of <PageContent> elements in a FixedDocument MUST be preserved by consumers that are also producers.	9.1.3
M2.17	JPEG image parts MUST contain images that conform to the JPEG specification.	9.1.5.1
M2.18	PNG image parts MUST contain images that conform to the PNG specification.	9.1.5.2
M2.19	The PNG ancillary chunk tRNS MUST be supported.	9.1.5.2
M2.20	The PNG ancillary chunk iCCP MUST be supported.	9.1.5.2
M2.21	The PNG ancillary chunk sRGB MUST be ignored.	9.1.5.2
M2.22	The PNG ancillary chunk cHRM MUST be ignored.	9.1.5.2
M2.23	The PNG ancillary chunk gAMA MUST be ignored.	9.1.5.2
M2.24	The PNG ancillary chunk sBIT MUST be ignored.	9.1.5.2
M2.25	TIFF image parts MUST contain images that conform to the TIFF specification	9.1.5.3
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.	9.1.5.3
M2.27	If a TIFF file contains multiple image file directories (IFDs), consumers MUST use only the first IFD and ignore all others.	9.1.5.3
M2.28	XPS Document consumers MUST support TIFF images using CCITT bilevel encoding.	9.1.5.3
M2.29	XPS Document consumers MUST support CMYK TIFF images.	9.1.5.3
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.	9.1.5.3
M2.31	XPS Document consumers MUST support TIFF images using LZW compression.	9.1.5.3
M2.32	XPS Document consumers MUST support TIFF images using	9.1.5.3

ID	Rule	Reference
	differencing predictors.	
M2.33	XPS Document consumers MUST support TIFF images using JPEG compression (compression mode 6 only).	9.1.5.3
M2.34	XPS Document consumers MUST support TIFF images with an embedded ICC profile.	9.1.5.3
M2.35	Windows Media Photo image files MUST conform to the Windows Media Photo specification.	9.1.5.4
M2.36	Each FixedPage part MUST NOT have more than one thumbnail part attached.	9.1.6
M2.37	Thumbnails MUST be either JPEG or PNG images	9.1.6
M2.38	If using a fragment in the FontURI 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 $n-1$ inclusive, where n is the number of font faces in the TrueType Collection.	9.1.7
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.	9.1.7, 9.1.7.1
M2.40	Producers MUST honor the licensing rights specified in OpenType fonts by following the embedding and obfuscation mechanisms described in this Standard.	9.1.7.2
M2.41	Consumers MUST be able to process XPS Documents using any combination of the embedding and obfuscation mechanisms described in this Standard (even if produced in violation of the production requirements).	9.1.7.2
M2.42	For fonts with "Restricted license embedding" licensing intent, producers MUST NOT embed the font.	9.1.7.2
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. A producer MUST NOT modify or edit the XPS Document markup or hierarchical structure starting from the <FixedDocument> element. When editing content, producers MUST NOT edit a document where the FixedDocument part has a Restricted Font relationship.	9.1.7.2, 9.1.7.4
M2.44	For fonts with "Print and preview embedding" licensing intent, producers MUST perform embedded font obfuscation.	9.1.7.2
M2.45	For fonts with "Print and preview embedding" licensing intent, consumers MUST NOT extract or permanently install the font.	9.1.7.2
M2.46	For fonts with "Editable embedding" licensing intent, producers MUST perform embedded font obfuscation.	9.1.7.2

ID	Rule	Reference
M2.47	For fonts with “Editable embedding” licensing intent, consumers MUST NOT extract or permanently install the font.	9.1.7.2
M2.48	For fonts with “No subsetting” licensing intent, producers MUST perform embedded font obfuscation.	9.1.7.2
M2.49	For fonts with “No subsetting” licensing intent, producers MUST NOT subset the font.	9.1.7.2
M2.50	For fonts with “No subsetting” licensing intent, consumers MUST NOT extract or permanently install the font.	9.1.7.2
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.	9.1.7.2
M2.52	For fonts with “Bitmap embedding only” licensing intent, consumers MUST NOT extract or permanently install the font.	9.1.7.2
M2.53	Producers and consumers MUST perform font obfuscation and de-obfuscation according to the steps described in §9.1.7.3.	9.1.7.3
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.	9.1.7.3
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. All further processing for that font MUST use the selected cmap table.	9.1.7.5
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 UnicodeString attribute to the corresponding codepoints used by the cmap before looking up glyphs.	9.1.7.5
M2.57	When processing <Glyphs> elements that reference a cmap (3,0) encoding font, consumers MUST handle the case where the UnicodeString attribute contains character codes instead of PUA codepoints by computing the correct glyph index according to the general recommendations of the OpenType specification.	9.1.7.5
M2.58	<p data-bbox="298 1558 1175 1621">This requirement was removed prior to Edition 1 of this Standard; its description is retained here for historical purposes.</p> <p data-bbox="298 1629 1094 1684">Consumers MUST process all PrintTicket parts when an XPS Document is printed.</p>	9.1.9, 9.1.9.3

ID	Rule	Reference
M2.59	<p>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. PrintTicket parts can be attached only to FixedDocumentSequence, FixedDocument, and FixedPage parts, and each of these parts MUST attach no more than one PrintTicket.</p>	9.1.9, 9.1.9.2, 9.1.9.3
M2.60	<p>This requirement was removed prior to Edition 1 of this Standard; its description is retained here for historical purposes.</p> <p>Consumers MUST process job-level, document-level and page-level settings of PrintTicket parts associated with FixedDocumentSequence parts.</p>	9.1.9.2
M2.61	<p>This requirement was removed prior to Edition 1 of this Standard; its description is retained here for historical purposes.</p> <p>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.</p>	9.1.9.2
M2.62	<p>This requirement was removed prior to Edition 1 of this Standard; its description is retained here for historical purposes.</p> <p>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.</p>	9.1.9.2
M2.63	<p>This requirement was removed prior to Edition 1 of this Standard; its description is retained here for historical purposes.</p> <p>When processing a PrintTicket, consumers MUST first remove all levels of PrintTicket content not applicable to the current element.</p>	9.1.9.3
M2.64	<p>This requirement was removed prior to Edition 1 of this Standard; its description is retained here for historical purposes.</p> <p>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.</p>	9.1.9.3
M2.65	<p>This requirement was removed prior to Edition 1 of this Standard; its description is retained here for historical purposes.</p> <p>Following validation of a PrintTicket, the printing consumer MUST properly interpret the print settings according to the rules for merging two PrintTicket parts.</p>	9.1.9.3

ID	Rule	Reference
M2.66	<p>This requirement was removed prior to Edition 1 of this Standard; its description is retained here for historical purposes.</p> <p>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.</p>	9.1.9.3
M2.67	<p>This requirement was removed prior to Edition 1 of this Standard; its description is retained here for historical purposes.</p> <p>If there is a print setting merge conflict between different PrintTicket levels, the setting from the most specific ticket MUST take precedence.</p>	1.1.1.1
M2.68	<p>Consumers MUST use semantic document structure provided in included DocumentStructure and StoryFragments parts in preference to any other analysis method of generating such structure.</p>	9.1.11
M2.69	<p>Consumers MUST support Markup Compatibility and Extensibility elements and attributes in DocumentStructure, FixedDocument, FixedDocumentSequence, FixedPage, Relationships, Remote Resource Dictionary, SignatureDefinitions, and StoryFragments parts.</p> <p>Before attempting to validate one of these parts against a schema, consumers MUST remove all Markup Compatibility and Extensibility elements and attributes, ignorable namespace declarations, and all ignored elements and attributes not defined in the expected version of XPS Document markup.</p>	9.3.1, 9.3.2
M2.70	<p>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 Standard), that declaration MUST NOT name any encoding other than UTF-8 or UTF-16.</p>	9.3.2
M2.71	<p>DTD content MUST NOT be used in the XML markup defined in this Standard, and consumers MUST instantiate an error condition when encountering treat the presence of DTD content as an error.</p>	9.3.2
M2.72	<p>XML content MUST be valid against the corresponding W3C XSD schema defined in this Standard. 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.</p>	9.3.2 ₁ 9.3.3.2.3₁ 9.3.5.1₁ 10.2.1₁ 10.3.2
M2.73	<p>XML content MUST NOT contain elements or attributes drawn from "xml" or "xsi" namespaces unless they are explicitly defined in the W3C XSD schema or by other means in the Standard.</p>	9.3.2
M2.74	<p>Properties MUST NOT be set more than once, regardless of the syntax used to specify the value. In certain cases, they can be</p>	9.3.3.2

ID	Rule	Reference
	specified using either property attributes or property elements. Consumers MUST instantiate an error condition when encountering properties that are specified in both ways as an error .	
M2.75	XPS Document markup MUST NOT use the xml:space attribute.	9.3.4
M2.76	The language of the contents of an XPS Document MUST be identified using the xml:lang 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.	9.3.5.1
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, RequiredResource, Restricted Font, StartPart, StoryFragments, and Thumbnail. Consumers that access the target part of any relationship with one of these relationship types MUST instantiate an error condition generate an error if the part is not included in the XPS Document.	9.1.1
M2.78	Consumers MUST support JPEG images that contain the EXIF-specified APP1 marker and interpret the EXIF color space correctly.	9.1.5.1
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.	9.1.5.3
M2.80	Each <DocumentReference> element in a FixedDocumentSequence part MUST reference a FixedDocument part by relative URI.	9.1.2
M2.81	Each <PageContent> element in a FixedDocument part MUST reference a FixedPage part by relative URI.	9.1.3
M2.82	<ImageBrush> and <Glyphs> elements MUST reference Image and Font parts by relative URI.	9.1.4
M2.83	If the ExtraSamples tag value is 0, the associated alpha data in this channel MUST be ignored	9.1.5.3
M2.84	The payload for an XPS Document may include additional parts not defined by this Standard. Consumers MUST ignore parts in valid XPS Documents that they do not understand.	9.1
M2.85	Consumers MUST ensure that they can distinguish between the uses of those markers listed in Table 9-3 and other data that is recorded using the same markers.	9.1.5.1
M2.86	Signature definitions MUST conform to the Signature Definitions schema as defined in §A.1.	9.1.10
M2.87	The order of child property elements is significant: they MUST occur before any contents of the parent element and they MUST	9.3.3.2.3

ID	Rule	Reference
	appear in the sequence specified in the schema	
M2.88	xml:lang is REQUIRED for <FixedPage> elements.	9.3.5.1
M2.89	xml:lang MUST NOT be used on any other fixed page markup element [than <FixedPage>, <Canvas>, <Path>, or <Glyphs>].	9.3.5.1
M2.90	xml:lang is REQUIRED for the <DocumentOutline> element for document structure.	9.3.5.1

1 **E.2.2E.3.2 SHOULD Conformance Requirements**

2 *Table E-3. Parts and Relationships SHOULD conformance requirements*

ID	Rule	Reference
S2.1	It is RECOMMENDED that there be exactly one Required Resource relationship from the FixedPage part for each resource referenced from the markup.	9.1.1
S2.2	This requirement was removed prior to Edition 1 of this Standard; its description is retained here for historical purposes. Each <DocumentReference> element in a FixedDocumentSequence part SHOULD reference a FixedDocument part by relative URI.	
S2.3	This requirement was removed prior to Edition 1 of this Standard; its description is retained here for historical purposes. Each <PageContent> element in a FixedDocument part SHOULD reference a FixedPage part by relative URI.	
S2.4	This requirement was removed prior to Edition 1 of this Standard; its description is retained here for historical purposes. <ImageBrush> and <Glyphs> elements SHOULD reference Image and Font parts by relative URI.	
S2.5	This requirement was removed prior to Edition 1 of this Standard; its description is retained here for historical purposes. (See [M8.41] and [M8.42].) Color profiles embedded in image files SHOULD be used if present and compatible with this Standard.	
S2.6	It is RECOMMENDED that JPEG image part names end with the extension “.jpg”.	9.1.5.1
S2.7	The use of CMYK JPEG images is NOT RECOMMENDED. TIFF or Windows Media Photo images SHOULD be used	9.1.5.1, 15.3.4.3

ID	Rule	Reference
	instead to represent CMYK images.	, 15.3.7
S2.8	It is RECOMMENDED that PNG image part names end with the extension “.png”.	9.1.5.2
S2.9	It is RECOMMENDED that TIFF image part names end with the extension “.tif”.	9.1.5.3
S2.10	Consumers SHOULD ignore unsupported TIFF tags (those not described in Table 9–5 and §9.1.5.3). Producers SHOULD NOT include unsupported tags.	9.1.5.3
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.	9.1.5.3
S2.12	It is RECOMMENDED that Windows Media Photo images end with the extension “.wdp”.	9.1.5.4
S2.13	It is RECOMMENDED that if thumbnails are used for pages, a thumbnail SHOULD be included for every page in the document.	9.1.6
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.	9.1.6
S2.15	Producers SHOULD use Unicode-encoded fonts.	9.1.7, 9.1.7.5
S2.16	For fonts with “Installable embedding” licensing intent, producers SHOULD perform embedded font obfuscation.	9.1.7.2
S2.17	For fonts with “Installable embedding” licensing intent, consumers SHOULD NOT extract or permanently install the font.	9.1.7.2
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 AutomationProperties.Name attribute of the <Path> element.	9.1.7.2
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.	9.1.7.3
S2.20	It is RECOMMENDED that the extension of an obfuscated Font part name be “.odttf” for TrueType fonts and “.odttc”	9.1.7.3

ID	Rule	Reference
	for TrueType collections.	
S2.21	<p data-bbox="289 401 1036 495">This requirement was removed prior to Edition 1 of this Standard; its description is retained here for historical purposes.</p> <p data-bbox="289 506 1036 569">Producers SHOULD include only PrintTicket settings that support portability of the XPS Document.</p>	9.1.9.1
S2.22	<p data-bbox="289 583 1036 678">This requirement was removed prior to Edition 1 of this Standard; its description is retained here for historical purposes.</p> <p data-bbox="289 688 1078 779">Producers SHOULD only attach PrintTicket parts containing only document-level and page-level settings with FixedDocument parts.</p>	9.1.9.2
S2.23	<p data-bbox="289 793 1036 888">This requirement was removed prior to Edition 1 of this Standard; its description is retained here for historical purposes.</p> <p data-bbox="289 898 1078 961">Producers SHOULD only attach PrintTicket parts containing only page-level settings with FixedPage parts.</p>	9.1.9.2
S2.24	<p data-bbox="289 976 1078 1136">The FixedDocumentSequence part SHOULD follow the part name recommendation <code>"/<FixedDocSeq>.fdseq"</code> where <code><FixedDocSeq></code> is the name of the FixedDocumentSequence. The FixedDocumentSequence SHOULD use the extension <code>".fdseq"</code>.</p>	9.2
S2.25	<p data-bbox="289 1150 1078 1373">A FixedDocument part SHOULD follow the part name recommendation <code>"/Documents/<n>/<FixedDocument>.fdoc"</code> where <code><n></code> is a numeral that indicates the ordinal position of the fixed document in the fixed document sequence and <code><FixedDocument></code> is the name of the fixed document. FixedDocument parts SHOULD use the extension <code>".fdoc"</code>.</p>	9.2
S2.26	<p data-bbox="289 1388 1078 1547">A FixedPage part SHOULD follow the part name recommendation <code>"/Documents/<n>/Pages/<m>.fpage"</code> where <code><n></code> represents the document that includes this page and <code><m></code> is the page number. FixedPage parts SHOULD use the extension <code>".fpage"</code>.</p>	9.2
S2.27	<p data-bbox="289 1562 1078 1722">A resource that is specific to a particular document SHOULD follow the part name recommendation <code>"/Documents/<n>/Resources/<Resource>"</code> where <code><n></code> is the document that uses the resource and <code><Resource></code> is the segments identifying the particular resource.</p> <p data-bbox="289 1732 1078 1822">A resource that is shared across multiple documents SHOULD follow the part name recommendation <code>"/Resources/<Resource>"</code>.</p> <p data-bbox="289 1833 1078 1894">A Font resource SHOULD use <code>"Fonts/<FontName>.<FontExt>"</code> for its <code><Resource></code> value,</p>	9.2

ID	Rule	Reference
	<p>where <i><FontExt></i> SHOULD be either ".ttf" or ".odttf" for non-obfuscated and obfuscated fonts respectively.</p> <p>An Image resource SHOULD use "Images/<i><ImageName></i>.<i><ImageExt></i>" for its <i><Resource></i> value, where <i><ImageExt></i> is the correct extension for the image type.</p> <p>A Remote Resource Dictionary resource SHOULD use "Dictionaries/<i><DictName></i>.dict" for its <i><Resource></i> value. A Remote Resource Dictionary SHOULD use ".dict" as its extension.</p> <p><i><FontName></i>, <i><ImageName></i>, and <i><DictName></i> SHOULD be a string representation of a GUID value if the resource is a shared resource.</p>	
S2.28	<p>A DocumentStructure part SHOULD follow the part name recommendation "/Documents/<i><n></i>/Structure/<i><DocStruct></i>.struct" where <i><n></i> is the fixed document that the document structure applies to. DocumentStructure parts SHOULD use the extension ".struct".</p>	9.2
S2.29	<p>A StoryFragments part SHOULD follow the part name recommendation "/Documents/<i><n></i>/Structure/Fragments/<i><m></i>.frag" where <i><n></i> is the fixed document that contains the story fragments and <i><m></i> is the page number the StoryFragments part applies to. StoryFragments parts SHOULD use the extension ".frag".</p>	9.2
S2.30	<p>ICC profile parts SHOULD follow the part name recommendation "/Documents/<i><n></i>/Metadata/<i><ProfileName></i>.<i><ProfileExt></i>" where <i><n></i> is the fixed document that contains the profile.</p> <p>ICC profile parts shared across multiple documents SHOULD follow the part name recommendation "/Metadata/<i><ProfileName></i>.<i><ProfileExt></i>". In this case, <i><ProfileName></i> SHOULD be a string representation of a GUID value.</p> <p>The <i><ProfileExt></i> SHOULD be appropriate to the color profile type, such as ".icm".</p>	9.2
S2.31	<p>Thumbnail parts SHOULD follow the part name recommendation "/Documents/<i><n></i>/Metadata/<i><ThumbName></i>.<i><ThumbExt></i>" where <i><n></i> is the fixed document that contains the thumbnail.</p> <p>If the Thumbnail part represents the package as a whole, it SHOULD follow the part name recommendation "/Metadata/<i><ThumbName></i>.<i><ThumbExt></i>". In this case,</p>	9.2

ID	Rule	Reference
	<p><code><ThumbName></code> SHOULD be a string representation of a GUID value.</p> <p>The <code><ThumbExt></code> SHOULD be appropriate to the image type, either <code>“.png”</code> or <code>“.jpg”</code>.</p>	
S2.32	<p>PrintTicket part names associated with the entire job SHOULD be associated via relationship with the FixedDocumentSequence part and contain two segments, using <code>"/Metadata/"</code> as the first segment.follow the part name recommendation <code>"/Metadata/<PrintTicketName>.xml"</code>.</p> <p>PrintTicket parts associated with a particular fixed document or fixed page SHOULD contain four segments, using <code>"/Documents/n/Metadata/"</code> as the first three segmentsfollow the part name recommendation <code>"/Documents/<n>/Metadata/<PrintTicketName>.xml"</code>, where <code><n></code> is the fixed document that uses these parts.</p> <p>PrintTicket parts based on XML SHOULD use the extension <code>“.xml”</code>.</p>	9.2
S2.33	<p>The names of any non-standard parts that are associated with a particular fixed document SHOULD follow the part name recommendation <code>"/Documents/<n>/Other/<PartName>"</code>, where <code><n></code> is the fixed document to which the part belongs.</p>	9.2
S2.34	<p>Consumers SHOULD support JPEG images that contain JFIF-specified APP0 and ICC-specified APP2 markers.</p>	9.1.5.1
S2.35	<p>If the referenced font part is a TrueType Collection, then if the fragment portion of the URI is not recognised as a valid integer, consumers SHOULD instantiate generate an error condition.</p>	9.1.7
S2.36	<p>If the consumer understands the content of the PrintTicket, then the PrintTicket part SHOULD be processed when the XPS Document is printed.</p>	9.1.9
S2.37	<p>For images that have a constant opacity, producers SHOULD NOT use the image format alpha channel; the Opacity attribute in the <code><ImageBrush></code> element SHOULD be used instead.</p>	9.1.5

1 **E.3.3 OPTIONAL Conformance Requirements**2 *Table E-4. Parts and Relationships OPTIONAL conformance requirements*

ID	Rule	Reference
O2.1	Thumbnail parts MAY be included in an XPS Document	9.1
O2.2	PrintTicket parts MAY be included in an XPS Document.	9.1
O2.3	ICC Profile parts MAY be included in an XPS Document.	9.1
O2.4	DocumentStructure parts MAY be included in an XPS Document.	9.1
O2.5	StoryFragments parts MAY be included in an XPS Document.	9.1
O2.6	SignatureDefinitions parts MAY be included in an XPS Document.	9.1, 9.1.10
O2.7	DiscardControl parts MAY be included in an XPS Document.	9.1
O2.8	A Core Properties relationship MAY be included in an XPS Document, from the package to the Core Properties part.	9.1.1
O2.9	A Digital Signatures Origin relationship MAY be included in an XPS Document, from the package to the Digital Signature Origin part.	9.1.1
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.	9.1.1
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.	9.1.1
O2.12	Digital Signature Definitions parts MAY be included in an XPS Document, from a FixedDocument part to the Digital Signature Definitions part.	9.1.1
O2.13	DiscardControl relationships MAY be included in an XPS Document, from the package to a DiscardControl part.	9.1.1
O2.14	DocumentStructure relationships MAY be included in an XPS Document, from a FixedDocument part to the DocumentStructure part.	9.1.1
O2.15	PrintTicket relationships MAY be included in an XPS Document, from a FixedDocumentSequence, FixedDocument, or FixedPage part to a PrintTicket part.	9.1.1
O2.16	StoryFragments relationships MAY be included in an XPS Document, from a FixedPage part to a StoryFragments part.	9.1.1
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.	9.1.1
O2.18	Color Profiles MAY be embedded in image files.	9.1.5
O2.19	Thumbnail images MAY be attached to a FixedPage part using a Thumbnail relationship.	9.1.6
O2.20	Fonts MAY be subsetted based on glyph usage.	9.1.7.1
O2.21	Producers MAY use a 128-bit random number instead of a true GUID for an obfuscated font name.	9.1.7.3

O2.22	An obfuscated Font part MAY have an arbitrary extension.	9.1.7.3
O2.23	Producers MAY add digital signature requests and instructions to an XPS Document in the form of signature definitions.	9.1.10
O2.24	A producer MAY sign against an existing signature definition to provide additional signature information.	9.1.10
O2.25	A recipient of an XPS Document MAY also sign it against a signature definition.	9.1.10
O2.26	This requirement was removed prior to Edition 1 of this Standard.	
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.	9.1.11
O2.28	A resource that is intended to be used across multiple fixed documents MAY be named according to the guidelines for shared resources.	9.2
O2.29	Producers MAY include Markup Compatibility and Extensibility elements and attributes in DocumentStructure, FixedDocument, FixedDocumentSequence, FixedPage, Relationships, Remote Resource Dictionary, SignatureDefinitions, and StoryFragments parts.	9.3.1
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).	9.3.4
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.	9.3.4
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 behavior specified in the XML Schema Standard.	9.3.4
O2.33	xml:lang MAY be used with <Canvas>, <Path>, and <Glyphs> elements.	9.3.5.1
O2.34	xml:lang is OPTIONAL for the <OutlineEntry> element.	9.3.5.1

1 ~~E.3~~**E.4 Documents**

2 ~~E.3.1~~**E.4.1 MUST Conformance Requirements**

3 *Table E-5. Document MUST conformance requirements*

ID	Rule	Reference
M3.1	The order of <DocumentReference> elements MUST match the order of the documents in the fixed document sequence.	10.1
M3.2	The Source attribute of the <DocumentReference> element MUST specify a FixedDocument part within the XPS Document.	10.1.1
M3.3	Producers MUST NOT produce a document with multiple <DocumentReference> elements that reference the same fixed document.	10.1.1

M3.4	The order of <PageContent> elements MUST match the order of the pages in the document.	10.2
M3.5	The Source attribute of the <PageContent> element MUST specify a FixedPage part within the XPS Document.	10.2.1
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.	10.2.1
M3.7	If the attribute is specified, the BleedBox BleedOriginX value MUST be less than or equal to 0.	10.3.1
M3.8	If the attribute is specified, the BleedBox BleedOriginY value MUST be less than or equal to 0.	10.3.1
M3.9	If the attribute is specified, the BleedBox BleedWidth value MUST be greater than or equal to the Width attribute value plus the absolute value of the BleedBox BleedOriginX value.	10.3.1
M3.10	If the attribute is specified, the BleedBox BleedHeight value MUST be greater than or equal to the fixed page Height value plus the absolute value of the BleedBox BleedOriginY value.	10.3.1
M3.11	If the attribute is specified, the ContentBox ContentOriginX value MUST be greater than or equal to 0 and less than the fixed page Width attribute value	10.3.2
M3.12	If the attribute is specified, the ContentBox ContentOriginY value MUST be greater than or equal to 0 and less than the fixed page Height attribute value.	10.3.2
M3.13	If the attribute is specified, the ContentBox ContentWidth value MUST be less than or equal to the difference between the fixed page Width attribute value and the ContentBox ContentOriginX value.	10.3.2
M3.14	If the attribute is specified, the ContentBox ContentHeight value MUST be less than or equal to the difference between the fixed page Height attribute value and the ContentBox ContentOriginY value.	10.3.2
M3.15	This requirement was removed prior to Edition 1 of this Standard; its description is retained here for historical purposes. When rendering a fixed page for printing, consumers MUST be aware of the interaction between the fixed page markup and the PrintTicket settings.	10.3.4
M3.16	This requirement was removed prior to Edition 1 of this Standard; its description is retained here for historical purposes. 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.	10.3.4.1
M3.17	This requirement was removed prior to Edition 1 of this Standard; its description is retained here for historical purposes. 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.	10.3.4.2
M3.18	This requirement was removed prior to Edition 1 of this Standard; its description is retained here for historical purposes.	10.3.4.3

	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.	
M3.19	This requirement was removed prior to Edition 1 of this Standard; its description is retained here for historical purposes.	10.3.4.4
	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.	
M3.20	The x:Key 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.	10.4
M3.21	The <PageContent> element has one allowable child element, <PageContent.LinkTargets>, and it MUST NOT contain more than a single child element.	10.2.1
M3.22	The fixed page MUST specify a height, width, and default language.	10.3

1 ~~E.3.2~~**E.4.2 SHOULD Conformance Requirements**

2 *Table E-6. Document SHOULD conformance requirements*

ID	Rule	Reference
S3.1	Specifying a ContentBox attribute for the <FixedPage> element is RECOMMENDED.	10.3, 10.3.2
S3.2	Invalid bleed box specifications SHOULD be ignored in favor of the default bleed box.	10.3.1
S3.3	Invalid content box specifications SHOULD be ignored in favor of the default content box NOT be rendered and SHOULD generate an error.	10.3.2 10.3.1
S3.4	This requirement was removed prior to Edition 1 of this Standard; its description is retained here for historical purposes. 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.	10.3.4
S3.5	By default, consumers SHOULD clip to the FixedPage Width and Height.	10.3.4

E.4.3 OPTIONAL Conformance Requirements

Table E-7. Document OPTIONAL conformance requirements

ID	Rule	Reference
O3.1	The positioning, scaling, orientation, and clipping of FixedPage content when mapping to physical media MAY be controlled by settings provided in the PrintTicket.	10.3.4
O3.2	Consumers MAY provide implementation-defined mechanisms to select alternative clipping strategies.	10.3.4

~~E.4~~E.5 Graphics

~~E.4.1~~E.5.1 MUST Conformance Requirements

Table E-8. Graphics MUST conformance requirements

ID	Rule	Reference
M4.1	The x:Key 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.	11.1
M4.2	The x:Key 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.	11.2.1.1
M4.3	A <PathGeometry> element contains a set of path figures specified either with the Figures attribute or with a child <PathFigure> element. Producers MUST NOT specify the path figures of a geometry with both the Figures attribute and a child <PathFigure> element.	11.2.1.1
M4.4	The <PathGeometry> element's Figures attribute can be used to describe the path figures the geometry contains using abbreviated syntax with the exception that the FillRule command MUST NOT be used.	11.1.3
M4.5	The x or y radius in the Size attribute MUST NOT be negative.	11.2.2.2.1
M4.6	This [FillRule] command MUST appear only as the first command in the abbreviated geometry syntax.	11.2.3
M4.7	This [FillRule] command MUST NOT be specified in the value of the Figures attribute of the <PathGeometry> element.	11.2.3
M4.8	The first figure in a geometry MUST begin with a Move command.	11.2.3

1 **~~E-4.2~~E.5.2 SHOULD Conformance Requirements**2 *Table E-9. Graphics SHOULD conformance requirements*

ID	Rule	Reference
S4.1	Line segments and curve segments SHOULD NOT be specified as zero-length.	11.2.2.1

3 **~~E-4.3~~E.5.3 OPTIONAL Conformance Requirements**4 *Table E-10. Graphics OPTIONAL conformance requirements*

ID	Rule	Reference
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 SnapsToDevicePixels attribute is specified as true.	11.1, 19.39
O4.2	A path geometry MAY define the fill algorithm to be used on the component path figures.	11.2
O4.3	Abbreviated geometry syntax MAY be used to specify a geometry of one or more figures comprised of multiple segments.	11.2.3
O4.4	If entering more than one drawing command of the same type sequentially, the duplicate command entry MAY be omitted.	11.2.3
O4.5	Every geometry MAY specify one or more figures, and MAY be preceded by a FillRule command where allowed.	11.2.3
O4.6	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 end point of the previous figure.	11.2.3

5 **~~E-5~~E.6 Text**6 **~~E-5.1~~E.6.1 MUST Conformance Requirements**7 *Table E-11. Text MUST conformance requirements*

ID	Rule	Reference
M5.1	If the CaretStops attribute is missing from the <Glyphs> element, a consumer MUST interpret the text as having a caret stop between each Unicode UTF-16 code unit and at the beginning and end of the text.	12.1
M5.2	If the UnicodeString attribute of the <Glyphs> element specifies an empty string (“” or “{ }”) and the Indices attribute is not specified or is empty, the a	12.1, 12.1.4

	consumer MUST instantiate an error condition generate an error .	
M5.3	The x:Key attribute of the <Glyphs> element MUST be present when the element is defined in a resource dictionary. It MUST NOT be specified outside a resource dictionary.	12.1
M5.4	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.	12.1.3
M5.5	If there is not a one-to-one mapping between code units in the UnicodeString attribute and the glyph indices, the GlyphIndex value in the Indices attribute MUST be specified.	12.1.3
M5.6	The AdvanceWidth of the Indices attribute 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.	12.1.3
M5.7	A UnicodeString attribute value that begins with an open brace ("{") MUST be escaped with a prefix of "{". If a UnicodeString attribute value starts with "{", consumers MUST ignore those first two characters in processing the UnicodeString and in calculating index positions for the characters of the UnicodeString.	12.1.4
M5.8	This requirement was removed prior to Edition 1 of this Standard.	
M5.9	If the UnicodeString 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 the Indices attribute, the consumer MUST display the .notdef glyph	12.1.4
M5.10	In the absence of entries in the Indices attribute to override the Unicode code units in the UnicodeString attribute value, consumers MUST treat Unicode control marks in the UnicodeString attribute like ordinary characters and render the glyphs to which the Unicode control marks are mapped in the CMAP table.	12.1.4
M5.11	Because advance-widths, glyph indices, and caret-stops are associated with the generated Unicode string, consumers MUST NOT normalize the UnicodeString attribute value to produce an internal representation.	12.1.4
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.	12.1.5
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 IsSideways is false) or the vertical metrics font table (if IsSideways is true) of the font MUST be increased by 2% of the em size.	12.1.5
M5.14	Producers MUST lay out algorithmically italicized glyphs using exactly the same advance widths as when not algorithmically italicized.	12.1.5
M5.15	Producers MUST NOT specify text that is both right-to-left (BidiLevel attribute set to an odd value value of 1) and vertical (IsSideways attribute set to true).	12.1.6.2
M5.16	If a consumer does not understand the specified device font name, it MUST render the embedded version of the font.	12.1.7

M5.17	When rendering a printer device font, consumers MUST use the UnicodeString attribute and ignore the glyph index components of the Indices attribute.	12.1.7
M5.18	When rendering a printer device font, consumers MUST still honor the advance width and x,y offset values present in the Indices attribute.	12.1.7
M5.19	For producers, a <Glyphs> element with a specified device font name MUST have exactly one Indices glyph per character in the UnicodeString attribute. Its Indices attribute MUST NOT include any cluster specifications. If the Indices attribute includes a cluster mapping, the consumer MUST NOT use the device font name and MUST render the embedded version of the font.	12.1.7
M5.20	For producers of a <Glyphs> element with a specified device font name, each of the Indices glyphs MUST include a specified advance width and MUST include specified x and y offset values if they are non-zero.	12.1.7
M5.21	This requirement was removed prior to Edition 1 of this Standard.	
M5.22	If there are insufficient flags in the CaretStops attribute value to correspond to all the UTF-16 code units in the UnicodeString attribute value, all remaining UTF-16 code units in the Unicode string MUST be considered valid caret stops.	12.1.9
M5.23	If the Indices attribute is specified, the values provided MUST be used in preference to values determined from the UnicodeString attribute alone.	12.1.3
M5.24	If the Indices attribute specifies a GlyphIndex that does not exist in the font, the consumer MUST instantiate an error condition generate an error .	12.1.3
M5.25	The Indices attribute MUST adhere to the glyph specification syntax.	12.1.3
M5.26	AdvanceWidth's advance MUST be 0 or greater.	12.1.3
M5.27	For larger blocks of text, the producer MAY specify the xml:lang attribute on the <Canvas> element.	12.1.8

1 ~~E.5.2~~E.6.2 SHOULD Conformance Requirements

2 *Table E-12. Text SHOULD conformance requirements*

ID	Rule	Reference
S5.1	The value of the CaretStops 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.	12.1
S5.2	If producers include control marks in the Unicode string, they SHOULD include an Indices attribute to specify glyph indices and/or character-to-glyph mapping information for the control marks.	12.1.4
S5.3	If alternate vertical character representations are available in the font, the producer SHOULD use those in preference to the IsSideways attribute and provide their glyph indices in the Indices attribute.	12.1.6
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.	12.1.7
S5.5	Specifying a UnicodeString for <Glyphs> elements is RECOMMENDED, as it supports searching, selection, and accessibility.	12.1.4

1 ~~E.5.3~~**E.6.3** **OPTIONAL Conformance Requirements**2 *Table E-13. Text OPTIONAL conformance requirements*

ID	Rule	Reference
05.1	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 definition within the Unicode Standard.	12.1.4
05.2	Producers MAY choose to generate UnicodeString attribute values that are not normalized by any Unicode-defined algorithm.	12.1.4
05.3	Consumers that understand the device font name MAY ignore the embedded font and use the device-resident version.	12.1.7
05.4	Glyph indices MAY be omitted from markup where there is a one-to-one mapping between the positions of Unicode scalar values in the UnicodeString 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.	12.1.10.1
05.5	Glyph advance widths MAY be omitted from markup where the advance width desired is specified in the font tables, once adjusted for algorithmic emboldening.	12.1.10.2
05.6	Glyph horizontal and vertical offsets MAY be omitted from markup where the offset is 0.0.	12.1.10.2
05.7	The <Glyphs> element MAY have an Indices attribute.	12.1.3
05.8	The glyph specifications within the Indices attribute are OPTIONAL.	12.1.3
05.9	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.	12.1.3
05.10	The Indices attribute MAY include glyph placement information.	12.1.3
05.11	The GlyphIndex entry MAY be empty.	12.1.3
05.12	A cluster map specification MAY precede the glyph specification for the first glyph of the cluster.	12.1.3.1
05.13	The language defaults to the value specified for the xml:lang attribute of the <FixedPage> element but MAY be overridden by an xml:lang attribute on a <Glyphs> element.	12.1.8

1 **E.6E.7** Brushes2 **E.6.1E.7.1** MUST Conformance Requirements3 *Table E-14. Brushes MUST conformance requirements*

ID	Rule	Reference
M6.1	The x:Key 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.	13.1
M6.2	The x:Key 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.	13.2
M6.3	An <ImageBrush> element MUST reference a JPEG, PNG, TIFF, or Windows Media Photo Image part within the XPS Document package.	13.2
M6.4	The x:Key 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.	13.3
M6.5	The x:Key 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.	13.5
M6.6	The x:Key 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.	13.6
M6.7	ViewboxUnits specifies the unit type for the Viewbox attribute. MUST have the value "Absolute".	13.4
M6.8	ViewportUnits specifies the unit type for the Viewport attribute. MUST have the value "Absolute".	13.4

4 **E.7E.8** Common Properties5 **E.7.1E.8.1** MUST Conformance Requirements6 *Table E-15. Common properties MUST conformance requirements*

ID	Rule	Reference
M7.1	Individual resource values MUST be specified within a resource dictionary.	14.2
M7.2	Namespace prefixes in resource definitions MUST apply in the context of the definition, rather than in the context of the resource reference.	14.2.3
M7.3	An xml:lang attribute within a resource definition MUST be interpreted in the context of the resource reference, not the resource definition.	14.2.3

M7.4	A remote resource dictionary MUST follow the requirements that apply to inline resource dictionaries.	14.2.3.1
M7.5	A remote resource dictionary MUST NOT contain any resource definition children that reference another remote resource dictionary.	14.2.3.1
M7.6	A <ResourceDictionary> element that specifies a remote resource dictionary in its Source attribute MUST NOT contain any resource definition children.	14.2.3.1
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.	14.2.3.1
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	14.2.5
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.	14.2.5
M7.10	If a resource dictionary contains Markup Compatibility and Extensibility elements and attributes, the processing of the Markup Compatibility and Extensibility markup MUST occur in the context of the definition of the resource dictionary, not in the context of resource references.	14.2.6
M7.11	The x:Key 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.	14.4.1
M7.12	The Opacity property attribute value MUST fall within the 0 (fully transparent) to 1 (fully opaque) range, inclusive.	14.1
M7.13	The <Canvas.Resources> or <FixedPage.Resources> property elements MUST precede any property elements of the <Canvas> or <FixedPage> elements.	14.2
M7.14	The <Canvas.Resources> or <FixedPage.Resources> property elements MUST precede any path, glyphs, or canvas children of the <Canvas> or <FixedPage> elements.	14.2
M7.15	<FixedPage.Resources> and <Canvas.Resources> elements that include a remote resource dictionary MUST include exactly one <ResourceDictionary> element.	14.2.3.1
M7.16	The value of the x:Key attribute MUST be unique within the resource dictionary.	14.2.5

1 ~~E.7.2~~**E.8.2** **OPTIONAL Conformance Requirements**

2 *Table E-16. Common properties OPTIONAL conformance requirements*

ID	Rule	Reference
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.	14.2

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.	14.2.3
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.	14.2.3
O7.4	This requirement was removed prior to Edition 1 of this Standard.	
O7.5	The resource dictionary of a <Canvas> element MAY re-use (and thus override within the scope of the re-use) an x:Key value defined in the resource dictionary of a parent or ancestor <Canvas> or <FixedPage> element.	14.2.5
O7.6	A resource definition MAY reference a previously defined resource with the same name that is defined in an ancestor resource dictionary.	14.2.5
O7.7	An abbreviated matrix transformation syntax MAY be used to specify a RenderTransform or Transform attribute value.	14.4

1 ~~E.8~~E.9 Color

2 ~~E.8.1~~E.9.1 MUST Conformance Requirements

3 *Table E-17. Color MUST conformance requirements*

ID	Rule	Reference
M8.1	Consumers MUST support alpha and gradient blending in sRGB. Consumers MUST support sRGB colors (8-bit-per-channel) in vector data, with and without alpha.	15.1
M8.2	Consumers MUST support sRGB colors in image data, using the JPEG, PNG, TIFF, or Windows Media Photo image formats.	15.1
M8.3	Consumers MUST support scRGB color specification in vector data, with and without alpha.	15.1
M8.4	Consumers MUST support scRGB colors in image data, using the Windows Media Photo image format.	15.1
M8.5	Consumers MUST support CMYK colors in vector data.	15.1
M8.6	Consumers MUST support CMYK colors in image data, using the TIFF or Windows Media Photo image formats.	15.1
M8.7	Consumers MUST support N-Channel colors in vector data.	15.1
M8.8	Consumers MUST support N-Channel colors in image data, using the Windows Media Photo image format.	15.1
M8.9	This requirement was removed prior to Edition 1 of this Standard; its description is retained here for historical purposes. It was subsumed by [M8.53]. Consumers MUST support profiles as specified in the ICC specification.	
M8.10	This requirement was removed prior to Edition 1 of this Standard;	

its description is retained here for historical purposes.

Consumers MUST support profiles compliant with ICC.1:2001-04 with a Windows Color System (WCS) profile embedded as a private tag.

M8.11	This requirement was removed prior to Edition 1 of this Standard, and replaced with S8.21; its description is retained here for historical purposes.	15.1.7
	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 or gradient blending.	
M8.12	XPS producers and consumers MUST provide color management using ICC profiles conforming to the requirements of the ICC Color Profile specification, ICC.1:2001-04, for color spaces other than sRGB and scRGB.	15.1.8, 15.6
M8.13	All ICC profiles used in XPS Documents MUST either be an Input profile, an Output profile, a Monitor (RGB) profile, or a ColorSpace Conversion profile, or a Named Color profile .	15.1.8
M8.14	Real numbers specified for color channel values of scRGB and ContextColor colors MUST NOT use exponent forms of numbers.	15.2
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.	15.2.2
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.	15.2.3
M8.17	For N-Channel colors, the context color MUST specify the number of channel float values equal to the number of channels in the profile.	15.2.3, 15.2.5
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.	15.2.5
M8.19	In the case of a named color with an associated tint LUT implemented in an ICC monochrome profile, the profile MUST include an AtoB1Tag (relative colorimetric rendering intent), mapping the named color tint values to valid PCS values. The profile for a named color MUST have 3, 4, 5, 6, 7 or 8 channels (and an nCLR signature, where n is the number of channels), mapping to a valid PCS.	15.2.6
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.	15.2.6
M8.21	This requirement was removed prior to Edition 1 of this Standard. Consumers that do not understand a named color MUST compute a color approximation through ICM-compliant color	15.2.5

	management functions using the specified profile.	
M8.22	This requirement was removed prior to Edition 1 of this Standard; its description is retained here for historical purposes. The color name specified by the DocumentImpositionColor PrintTicket setting MUST be matched only to profiles containing exactly one non-zero-length colorant name in the profile's colorantTable.	15.4
M8.23	This requirement was removed prior to Edition 1 of this Standard; its description is retained here for historical purposes. 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.	15.4
M8.24	This requirement was removed prior to Edition 1 of this Standard; its description is retained here for historical purposes. 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.	15.4
M8.25	For gradients, the specified blending color space in the blending color space 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 blending color space PageBlendColorSpace-PrintTicket setting, the color interpolation mode of the gradient brush MUST be used instead.	15.5
M8.26	This requirement was removed prior to Edition 1 of this Standard.	
M8.27	This requirement was removed prior to Edition 1 of this Standard; its description is retained here for historical purposes. 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.	15.6
M8.28	This requirement was removed prior to Edition 1 of this Standard; its description is retained here for historical purposes. If the PageBlendColorSpace PrintTicket setting is set to ICCProfile, the profile MUST be an output profile, otherwise it MUST be ignored.	15.6
M8.29	This requirement was removed prior to Edition 1 of this Standard; its description is retained here for historical purposes. Elements using the named color identified by the DocumentImpositionColor PrintTicket setting MUST appear on all color separations.	15.6
M8.30	If no usable profile is present with an image, then a consumer MUST apply a color rule based on the pixel format. Each pixel format is interpreted to be the encoding of a particular color	9.1.5.1, 9.1.5.2, 9.1.5.3, 9.1.5.4; 15.3.8Error!

	space as shown in Table 15-2. 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 §15.3.8 Error! Reference source not found.	Reference source not found.
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. Before the value is used as input for an ICC profile color transformation, it MUST be linearly scaled (with specified rounding/clipping) 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 before input to the profile.	15.2.3, 15.2.4, 15.2.5, 15.2.6
M8.32	For 1-channel color, i.e., monochrome, use a monochrome input (or output) profile. The profile MUST include the ICC-optional AToB1Tag (relative colorimetric intent) if the single color is chromatic (not neutral).	15.2.5
M8.33	A named color with an associated tint LUT MUST be implemented in an XPS Document using an associated ICC monochrome profile.	15.2.6
M8.34	The ICC profile for a named color with an associated tint LUT MUST MUST contain the tint LUT for a single named color.	15.2.6
M8.35	The ICC profile for a named color with an associated tint LUT MUST be an ICC monochrome input or output profile.	15.2.6
M8.36	In the case of a named color with an associated tint LUT the ASCII prefix-root-suffix name of the named color MUST be encoded into the profileDescriptionTag of the ICC profile.	15.2.6
M8.37	In the case of a named color with an associated tint LUT the profile header color space signature MUST be 'GRAY'.	15.2.6
M8.38	Two or more named colors implemented in an XPS Document using a single associated profile MUST use an ICC Named Color type profile.	15.2.6
M8.39	An ICC Named Color type profile MUST contain the namedColor2Tag including the ASCII prefix-root-suffix name for each named color.	15.2.6
M8.40	The namedColor2Tag in a Named Color type profile MUST be populated with the ICC PCS color value for each named color.	15.2.6
M8.41	If present and usable, an associated profile MUST be used by consumers.	15.3.8
M8.42	If present and usable, a color profile embedded in an image file MUST be used by consumers when no usable associated profile is present with the image.	15.3.8
M8.43	A producer MUST associate or embed a usable color profile if the color rules of Table 15-2 do not guarantee appropriate color interpretation for an image.	15.3.8
M8.44	Profiles associated as described in Table 15-1, and determined to be usable, MUST be used by consumers.	15.2
M8.45	If no usable profile is present in a context color syntax, then a	15.2

	consumer MUST apply a color rule based on the context color syntax.	
M8.46	Single component integer default for vector data MUST be grayscale with the sRGB non-linearity, black point, and white point.	15.2
M8.47	Three component integer default for vector data MUST be sRGB.	15.2
M8.48	Three component float default for vector data MUST be scRGB.	15.2
M8.49	The specific CMYK to be used as the four component data default for vector data MUST be determined by the consumer.	15.2
M8.50	N-Channel data with $N \leq 3$ and any named color data: the data of the first channel MUST be interpreted independently as grayscale. Other channels are disregarded.	15.2
M8.51	N-Channel with $N > 4$ MUST be treated as four component data using the four component data default for vector data determined by the consumer.	15.2
M8.52	A producer MUST associate or embed a usable color profile if the color rules above do not guarantee appropriate color interpretation for the vector color content.	15.2
M8.53	XPS consumers MUST use associated and embedded ICC profiles, according to the precedence order of §15.3.8 for raster images and according to §15.2 for vector content.	15.1.8
M8.54	Implementations MUST ignore and preserve private tags that they do not understand	15.1.8
M8.55	For Named colors, the XPS context color syntax MUST specify the matching number of tint float values.	15.2.6
M8.56	Consumers MUST support grayscale colors (single channel) in vector data, with and without alpha.	15.1
M8.57	Consumers MUST support grayscale colors in image data, using the JPEG, PNG, TIFF, or HDPhoto image formats.	15.1
M8.58	Producers MUST restrict associated ICC profiles to conform to the requirements of the older ICC Color Profile specification, ICC.1:2001-04, when consumer support of the newer ISO version cannot be ascertained.	15.1.8
M8.59	If a Producer includes an image with an embedded profile conforming to the requirements of ISO 15076-1, then the Producer MUST associate an ICC profile conforming to the requirements of the older ICC Color Profile specification, ICC.1:2001-04, to have precedence over such an embedded profile, when consumer support of the newer ISO version cannot be ascertained.	15.1.8

1 **E-8.2E.9.2 SHOULD Conformance Requirements**2 *Table E-18. Color SHOULD conformance requirements*

ID	Rule	Reference
S8.1	<p>This requirement was removed prior to Edition 1 of this Standard; its description is retained here for historical purposes. It was subsumed by [M8.53] and [M8.41] through [M8.44].</p> <p>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.</p>	
S8.2	<p>If consistency of appearance of grayscale images is important, the producer SHOULD adjust the gray tone response curve of such images before adding to the XPS Document. 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.</p>	15.1.8
S8.3	<p>This requirement was removed prior to Edition 1 of this Standard; its description is retained here for historical purposes.</p> <p>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).</p>	
S8.4	<p>This requirement was removed prior to Edition 1 of this Standard; its description is retained here for historical purposes.</p> <p>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.</p>	
S8.5	<p>If the consumer does not know ALL of the colorants named in the clrt 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.</p>	15.2.6
S8.6	<p>This requirement was removed prior to Edition 1 of this Standard; its description is retained here for historical purposes.</p> <p>Support for JPEG CMYK images varies by implementation and SHOULD NOT be used in XPS Documents.</p>	
S8.7	<p>For consumers that do perform separation, the occurrence of the document registration named color in a color syntax is <i>only</i> an indicator that the tint level supplied in the syntax SHOULD be used when drawing the registration marking in each colorant separation. 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.</p>	15.4

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 Producers SHOULD create the profile for the document registration named color in such a way that it does not lay down excessive ink when printed on a device that does not perform separation.	15.4
S8.9	A page-level PrintTicket setting can be used to specify the blending color space that SHOULD be used for blending gradients and transparencies If a consumer understands the PageBlendColorSpace PrintTicket setting, it SHOULD convert all color to the specified blending color space before performing a blend operation. If a consumer understands the blending color space PrintTicket setting, it SHOULD convert all color to the specified blending color space before performing a blend operation.	15.5
S8.10	This requirement was removed prior to Edition 1 of this Standard; its description is retained here for historical purposes. 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.	15.6
S8.11	This requirement was removed prior to Edition 1 of this Standard; its description is retained here for historical purposes. 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.	15.6
S8.12	This requirement was removed prior to Edition 1 of this Standard; its description is retained here for historical purposes. 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.	15.6
S8.13	This requirement was removed prior to Edition 1 of this Standard; its description is retained here for historical purposes. The PageICMRenderingIntent PrintTicket setting SHOULD be ignored for elements using a profile that specifies the rendering intent in the profile.	15.6
S8.14	This requirement was removed prior to Edition 1 of this Standard; its description is retained here for historical purposes. It was subsumed by new rules for named colors in §15.2.3. 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).	
S8.15	This requirement was removed prior to Edition 1 of this Standard; its description is retained here for historical purposes. It was replaced by new rules M8.58 and M8.59. Producers SHOULD restrict ICC profiles to conform to the requirements of the older ICC Color Profile specification, ICC.1:2001-04, when consumer support	15.1.8

	of the newer ISO version cannot be ascertained.	
S8.16	If the XPS system environment allows the use of ICC ISO 15076-1 profiles, the optional colorantTableTag SHOULD be included in such ISO 15076-1 profiles to indicate the names and corresponding PCS values of the individual N-color colorants.	15.2.5, 15.3.5
S8.17	A profile associated or embedded with an image SHOULD be considered unusable by a consumer if 1) The profile is not compatible with the pixel format of the image, 2) The profile contains optional tags that ambiguate XPS use, and 3) The profile contains invalid tag type signatures that invalidate XPS use.	15.3.8
S8.18	A profile associated as in Table 15–1 SHOULD be considered unusable by a consumer if 1) The profile is not compatible with the context color syntax, 2) The profile contains optional tags that ambiguate XPS use , and 3) The profile contains invalid tag type signatures that invalidate XPS use.	15.2
S8.19	The specific CMYK to be used as the four-component raster data default, and the N-Channel (N=>4) default, is implementation-defined. In the absence of specific requirements the use of CGATS/SWOP TR003 2007 CMYK is recommended.	15.3.8
S8.20	In the absence of ICC rendering intents, in a typical case, with ICC profiles conforming to the ICC Color Profile specification, ICC.1:2001-04, a consumer SHOULD apply the defaults shown in Table 15.4^[rcj8].	15.6
S8.21	If a consumer recognizes that a profile given in the syntax for a page element matches the page level PrintTicket output-ready ICC profile and that the page level PrintTicket output-ready ICC profile is suitable for the output device conditions, then the consumer SHOULD elect to treat the element colors as output-ready colors and not color-manage them, unless forced to do so for transparency effects or gradient blending	15.1.7
S8.22	The name of the document registration named color is given in the profile’s profileDescriptionTag. Such a document registration named color SHOULD be unique for that use in the XPS Document instance	15.4

1 ~~E-8.3~~[E.9.3](#) **OPTIONAL Conformance Requirements**

2 *Table E–19. Color OPTIONAL conformance requirements*

ID	Rule	Reference
O8.1	Consumers are not required to handle all color spaces natively through every processing stage, but, rather, MAY convert data specified in a color space other than sRGB to sRGB at an early stage (possibly resulting in reduced fidelity).	15.1
O8.2	This requirement was removed prior to Edition 1 of this Standard; its description is retained here for historical purposes. An ICC profile MAY contain the private tag, “MS00”, which specifies an	

	embedded Windows Color System (WCS) profile.	
08.3	When a named color is used in a gradient brush or with transparency, the result produced by consumers determining the color from the ASCII color name found in the associated ICC Profile that are not named-color aware MAY differ significantly from the result produced by consumers that are named-color aware using the encoded color value of the named color from the associated ICC profile .	15.2.6
08.4	This requirement was removed prior to Edition 1 of this Standard; its description is retained here for historical purposes. A named color profile MAY be used with images for spot coloring.	
08.5	Producers MAY elect to generate content that provides registration marks for consumers that perform color separation Consumers MAY perform color separation, if desired.	15.4
08.6	Consumers MAY support alpha and gradient blending with color spaces such as sRGB or CMYK. Consumers that encounter any document using non-sRGB colors MAY process those colors using conversion to the simpler sRGB color space, resulting in deviations, especially for alpha blending.	15.5
08.7	This requirement was removed prior to Edition 1 of this Standard; its description is retained here for historical purposes. If the PageColorManagement PrintTicket setting specifies a value of Driver, the driver MAY color manage elements or convert them to different color spaces.	15.6
08.8	This requirement was removed prior to Edition 1 of this Standard; its description is retained here for historical purposes. 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.	15.6
08.9	XPS producers and consumers MAY provide color management using ICC profiles conforming to the requirements of ISO 15076-1.	15.1.8, 15.6
08.10	This requirement was removed prior to Edition 1 of this Standard; its description is retained here for historical purposes. The specific CMYK to be used as the four-component raster data default, and the N-Channel (N=>4) default, is implementation-defined. In the absence of specific requirements the use of CGATS/SWOP TR003 2007 CMYK is recommended. Alternatively, a consumer MAY choose to instantiate an error condition.	
08.11	A single named color MAY be implemented in an XPS Document using an associated ICC Named Color type profile.	15.2.6
08.12	An the case of an ICC Named Color type profile the namedColor2Tag MAY be populated with specific device color values for each named color.	15.2.6
08.13	A consumer incapable of supporting a particular ICC profile tag that is optional in both ICC and XPS MAY treat this tag as a user-defined custom tag, and therefore ignore it.	15.3.8
08.14	When no usable profile is present a consumer MAY choose to instantiate an error condition.	15.3.8
08.15	ICC profiles embedded in any image format (according to the restrictions	15.1.8

	of the image file format) with any color space.	
08.16	XPS producers MAY include ICC profiles for sRGB and scRGB color spaces.	15.1.8
08.17	An ICC profile MAY contain private tags.	15.1.8
08.18	Implementations MAY act on private tags.	15.1.8
08.19	Producers and consumers MAY support N-Channel colors in image data, using the TIFF image format.	15.1
08.20	A consumer MAY use the profile to obtain the encoded name of the named color.	15.2.6
08.21	A consumer MAY use the encoded name of a named color to lookup a device-specific color value for the named color.	15.2.6
08.22	A consumer MAY use the profile to obtain the encoded name of the named color.	15.2.6, 15.3.6
08.23	A consumer MAY use the encoded name of a named color to lookup a device-specific color value for the named color.	15.2.6, 15.3.6
08.24	A consumer MAY use the encoded name of a named color to lookup a device-specific color value for the named color	15.3.6
08.25	Consumers MAY use the ASCII name in the ICC profile or MAY compute a color approximation using a specified color value in the ICC profile; the results of these two methods MAY differ significantly.	15.3.6
08.26	A document registration named color identified in a PrintTicket MAY occur in an XPS Document using the single named color and monochrome profile with tint LUT syntax.	15.4

~~E.9~~E.10 Document Structure and Interactivity

~~E.9.1~~E.10.1 MUST Conformance Requirements

Table E-20. Document structure MUST conformance requirements

ID	Rule	Reference
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.	16.1.2
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 can be activated in the overlapping region.	16.2.1
M9.3	If a producer specifies a FixedPage.NavigateUri attribute on a <Canvas> element, consumers MUST treat all child elements of that canvas that do not override this value with their own FixedPage.NavigateUri attribute setting as having an associated hyperlink.	16.2.1
M9.4	Relative internal hyperlinks between FixedPage parts MUST specify, at a minimum, the named address relative to the FixedDocument part.	16.2.1
M9.5	In order to be addressable by either a hyperlink or the document outline, the	16.2.1

	named address MUST appear in the <PageContent.LinkTargets> element in the fixed document.	
M9.6	If a named address appears in the <PageContent.LinkTargets> element in the fixed document but is not found in the Name attribute of an element within the associated fixed page, consumers MUST treat the top of the associated fixed page as the named address.	16.2.1
M9.7	If a named address in a URI fragment is not found, consumers MUST ignore the fragment portion of the URI.	16.2.1
M9.8	Internal references MUST specify a page address relative to the fixed document sequence.	16.2.2
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.	16.4.1
M9.10	The Name attribute MUST NOT be specified on any children of a <ResourceDictionary> element.	16.2.3
M9.11	The FragmentName attribute MUST be unique within the scope of the story.	16.1.1.6
M9.12	A <StoryBreak> element MUST NOT be included in a position other than the first or last child element of a <StoryFragment> element.	16.1.2
M9.13	A <TableRowGroupStructure> element is REQUIRED in order to specify a set of <TableRowStructure> elements.	16.1.2.7
M9.14	If specified, the Name value MUST meet the following requirements: The initial character MUST be an underscore character or a letter, that is, it falls within the Lu, Ll, Lo, Lt, and Nl categories. Trailing characters MUST be an underscore character or a letter or number, that is, they fall within the Lu, Ll, Lo, Lt, Nl, Mn, Mc, and Nd categories.	16.2.3

1 ~~E.9.2~~**E.10.2 SHOULD Conformance Requirements**

2 *Table E-21. Document structure SHOULD conformance requirements*

ID	Rule	Reference
S9.1	Every meaningful element in the fixed page markup SHOULD specify a Name attribute in order for the document structure markup to refer to it	16.1.1
S9.2	This requirement was removed prior to Edition 1 of this Standard.	
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.	16.1.1
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	16.1.2

	story fragment that must be merged to determine the cross-fragment content structure.	
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.	16.1.2.2
S9.6	If consumers enable user interactivity, they SHOULD support hyperlink activation and addressing.	16.2
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.	16.2.1
S9.8	The value of the Name attribute on a <FixedPage>, <Canvas>, <Path>, or <Glyphs> element SHOULD be unique within the scope of the fixed document.	16.2.1
S9.9	It is RECOMMENDED that Name attribute values on <FixedPage>, <Canvas>, <Path>, and <Glyphs> elements be unique within an entire fixed document sequence.	16.2.1
S9.10	If the Name 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	16.2.3
S9.11	A hyperlink destination in the same fixed document SHOULD be expressed as a relative URI.	16.2.4
S9.12	This requirement was removed prior to Edition 1 of this Standard. If selection is supported, consumers SHOULD provide a visual cue over or around selected elements.	
S9.13	Selection order within an XPS Document SHOULD follow reading order.	16.3
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.	16.4.1
S9.15	Producers SHOULD order the markup in FixedPage parts to reflect the order in which it is intended to be read.	16.4.1
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.	16.4.1
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.	16.4.1
S9.18	Producers SHOULD order <StoryFragment> elements in each StoryFragments part in their intended reading order.	16.4.1
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	16.4.1

	all un-referenced elements that appear in the fixed page markup, ordered by page number, then markup order	
S9.20	Producers SHOULD order <Story> elements in the DocumentStructure part in their intended reading order.	16.4.1
S9.21	Producers SHOULD order <StoryFragmentReference> elements within a <Story> element in their intended reading order.	16.4.1
S9.22	A screen reader consumer SHOULD read the document according to its reading order.	16.4.2
S9.23	A screen reader SHOULD use the UnicodeString attribute of each <Glyphs> element to determine the text to read.	16.4.2
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.	16.4.2
S9.25	If the screen reader provides features to describe images, it SHOULD read the text provided in the AutomationProperties.Name and AutomationProperties.HelpText attributes.	16.4.2
S9.26	If the screen reader provides features to describe hyperlink addresses, it SHOULD read the text provided in the FixedPage.NavigateUri attribute.	16.4.2
S9.27	Images and graphics SHOULD specify text alternatives for images and graphics to make this content accessible to vision-impaired individuals. The AutomationProperties.Name 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.	16.4.3
S9.28	An image SHOULD specify the AutomationProperties.Name and AutomationProperties.HelpText attributes on the <Path> element that is filled with an <ImageBrush> that describes the content specified by the ImageSource attribute of the <ImageBrush> element.	16.4.3
S9.29	A vector graphic (a collection of one or more <Path> elements representing a single drawing) SHOULD specify the AutomationProperties.Name and AutomationProperties.HelpText attributes only once, directly on a <Canvas> element wrapping the <Path> elements comprising the graphic.	16.4.3
S9.30	Children of <VisualBrush> elements SHOULD NOT be referenced by document structure markup.	16.1.2.13

1 **E.9.3E.10.3 OPTIONAL Conformance Requirements**

2 *Table E-22. Document structure OPTIONAL conformance requirements*

ID	Rule	Reference
O9.1	Producers MAY choose to add document structure information to XPS Documents. Consumers MAY ignore any authored document structure or hyperlinks.	Clause 16
O9.2	Producers MAY provide either the document outline or the document content,	16.1

	or both; consumers MAY ignore either or both.	
09.3	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.	16.1.1
09.4	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.	16.1.1
09.5	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.	16.1.2.6
09.6	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	16.2.1
09.7	Consumers MAY ignore the Name attribute.	16.2.3
09.8	Consumers MAY ignore the FixedPage.NavigateUri attribute.	16.2.4
09.9	Viewing consumers that support interactivity MAY support selection and copying.	16.3
09.10	Consumers MAY use the FragmentType 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.	16.3
09.11	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.	16.4.1
09.12	Consumers MAY use the FragmentType attribute of the <StoryFragment> element to determine reading order by interpreting elements that have FragmentType values of Header and Footer as belonging first or last in the reading order, respectively.	16.4.1
09.13	Screen readers MAY inspect the Indices attribute to resolve potential ambiguities in the UnicodeString attribute.	16.4.2
09.14	The <DocumentStructure> element MAY contain a single <DocumentStructure.Outline> element and zero or more <Story> elements	16.1.1.1
09.15	A <StoryFragment> element MAY be identified with a FragmentName attribute to distinguish it from other fragments for the same story on a single page.	16.1.2.2
09.16	The <TableCellStructure> element defines the appearance of a table cell. It MAY contain nested <TableStructure> elements	16.1.2.9
09.17	The FixedPage.NavigateUri attribute is OPTIONAL.	16.2.4

1 ~~E.10~~**E.11** **XPS Document Package Features**2 ~~E.10.1~~**E.11.1** **MUST Conformance Requirements**3 *Table E-23. XPS Document package feature MUST conformance requirements*

ID	Rule	Reference
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.	17.1
M10.2	Consumers MUST be able to consume packages regardless of their interleaving structure.	17.1.3
M10.3	This requirement was removed prior to Edition 1 of this Standard; its description is retained here for historical purposes. Consumers that lack the resources to process a part MUST instantiate indicate an error condition.	
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.	17.1.3
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).	17.1.3
M10.6	The DiscardControl part MUST NOT reference itself.	17.1.4.1
M10.7	If either the Target attribute or the SentinelPage attribute of the <Discard> element contain an invalid reference (refer outside the package), the <Discard> element MUST be ignored.	17.1.4.1.2
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.	17.2.1
M10.9	Consumers MUST NOT prevent an end user from taking an action solely because doing so will invalidate a signature.	17.2.1
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 §17.2.1.1 are observed. The following parts MUST be signed: All FixedDocument parts referenced in the markup of the FixedDocumentSequence part; FixedDocumentSequence part that is the target of the Start Part package relationship; All FixedPage parts referenced by all signed FixedDocument parts; <SignedInfo> portion of the Digital Signature XML Signature part containing this signature; All parts associated with each signed FixedPage part by means of a Required Resource relationship; All DocumentStructure parts associated via a Document Structure relationship with all signed FixedDocument parts; All StoryFragments parts associated via Story Fragments relationship with all signed FixedPage parts; All SignatureDefinitions parts associated via a	17.2.1.1

	Signature Definitions relationship with any signed FixedDocument part; All Thumbnail parts associated via a Thumbnail relationship from the package root or with any signed FixedPage or FixedDocument part	
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.	17.2.1.1
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.	17.2.1.1
M10.13	An XPS Document digital signature MUST be shown as an incompliant digital signature if it violates any of the signing rules regarding parts or relationships that MUST or MUST NOT be signed.	17.2.1.2
M10.14	An XPS Document digital signature MUST be shown as a broken digital signature if it is not an incompliant digital signature, but the signature fails the signature validation routines described in the OPC.	17.2.1.2
M10.15	An XPS Document digital signature MUST be shown as a questionable digital signature if it is not an incompliant 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.	17.2.1.2
M10.16	An XPS Document digital signature MUST be shown as a valid digital signature if it is not an incompliant, broken, or questionable digital signature.	17.2.1.2
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.	17.2.1.3
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.	17.2.1.4
M10.19	To link a <SignatureDefinition> to a signature, the value of the SpotID MUST be specified in the Id attribute of the corresponding <Signature> element in the Digital Signature XML Signature part.	17.2.2.2.1
M10.20	Due to space and rendering limitations, producers MUST NOT assume that consumers will use the values specified in the <SpotLocation> element.	17.2.2.3
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.	17.2.2.4
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."	17.2.2.5
M10.23	There MUST NOT be more than one DiscardControl package relationship.	17.1.4.1
M10.24	In some cases, producers might rewrite the contents of a package so that	17.1.4.1

	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.	
M10.25	An XPS Document digital signer MUST NOT sign a PrintTicket part if it does not fully understand the PrintTicket content.	17.2.1.1
M10.26	If the SignatureDefinitions part exists, it MUST contain only one <SignatureDefinitions> element.	17.2.2.1
M10.27	If the SignatureDefinitions part exists, there MUST be at least one <SignatureDefinition> element.	17.2.2.2
M10.28	The SpotID attribute is REQUIRED.	17.2.2.2.1
M10.29	The value of this attribute MUST be globally unique to ensure that a Signature part can be linked to only one <SignatureDefinition> element.	17.2.2.2.1

1 ~~E.10.2~~**E.11.2 SHOULD Conformance Requirements**

2 *Table E-24. XPS Document package feature SHOULD conformance requirements*

ID	Rule	Reference
S10.1	When interleaving, the Content Types stream SHOULD be interleaved according to the recommendations in the OPC Standard.	17.1
S10.2	When interleaving, PrintTicket parts SHOULD be written to the package before the part to which they are attached.	17.1
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.	17.1
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 which it is attached.	17.1
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.	17.1
S10.6	It is RECOMMENDED that one empty PrintTicket be shared for all parts that attach an empty PrintTicket.	17.1.1
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.	17.1.4
S10.8	DiscardControl parts that are not well-formed SHOULD NOT be processed	17.1.4.1

	and an error condition SHOULD NOT be instantiated reported .	
S10.9	If a <Discard> element is encountered where either or both of the Target attribute and SentinelPage 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 Target attribute and SentinelPage attribute have been processed or until the end of the package is reached.	17.1.4.1.2
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.	17.1.5
S10.11	Consumers SHOULD inform the end user if an action they are going to take will invalidate an existing signature.	17.2.1
S10.12	This requirement was removed prior to Edition 1 of this Standard; its description is retained here for historical purposes. 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	17.2.1.5
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.	17.2.2.3
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.	17.2.2.3
S10.15	It is RECOMMENDED that a signature spot be a clickable area used to launch the digital signing process.	17.2.2.3
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.	17.2.2.5
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.	17.3
S10.18	Head-first XPS Document consumers SHOULD attempt to detect inconsistent packages as soon as possible and SHOULD instantiate an error condition generate an error message , even if they have already processed the pages that resulted in the error.	17.1
S10.19	The viewing consumer SHOULD use the values specified in the <SpotLocation> element to place a signature spot.	17.2.2.3
S10.20	The relationships for the DiscardControl part and the StartPart SHOULD both be written in the first piece of the package relationship part, and that piece SHOULD be before the first FixedPage part in the package.	17.1
S10.21	The piece of the DiscardControl part that includes a Discard element with a SentinelPage attribute referencing a FixedPage part SHOULD be written to the package before that FixedPage part.	17.1
S10.22	Consumers that support printing of signed documents SHOULD support control through PrintTicket settings pertaining to the treatment of XPS	17.2.1.5

	Documents with invalid or questionable signatures.	
--	--	--

1 **E.10.3E.11.3 OPTIONAL Conformance Requirements**

2 *Table E-25. XPS Document package feature OPTIONAL conformance requirements*

ID	Rule	Reference
O10.1	Interleaving is OPTIONAL.	17.1
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.	17.1.2
O10.3	Consumers MAY discard FixedPage parts once they have been processed.	17.1.3
O10.4	Consumers MAY discard FixedDocument and FixedDocumentSequence parts after all their child elements and their closing tags have been processed.	17.1.3
O10.5	In the absence of explicit directives to the contrary, consumers MAY discard parts as directed by the DiscardControl part.	17.1.3
O10.6	Some producers (typically drivers) MAY choose a suitable interleaving order by modeling the resource management behavior of the consumer.	17.1.4
O10.7	A consumer MAY decide to ignore a malformed DiscardControl part in its entirety or from the first malformed node onward.	17.1.4.1
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.	17.2.1.1
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.	17.2.1.2
O10.10	XPS Documents MAY be signed more than once.	17.2.1.3
O10.11	This requirement was removed prior to Edition 1 of this Standard; its description is retained here for historical purposes. Producers MAY include the JobDigitalSignatureProcessing setting in the job-level PrintTicket within the XPS Document content.	17.2.1.5
O10.12	The SpotID attribute of the <SignatureDefinition> element MAY be used to link to an existing signature.	17.2.2.2.1
O10.13	Consumers MAY choose a size and shape to display a signature spot based on the desired display information and page content.	17.2.2.3
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.	17.2.2.6
O10.15	The <SpotLocation> element is OPTIONAL.	17.2.2.3
O10.16	The following parts MAY be signed: CoreProperties, Digital Signature Certificate, Digital Signature Origin, DiscardControl, and PrintTicket.	17.2.1.1

E.12 Rendering Rules**E.12.1 MUST Conformance Requirements**

Table E-26. Rendering rules MUST conformance requirements

ID	Rule	Reference
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 18
M11.2	If a non-invertible transform is encountered during rendering, consumers MUST omit rendering the affected element and all of its child and descendant elements.	18.1.3
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.	18.1.3
M11.4	Producers MUST NOT assume a specific placement error for curve decomposition or rely on side-effects of a specific consumer implementation.	18.1.5
M11.5	If a consumer encounters markup with characteristics outside its implementation-defined limits, it MUST instantiate an error condition Encountering markup with characteristics outside of the consumer-specific implementation limits MUST cause an error condition.	18.2
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.	18.4.1
M11.7	Composition MUST have the same effect as the application of the rules in §18.5, in sequence.	18.5
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.	18.7.2
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.	18.7.3
M11.10	The Width and Height values specified in the Viewbox and Viewport attributes of an <ImageBrush> or <VisualBrush> element MUST NOT be negative.	18.1.3

1 ~~E.11.2~~**E.12.2 SHOULD Conformance Requirements**

2 Table E-27. Rendering rules SHOULD conformance requirements

ID	Rule	Reference
S11.1	Coordinates are real numbers. 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.	18.1.2, Table 18-1
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.	18.1.4
S11.3	When rendering a shape, a <i>practical</i> implementation (such as a bi-tonal printing device) SHOULD turn on each pixel 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.	18.1.4, 18.6.12
S11.4	When rendering geometries, consumers SHOULD render curves so they appear smooth from a normal viewing distance.	18.1.5
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.	18.1.7
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.	18.1.8
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.	18.1.8
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	18.2

	Documents that stay within these implementation limits.	
S11.9	This requirement was removed prior to Edition 1 of this Standard; its description is retained here for historical purposes. Coordinates are real numbers and SHOULD be computed with at least single floating point precision.	
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.	18.2
S11.11	Gradients SHOULD be rendered according to the guidelines described in §18.3.	18.3
S11.12	Consumers SHOULD pre-process gradient stops for all gradients using the steps described in §18.3.1.1.	18.3.1.1
S11.13	If any gradient stops use an sRGB or scRGB color specification consumers SHOULD blend colors between gradient stops in the color space indicated by the ColorInterpolationMode attribute of the gradient brush, unless a PrintTicket setting provides an alternative blending color space that the consumer understands. If none of the gradient stop elements uses an sRGB or scRGB color specification and the consumer understands the blending color space PrintTicket setting, the blending color space PrintTicket setting SHOULD be used. 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 ColorInterpolationMode attribute of the gradient brush. If none of the gradient stop 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.	18.3.1.2
S11.14	If a ColorInterpolationMode value of SRgbLinearInterpolation is used, the BLEND() function SHOULD convert the color values to sRGB first, and then perform a linear interpolation between them.	18.3.1.2
S11.15	If a ColorInterpolationMode value of ScRgbLinearInterpolation is used, the BLEND() function SHOULD convert the color values to scRGB first, and then perform a linear interpolation between them.	18.3.1.2
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 might be large, resulting in a large change over the extent of a single device pixel. In this case, it is RECOMMENDED that the	18.3.1.2

	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.	
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.	18.3.1.2
S11.18	All opacity calculations SHOULD be performed with at least 8-bit precision to provide sufficient quality for nested content.	18.4
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.	18.4.1
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.	18.5
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.	18.6
S11.22	Consumers SHOULD ensure that parallel edges of strokes appear parallel.	18.6.1
S11.23	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.	18.6.2
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).	18.6.3
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.	18.6.8
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.	18.7.1
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	18.7.1

	when they do	
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.	18.7.2
S11.29	When up-sampling an image presented at a lower resolution than the device resolution, bilinear filtering SHOULD be used.	18.7.2
S11.30	When down-sampling an image presented at a higher resolution than the device resolution, at least a bilinear filter SHOULD be used.	18.7.2
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 StrokeThickness 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.	18.6.12
S11.32	Producers SHOULD NOT create files containing the extreme degenerate case of StrokeDashArray = "0 0". Such lines SHOULD be rendered as a solid line.	18.6.4.6
S11.33	Consumers SHOULD render an element filled with a linear gradient brush such that the appearance is the same as if the steps described in §18.3.2 had been taken.	18.3.2
S11.34	Consumers SHOULD render an element filled with a radial gradient brush such that the appearance is the same as if these steps described in §18.3.3 had been taken.	18.3.3

1 **~~E.11.3~~E.12.3 OPTIONAL Conformance Requirements**

2 *Table E-28. Rendering rules OPTIONAL conformance requirements*

ID	Rule	Reference
O11.1	Very high resolution devices MAY use lower fractional precision than a 28.4 fixed-point representation to represent device coordinates.	18.1.2
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.	18.1.4
O11.3	Devices MAY use sub-pixel masking.	18.1.4
O11.4	An implementation capable of anti-aliasing MAY draw a thin line in a way that blends with the background to varying degrees.	18.1.4
O11.5	A bi-tonal implementation on a printer MAY draw thin lines with or without	18.1.4 _z

	drop-outs, or by applying half-toning, depending on the desired output quality.	18.5
O11.6	Consumers MAY apply pixel placement rules optimized for character rendering to individual glyphs in a <Glyphs> element.	18.1.6
O11.7	Behavior of blending with very close gradient stops MAY vary in an implementation-defined manner (see S11.16).	18.3.1.2
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.	18.3.1.2
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.	18.4.1
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.	18.4.1
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.	18.5
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	18.5.1
O11.13	When doing page composition, if an element is fully transparent (an opacity of 0.0), it MAY be skipped.	18.5.1
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.	18.5.1
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.	18.5.1
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.	18.5.1
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.	18.5.1
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.	18.7.1
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.	18.7.2
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	18.7.3

according to how the tiles fit on a grid.

O11.24	Temporary work canvases MAY be re-used when tiling transparent brushes.	18.7.4
O11.25	Producers MAY generate a <Path> element intended to be treated as having a consistent nominal stroke width by specifying the StrokeDashArray attribute and by specifying the StrokeDashOffset attribute value less than -1.0 times the sum of all the numbers in the StrokeDashArray attribute value.	18.6.12
O11.26	If an implementation chooses to draw thin lines, then it MAY choose to draw them with drop outs, following requirement S11.3 in §18.1.4, or as solid rules of 1 pixel thickness.	18.1.4

~~E.12~~[E.13](#) Additional Conformance Requirements

~~E.12.1~~[E.13.1](#) MUST Conformance Requirements

Table E-29. Additional MUST conformance requirements

ID	Rule	Reference
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.	-
M12.2	A FixedDocument part MUST NOT be referenced more than once by a FixedDocumentSequence part.	-
M12.3	A FixedPage part MUST NOT be referenced more than once <i>in total</i> , throughout all FixedDocument parts.	-
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.	-
M12.5	This requirement was removed prior to Edition 1 of this Standard; its description is retained here for historical purposes. Any FixedDocumentSequence, FixedDocument, or FixedPage part that is reachable from the primary fixed payload root or its related parts by relationship or by the Source attribute on a <DocumentReference> or <PageContent> element MUST have no more than one attached PrintTicket part.	-
M12.6	Every Font part reachable from the primary fixed payload root or its related parts by relationship or by the Source attribute on a <DocumentReference> or <PageContent> element MUST be a valid OpenType font.	-

M12.7	The content types in the tables below MUST NOT include parameters. If a consumer encounters the presence of parameters on these content types when the affected part is accessed it MUST instantiate an error condition. 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.	D.2 E.2
-------	---	----------------------------------

1

2 ~~E.13~~**E.14** 3D Graphic Content

3 ~~E.13.1~~**E.14.1** MUST Conformance Requirements

4 *Table E-30. 3D Graphic Content MUST conformance requirements*

ID	Rule	Reference
M13.1	A consumer that renders XPS pages for printing on a 2D medium MUST NOT read and render the 3D content to a 2D representation unless at least one of the following is true: It has been explicitly configured to do so by the user; or it provides explicit feedback to the user that the 3D content is being used rather than the 2D representation.	A
M13.2	The X3D file MUST conform to one of the following standards meeting the X3D "Interoperability" conformance level: ISO/IEC FCD 19775-1r1:200x , ISO/IEC 19776-1:2005 , ISO/IEC 19776-3:2007 .	A
M13.3	3D Producers MUST use a <Brush3D> element within a <Path.Fill> element, with Markup Compatibility, to place the 3D content on a page within a defined Viewport.	A
M13.4	3D producers MUST define a conforming <AlternateContent.Fallback> element within the parent <Path> element of the <Brush3D> element. The <AlternateContent.Fallback> element MUST contain a 2D representation for viewing and printing of the 3D content.	A
M13.5	3D Producers MUST define the AlternateContent.Fallback that is visually representative of the default 3D viewpoint.	A
M13.6	Implementations that modify the 3D default viewpoint MUST update the AlternateContent.Fallback to match that 3D viewpoint prior to printing or saving the file	A
M13.7	3D Consumers MUST display at least the AlternateContent.FallBack.	A
M13.8	The active 3D window display MUST be contained within the defined Brush3D.Viewport.	A

1 **E.13.2 E.14.2 SHOULD Conformance Requirements**2 *Table E-31. 3D Graphic Content SHOULD conformance requirements*

ID	Rule	Reference
S13.1	3D Consumers SHOULD allow user interaction to navigate and play animations present in the 3D graphics content.	A
S13.2	3D Producers generating X3D model content SHOULD follow these rules: 1. Use Triangle based X3D elements to facet the model. 2. Use positive values for all X Y Z coordinates. 3. Define a face normal for all Triangle faces. 4. Define Triangle faces counter clockwise (right-hand rule) when facing outwards or upwards in ground terrain models. 5. Ensure that all triangle faces share two points with their neighboring faces.	A
S13.3	✖ This requirement was removed prior to Edition 1 of this Standard; its description is retained here for historical purposes. 3D Producers SHOULD define equivalent AlternateContent.Fallback sufficient for 2D viewing and printing	G
S13.4	3D Consumers SHOULD display the X3D 3D content rendered to display the default 3D viewpoint and perspective as defined in the X3D part.	A
S13.5	3D Consumers SHOULD allow user interaction to navigate and play animations present in the 3D graphics content.	J

3 **E.13.3 E.14.3 OPTIONAL Conformance Requirements**4 *Table E-32. 3D Graphic Content OPTIONAL conformance requirements*

ID	Rule	Reference
O13.1	An XPS producer MAY include three-dimensional (3D) graphics within an XPS package.	A

5 **End of informative text.**

1 **F. 3D Graphic Content**

2 An XPS producer MAY include three-dimensional (3D) graphics within an XPS package [O13.1].
3 The 3D graphics content provides XPS documents with animatable 3D models to supplement
4 the document text providing a visually richer user experience in 3D Consumers for purposes
5 such as, but not limited to:

- 6 1. 3D model examination and walkthroughs.
- 7 2. Animations depicting assembly instructions.
- 8 3. Animations showing usage instructions.
- 9 4. Animations depicting proposed phased building construction projects.

10 3D content is included in such a way that an alternative representation that is suitable for use
11 in two-dimensional (2D) rendering is provided for a consumer that does not support
12 3D content; e.g., for printing on paper.

13 This Annex does not introduce any additional requirements for a producer or consumer that
14 does not support 3D content.

15 A consumer that renders XPS pages for printing on a 2D medium MUST NOT read and render
16 the 3D content to a 2D representation unless at least one of the following is true [M13.1]:

- 17 1. It has been explicitly configured to do so by the user; or
- 18 2. It provides explicit feedback to the user that the 3D content is being used rather than
19 the 2D representation.

20 ~~If the FixedPage containing the 3D content has been signed, then a consumer that renders XPS~~
21 ~~pages for printing on a 2D medium MUST NOT read and render the 3D content to a~~
22 ~~2D representation even if the requirements above are met.~~

23 An instance of 3D graphics content is created by placing a conformant X3D file within the XPS
24 document OPC Package. The X3D file MUST conform to one of the following standards meeting
25 the X3D "Interoperability" conformance level [M13.2]: ISO/IEC FCD 19775-1r1:200x, ISO/IEC
26 19776-1:2005, or ISO/IEC 19776-3:2007.

27 Markup compatibility is used to encapsulate the 3D content and its 2D alternative
28 representation.

29 A 3D Content Capable Producer (3D Producer) is defined as an XPS producer that understands
30 the "<http://schemas.openxps.org/openxps/3d/2008>" namespace.

31 A 3D Content Capable Consumer (3D Consumer) is defined as an XPS consumer that
32 understands the "<http://schemas.openxps.org/openxps/3d/2008>" namespace.

33 3D Producers generating X3D model content SHOULD follow these rules [S13.2]:

- 34 1. Use Triangle based X3D elements to facet the model.
- 35 2. Use positive values for all X Y Z coordinates.
- 36 3. Define a face normal for all Triangle faces.

1 4. Define Triangle faces counter clockwise (right-hand rule) when facing outwards or
2 upwards in ground terrain models.

3 5. Ensure that all triangle faces share two points with their neighboring faces.

4 ****3D Producers MUST use a <Brush3D> element within a <Path.Fill> element, with Markup**
5 **Compatibility, to place the 3D content on a page within a defined Viewport [M13.3].**

6 3D producers MUST define a conforming <AlternateContent.Fallback> element within the
7 parent <Path> element of the <Brush3D> element. The <AlternateContent.Fallback> element
8 MUST contain a 2D representation for viewing and printing of the 3D content [M13.4].

9 3D Producers MUST define the AlternateContent.Fallback that is visually representative of the
10 default 3D viewpoint [M13.5].

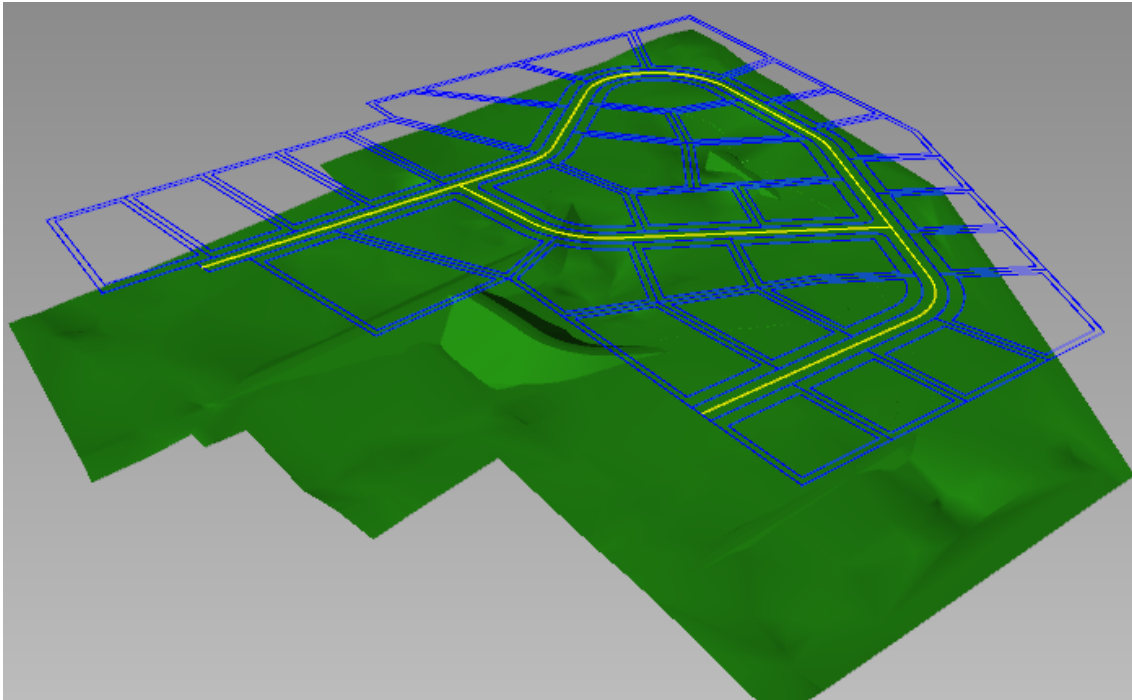
11 ~~3D Producers SHOULD define equivalent AlternateContent.Fallback sufficient for 2D viewing and~~
12 ~~printing [S13.3].~~

13 Implementations that modify the 3D default viewpoint MUST update AlternateContent.Fallback
14 to match that 3D viewpoint prior to printing or saving the file [M13.6].

15 *Example F-1. 3D graphics content in FixedPage.fpage*

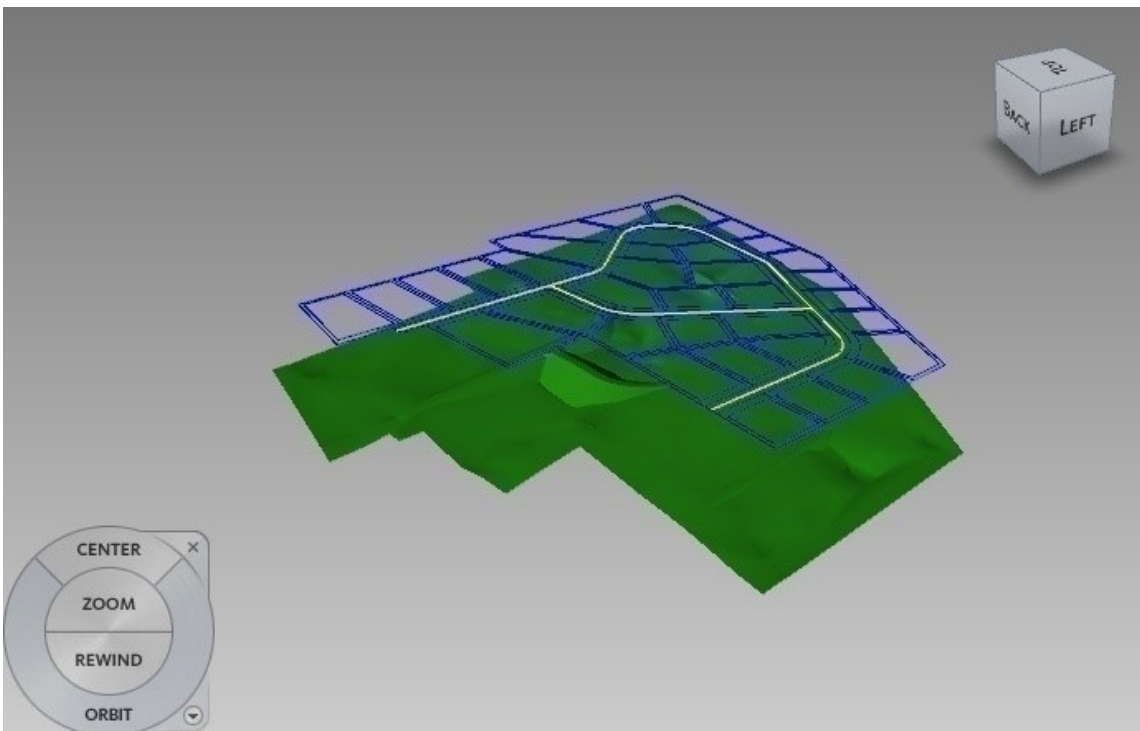
```
16 <FixedPage xmlns="http://schemas.microsoft.com/xps/2005/06" Height="1056"
17 Width="816" xml:lang="und"
18 xmlns:x3d="http://schemas.openxps.org/openxps/3d/2008"
19 xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006">
20 <Canvas Name="dwfresource_1" RenderTransform="96, -0, -0,96,0,0">
21 <Path Data="M0,0 L8.5,0 L8.5,11 L0,11 z">
22 <Path.Fill>
23 <mc:AlternateContent>
24 <mc:Choice Requires="x3d">
25 <x3d:Brush3D x3d:Source3D="olympus.x3d" x3d:Viewport="0,2.3125,
26 8.5,8.6875" x3d:ViewportUnits="Absolute" x3d:Viewbox="0,0,
27 640,640" x3d:ViewboxUnits="Absolute"/>
28 </mc:Choice>
29 <mc:Fallback>
30 <ImageBrush ImageSource="ProxyGraphics.png" Viewport="0,2.3125,
31 8.5,8.6875" ViewportUnits="Absolute" Viewbox="0,0, 640,640"
32 ViewboxUnits="Absolute"/>
33 </mc:Fallback>
34 </mc:AlternateContent>
35 </Path.Fill>
36 </Path>
37 </Canvas>
38 </FixedPage>
```

39 Non-3D Consumers will display AlternateContent.Fallback; that is, in this example, the following
40 2D image:



1

2 For this example, 3D Consumers should display an active, navigatable 3D window similar to the
3 image below. [rex9]



4

5 *end example]*

1 *Example F-2. 3D graphics content in FixedPage.fpage*

```

2   <FixedPage xmlns="http://schemas.microsoft.com/xps/2005/06" Height="1056"
3   Width="816" xml:lang="und"
4   xmlns:x3d="http://schemas.openxps.org/openxps/3d/2008"
5   xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006">
6   <Canvas Name="dwfresource_1" RenderTransform="96, -0, -0,96,0,0">
7     <Path>
8       <mc:AlternateContent>
9         <mc:Choice Requires="x3d">
10          <Path.Fill>
11            <x3d:Brush3D x3d:Source3D="olympus.x3d" x3d:Viewport="0,2.3125,
12              8.5,8.6875" x3d:ViewportUnits="Absolute" x3d:Viewbox="0,0,
13              640,640" x3d:ViewboxUnits="Absolute"/>
14          </Path.Fill>
15        </mc:Choice>
16        <mc:Fallback>
17          <Path.Data>
18            <PathGeometry>
19              <PathFigure StartPoint="20,80">
20                <PolyBezierSegment Points="70,0 120,160 170,80 120,0 70,160
21                  20,80"/>
22              </PathFigure>
23            </PathGeometry>
24          </Path.Data>
25        </mc:Fallback>
26      </mc:AlternateContent>
27    </Path>
28  </Canvas>
29 </FixedPage>

```

30 *end example]*

31 `XX[rex10]`

32 3D Consumers MUST display at least the AlternateContent.FallBack [M13.7].

33 3D Consumers SHOULD display the X3D 3D content rendered to display the default
34 3D viewpoint and perspective as defined in the X3D part [S13.4].

35 3D Consumers SHOULD allow user interaction to navigate and play animations present in the
36 3D graphics content [S13.1 ~~S13.5~~]. The active 3D window display MUST be contained within the
37 defined Brush3D.Viewport [M13.8].

38

targetNamespace: <http://schemas.openxps.org/openxps/3d/2008>

39 **F.1 Brush3D**

40 Table 13–1. Brush types is extended by the addition of the following brush type:

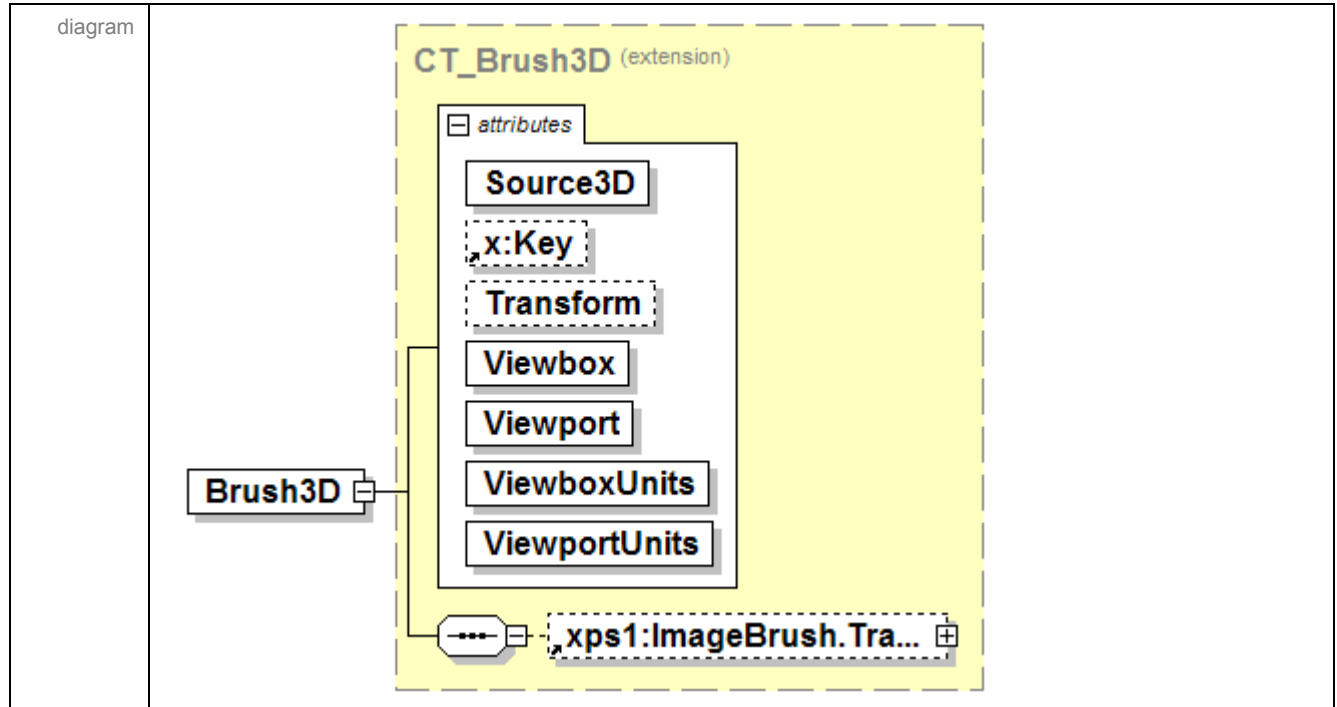
41

<u>Name</u>	<u>Description</u>
3D Content Brush	Fills a region with a 3D graphics model (Annex A)

1 In the following diagrams and text the prefix "fp" refers to the FixedPage name space. See §19
 2 for definitions of those elements and attributes. [xx\[rex11\]](#)

3 [xx\[rex12\]](#)

4



namespace	http://schemas.openxps.org/openxps/3d/2008					
type	extension of CT_Brush3D					
properties	content complex					
children	xps1:ImageBrush.Transform					
attributes	Name	Type	Use	Default	Fixed	annotation
	Source3D	ST_UriImage3D	required			
	x:Key					
	Transform	xps1:ST_RscRefMatrix				
	Viewbox	xps1:ST_ViewBox	required			
	Viewport	xps1:ST_ViewBox	required			
	ViewboxUnits	xps1:ST_ViewUnits	required		Absolute	
	ViewportUnits	xps1:ST_ViewUnits	required		Absolute	
source	<pre><xs:element name="Brush3D"> <xs:complexType> <xs:complexContent> <xs:extension base="CT_Brush3D"/> </xs:complexContent> </xs:complexType> </xs:element></pre>					

	<pre></xs:complexType> </xs:element></pre>
--	--

1 [Example:

2 <Brush3D Source3D="olympus.x3d" Viewport="0,2.3125, 8.5,8.6875"

3 ViewportUnits="Absolute" Viewbox="0,0, 640,640" ViewboxUnits="Absolute"/>

4 This example shows the required attributes and example values. The Source3D attribute is a
5 URI that must resolve to a conforming X3D file contained within the XPS OPC Package. See §19
6 for definitions of Viewport, ViewportUnits, Viewbox, ViewBoxUnits and ImageBrush.Transform.
7 *end example*]

1 **G. Bibliography**

2 Independent JPEG Group. <http://www.ijg.org/files/>

3 *A Nonaliasing, Real-Time Spatial Transform Technique*. Fant, Karl M. *IEEE Computer Graphics*
4 *and Applications* 6 (Jan. 1986): 71–80.

5 *OS/2 and Windows Metrics*. Microsoft Corporation. 2001.

6 <http://www.microsoft.com/typography/otspec/os2.htm>

7 *OpenType Font File*. Microsoft Corporation. 2001.

8 <http://www.microsoft.com/typography/otspec/otff.htm>

9 *OpenType Specification, Version 1.4*. Microsoft Corporation. 2004.

10 <http://www.microsoft.com/typography/otspec/default.htm>

11 ~~*Print Schema*. Microsoft Corporation. 2006. [http://windowssdk.msdn.microsoft.com/en-](http://windowssdk.msdn.microsoft.com/en-us/library/default.aspx)~~
12 ~~[us/library/default.aspx](http://windowssdk.msdn.microsoft.com/en-us/library/default.aspx)~~

13 Windows Media Photo Microsoft Corporation. <http://www.microsoft.com/xps>

14 X3D Specification Web3D, Consortium. <http://www.web3d.org>

1 **H. Index**

2 In the index that follows, italic page numbers are used to indicate illustrations and examples
 3 with illustrations. Bold page numbers are used to indicate a primary reference when several
 4 pages are listed. Page ranges are elided. "See" references indicate the primary index location
 5 for that topic, while "See also" references indicate related index topics.

6	<hr/>	46	gradient computations	269–76
7	3	47	gradient stop	
8	3D Graphic Content	48	color, specifying	165
		49	described	165
		50	offset	165
		51	specifying	165
9	<hr/> A	52	image brush	
10	abbreviated geometry syntax	53	described	127 , 129
11	syntax	54	example	129
12	accessibility	55	image	<i>See</i> image
13	document structure, enabled by	56	mentioned	28, 69
14	image text alternative, long	57	source, specifying	128
15	described	58	tile size and placement	<i>See</i> brush, view port
16	on canvas	59	tile source	<i>See</i> brush, view box
17	on path	60	linear gradient brush	
18	image text alternative, short	61	color interpolation mode	151
19	described	62	described	152
20	on canvas	63	end point	151
21	on path	64	example	152
22	importance of	65	gradient stops, specifying	156
23	mentioned	66	mappng mode	151
24	of text	67	rendering	271
25	page elements, requirement to expose all	68	specifying	151
26	reading order	69	spread method	
27	document structure, dependent on	70	described	153
28	fragment type, dependent on	71	Pad	153
29	markup order, dependent on	72	Reflect	154
30	of page-centric application	73	Repeat	155
31	of story-centric application	74	specifying	151
32	screen reader	75	start point	151
33	considerations	76	opacity	<i>See</i> opacity
34	mentioned	77	radial gradient brush	
35	alpha	78	described	159
36	anti-aliasing	79	example	159
37	disabling of	80	gradient center	157
38	rendering of	81	gradient origin	157
39	arc	82	gradient stops, specifying	164
		83	gradient x-radius	157
		84	gradient y-radius	157
		85	mappng mode	157
40	<hr/> B	86	rendering	273
41	bleed area	87	specifying	157
42	bookmark	88	spread method	
43	brush	89	described	160
44	alpha	90	Pad	160
45	described	91	Reflect	162
		92	Repeat	163

1	specifying	157	55	circle	See geometry, segment, arc
2	solid color brush	126, 127	56	clipping	
3	tile		57	described	178
4	behavior	See brush, tile, mode	58	elements applied to	169, 170
5	image scaling	296	59	geometry, reference to	78
6	mode		60	of canvas	63, 178
7	described	140–50	61	of glyphs	100, 180
8	FlipX	145–46	62	of path	71, 179
9	FlipXY	149–50	63	rules	267
10	FlipY	147–48	64	CMYK	See color, color space, CMYK
11	mentioned	135	65	color	
12	None	140–42	66	alpha	See opacity
13	specifying	128, 131, 135	67	blending	9
14	Tile	143–44	68	behavior described	216
15	placement	See brush, view port size	69	blend color space for linear gradient brush	151
16	size	See brush, view port	70	for gradients	270
17	small tile rendering	296	71	implementation dependent	216
18	source	See brush, view box	72	brush, specifying for	See brush
19	transparent brush tiling	297	73	color profile	
20	transformation	See transformation	74	embedded in image . 28, 29, 30, 31, 32, 33, 203, 214, 444	
21	view box		75	ICC profile	
22	calculating source coordinates for images	135	76	color channels supported	203
23	described	135	77	described	203
24	example	136–39	78	grayscale image, usage with	203
25	larger than image	135, 139	79	parts	See parts
26	mapping to view port	128, 131	80	profile types allowed	203
27	specifying for image brush	128	81	Windows Color System (WCS) profile	
28	specifying for visual brush	131	82	embedded in ICC profile	204
29	syntax	128, 131	83	color separation	215
30	unit type	128, 131, 135	84	color space	
31	units, for images	128, 135	85	CMYK	202
32	view port		86	in images	201
33	described	135	87	in vector graphics	201
34	example	136–39	88	gray colors	202
35	placement precision	296	89	ICC profile	
36	specifying for image brush	128	90	version supported	203, 220, 442, 448, 450
37	specifying for visual brush	131	91	named color	10 , 202
38	syntax	128, 131	92	n-channel	202
39	unit type	128, 131, 135	93	in images	201
40	visual brush		94	in vector graphics	201
41	described	131 , 132	95	scRGB	202
42	example	133	96	in images	201
43	visual	131, 132	97	in vector graphics	201
44	C		98	spot colors	See color, color space, named colors
45	canvas		99	sRGB	201
46	anti-aliasing control	63	100	in images	201
47	clipping	See clipping	101	in vector graphics	201
48	composing properties	65	102	support required	201
49	described	63	103	support summarized	201
50	opacity	See opacity	104	fidelity, improved	201
51	opacity mask	See opacity mask	105	raster image support	
52	transformation	See transformation	106	associating a color profile part	214
53	caret stop	See selection, caret stop	107	CMYK	212
54	CFF font	See font:CFF	108	device color	213
			109	gray colors	211
			110	named colors	213
			111	n-channel	212

1	scRGB	211	55	markup example	260
2	sRGB	210	56	mentioned	24
3	syntax		57	namespace	<i>See namespace, See namespace</i>
4	CMYK	207	58	relationship to	26
5	named color	209	59	sign by date and time	264
6	n-channel	208	60	signer name	261
7	scRGB	206	61	signing intent	263
8	sRGB	205	62	signing location	264
9	summarized	204	63	specifying	261
10	where used	204	64	spot ID	261, 262
11	color profile	<i>See color, color profile</i>	65	spot location	262
12	composability of properties	<i>See XPS Document format,</i>	66	signature policy	
13	properties, composability		67	conditions where policy does not apply	258
14	compression		68	described	257
15	image	<i>See image</i>	69	markup compatibility impact	258, 259
16	package	<i>See Open Packaging Conventions, Standard</i>	70	parts to sign	
17	conformance		71	optional	257
18	inherited from Open Packaging Conventions	19, 417	72	required	257
19	of implementation	3	73	relationships to sign	
20	requirements tables	413–67	74	as a group, required	258
21	consumer	9	75	conditionally required	258
22	implementation burden	413–67	76	required	258
23	content area	<i>See page, content area</i>	77	signing rules	11, 257–58
24	content type	9	78	signing validity	259
25	namespace	<i>See namespace</i>	79	single signature	258
26	summarized	412	80	signature spot	<i>See digital signature, signature definitions</i>
27	usage of	23	81	signature status	
28	contour intersection point	9	82	broken	9
29	copy and paste	<i>See also selection</i>	83	broken digital signature	259
30	document structure, improved by	223	84	compliant	9
31	core properties	<i>See Open Packaging Conventions</i>	85	incompliant	9
32	curve	<i>See geometry, segment</i>	86	incompliant digital signature	259
<hr/>					
33	D		87	questionable	9
34	device	9	88	questionable digital signature	259
35	device color	<i>See color, color space, device color</i>	89	valid	9
36	digital signature		90	valid digital signature	259
37	certificate		91	discard control	
38	relationship to	26	92	consumer considerations	254
39	store	259	93	elements	255
40	validity	259	94	markup example	255
41	co-signature	<i>See digital signature, signature definitions</i>	95	namespace	<i>See namespace</i>
42	multiple signatures	259	96	part	<i>See parts</i>
43	namespace	<i>See namespace</i>	97	reference, invalid	256
44	Open Packaging Conventions, extended from	256	98	reference, not yet encountered	256
45	origin		99	resource constraints, addressing	254
46	part	<i>See parts</i>	100	sentinel page	256
47	relationship to	26	101	target resource	256
48	parts	<i>See parts</i>	102	usage of	254, 268
49	printing	259	103	document	
50	relationship to	26	104	markup	<i>See XPS elements, document-level</i>
51	relationships	<i>See relationships</i>	105	namespace	<i>See namespace</i>
52	request	<i>See digital signature, signature definitions</i>	106	order of	51
53	signature definitions	11	107	reference to	51
54	described	42, 260	108	document conventions	13
			109	diagram notes	13
			110	document outline	<i>See document structure, outline</i>
			111	document roll-up	<i>See document, order of</i>

1	document sequence, namespace	See namespace	58	relation to fixed page	223
2	document structure			59	story belonged to	235
3	constructed algorithmically	42	60	story, not belonging to any	236
4	constructed explicitly	42	61	story fragments	See namespace, See namespace
5	content	223	62	table		
6	content structure	9	63	cell		
7	document content	9	64	column span	241
8	figure	242	65	merging	230
9	list			66	row span	241
10	item			67	specifying	241
11	marker	242	68	row group, specifying	240
12	specifying	242	69	row, specifying	240
13	specifying	241	70	specifying	240
14	markup example	224	71	thread	See document structure, story
15	named element	10	72	usage of	42
16	canvas descendant elements	224	73	usage optional	223
17	described	223	74	driver	9
18	link target, optimizing location with	224	75	described	254
19	markup compatibility, updating for	243				
20	referencing each once	223				
21	specifying	243				
22	visual brush descendants prohibited	224				
23	namespace	See namespace	76	E		
24	naming page elements	169, 223	77	effective coordinate space	See layout, coordinate space
25	outline	9, 223	78	elements	See XPS elements
26	language	See language, of outline , See language, of	79	error condition	4
27	outline			80	EXIF		
28	levels	226	81	usage in JPEG	See image, JPEG, EXIF
29	markup described	225	82	usage in TIFF	See image, TIFF, EXIF
30	markup example	226	83	extensibility of XPS Documents	See markup compatibility
31	mentioned	42				
32	outline entry	225				
33	target URI	226				
34	paragraph			84	F		
35	specifying	239	85	figure		
36	parts	See parts	86	illustration	See document structure, figure
37	relationships	See relationships	87	shape	See geometry, figure
38	section			88	fill		
39	specifying	239	89	algorithm	See geometry, fill algorithm
40	story	12	90	of path	See path, fill brush
41	described	42, 223	91	of stroke	See path, stroke brush
42	markup	227	92	find	49, 112, 119
43	markup example	229	93	fixed document sequence	See document, order of
44	story fragment, correlating to page	228	94	fixed page	See page
45	story fragment, reference to	228	95	fixed payload	See XPS Document format
46	story fragment	12	96	FixedDocument part	10
47	break indicator	230, 239	97	FixedDocumentSequence part	10
48	content structure, contains	43, 223	98	FixedPage part	10
49	described	43, 223	99	font		
50	fragment name	235	100	CFF	35, 114
51	fragment name, uniqueness	228	101	compatibility encoding	38
52	fragment type	235	102	device font	100, 118
53	markup described	233, 235	103	embedding	35–38, 37
54	markup elements summarized	229	104	extraction	36, 37
55	markup example	236, 237	105	language impact on copy and paste	49
56	merging fragments	230–33	106	licensing rights	35, 37, 38
57	referencing every page element	236	107	obfuscation	36, 37
				108	algorithm	37
				109	OpenType	35

1	parts	See parts	53	addressability	
2	rasterization	36	54	appearance in link targets	244
3	relationships	See relationships, required resource	55	mentioned	53, 244
4	restricted editing	See relationships, restricted font	56	missing name, handling of	244
5	restricted font	27	57	name	244
6	sharing	35	58	name uniqueness	244
7	subsetting	35, 36	59	of canvas	63, 244
8	TrueType	35	60	of glyphs	100, 244
9	TrueType collection (TTC)	35, 429	61	of page	55, 244
10	Unicode encoding	35	62	of path	71, 244
11	usage of	35	63	of visual brush contents	244
<hr/>					
12	G		64	page number	244
13	geometry		65	document-level listing	See hyperlink, target
14	abbreviated syntax	81, 91–97, 434	66	example	244
15	algorithm	399	67	mentioned	223
16	circle	See geometry, segment, arc	68	overlapping behavior	243
17	curve	See geometry, segment	69	source	
18	described	77	70	base URI	245
19	figure		71	described	245
20	closed	82, 90, 94	72	from canvas	63, 243
21	described	77	73	from glyphs	100, 243
22	fill control	82	74	from path	71, 243
23	markup	82	75	inheritance of	243
24	reference to	79, 81	76	specifying	169
25	segments, composed of	77	77	support recommended	245
26	start point	82, 92	78	target	
27	stroking of segments	77	79	addressability	See hyperlink, addressability
28	figures, composed of	77	80	document-level definition of	54
29	fill algorithm		81	external	243
30	described	80	82	internal	243
31	EvenOdd	80	83	link target, specifying as	53
32	mentioned	77, 435	84	missing behavior	244
33	NonZero	81	85	name	See name
34	specifying	79, 92	86	relative target handling	244
35	filled area	79	87	relative target recommended	245
36	segment		88	relative to document, at minimum	243
37	arc	83–86, 93	<hr/>		
38	Bézier curve	87, 93	89	I	
39	Bézier curve, quadratic	89, 93	90	ICC	See color, color profile
40	Bézier curve, smooth	93, 95	91	image	
41	line	88, 92	92	brush	See brush, image brush
42	segment, degenerate	294	93	JPEG	24, 28
43	transformation	See transformation	94	APP markers	28, 29
44	usage described	78	95	CMYK	29
45	glyphs	See text	96	EXIF	28
46	gradient	See brush, See brush	97	naming	28
47	graphics	See path	98	specification	28
48	grouping markup	See canvas	99	parts	See parts
<hr/>					
49	H		100	PNG	24, 28, 29
50	hairline	See stroke, hairline	101	chunks	30
51	hyperlink		102	naming	29
52	activation	243	103	specification	29
			104	relationships	See relationships, required resource
			105	resolution	135
			106	resource	See resource

1	sharing.....	28	53	examples.....	281–83
2	thumbnail.....	24	54	optimization.....	280
3	TIFF.....	24, 28, 30	55	rules.....	279
4	alpha, associated.....	33	56	coordinate rounding.....	265
5	CCITT bilevel encoding.....	32	57	coordinate space.....	181
6	CMYK.....	32	58	composability.....	56
7	compression.....	33	59	effective coordinate space.....	9
8	EXIF.....	33	60	described.....	48
9	features, supported.....	32	61	mentioned ... 55, 71, 100, 111, 125, 135, 151, 157, 170,	
10	image file directory (IFD).....	32	62	176.....	
11	naming.....	30	63	mentioned.....	51
12	specification.....	30, 33	64	origin.....	265
13	tags, supported.....	30	65	transformation.....	<i>See transformation</i>
14	tags, unsupported.....	32	66	units.....	265
15	tags, unsupported.....	33	67	x-axis.....	265
16	variations, handling.....	33	68	y-axis.....	265
17	types supported.....	28, 129	69	degenerate segments.....	294
18	usage described.....	28, 69	70	implementation limits.....	<i>See implementation limits</i>
19	Windows Media Photo.....	24, 28, 33	71	page dimensions.....	<i>See page</i>
20	CMYK.....	34	72	pixel.....	
21	compression.....	<i>See image, Windows Media Photo,</i>	73	center location.....	<i>See layout, pixel</i>
22	specification.....		74	inclusion.....	266
23	features supported.....	34	75	placement.....	266
24	grayscale.....	34	76	placement behavior for glyphs.....	267
25	named color.....	34	77	placement error maximum.....	267
26	naming.....	33	78	rendering.....	266
27	N-channel.....	34	79	sub-pixel masking.....	267
28	profiled RGB.....	34	80	PrintTicket interactions.....	58
29	scRGB.....	34	81	shape abutment.....	267
30	specification.....	33, 475	82	line.....	
31	sRGB.....	34	83	characteristics of.....	<i>See stroke</i>
32	implementation limits.....	268–69	84	curved.....	<i>See geometry, segment</i>
33	implementation-defined behavior.....	10	85	drawing of.....	<i>See path</i>
34	ink area.....	<i>See page, content area</i>	86	geometry of.....	<i>See geometry, segment, line</i>
35	interleaving.....	<i>See Open Packaging Conventions, interleaving</i>	87	linear gradient.....	<i>See brush, linear gradient brush</i>
<hr/>			88	link.....	<i>See hyperlink</i>
36	J		89	link target.....	<i>See hyperlink, target</i>
37	JPEG.....	<i>See image, JPEG</i>	90	list.....	<i>See document structure, list</i>
<hr/>			<hr/>		
38	L		91	M	
39	language		92	markup compatibility	
40	markup.....	50, 170	93	digital signature, impacted by.....	258
41	of canvas.....	63	94	document structure, usage in.....	243
42	of glyphs.....	100, 119	95	mentioned.....	1
43	of outline.....	50, 225	96	namespace.....	<i>See namespace</i>
44	of outline entry.....	50, 226	97	mentioned.....	47
45	of page.....	55	98	preprocessing requirements.....	46
46	of path.....	71	99	processing required.....	46
47	of resource.....	174	100	property elements, usage with.....	48
48	of signature definition.....	261	101	resource dictionary, usage in.....	178
49	usage.....	49	102	usage of.....	46
50	layout.....	<i>See also transformation, matrix</i>	103	memory management, device.....	<i>See interleaving; discard</i>
51	composition		104	control	
52	behavior.....	280	105	miter.....	<i>See stroke, line, join</i>

1 **N**

2 name

3 elements applied to169

4 link target correspondence245

5 link target, specifying as53

6 purpose of244

7 resource entries, prohibited for245

8 syntax245

9 uniqueness244

10 named color *See color, color space, named color*

11 namespace411

12 content type411

13 core properties411

14 digital signatures411

15 discard control411

16 document411

17 document sequence411

18 document structure411

19 markup compatibility411

20 page411

21 relationships411

22 resource dictionary key411

23 signature definitions411

24 story fragments411

25 naming of parts *See parts, naming recommendations*

26 natural language *See language*

27 n-channel color *See color, color space, n-channel*

28 **O**

29 opacity

30 blending *See color, blending*

31 brush initial opacity135

32 composition effects280

33 computations276–78

34 described170

35 elements applied to169

36 mask *See opacity mask*

37 of canvas63

38 of color34, 126, 151, 157, 201, 204, 205, 207, 208, 209, 216, 220

39 of glyphs100

40 of image brush128

41 of linear gradient brush151

42 of path71

43 of pixel formats210, 211, 212

44 of radial gradient brush157

45 of solid color brush126

46 of stroke280

47 of visual brush131

48 pre-multiplied alpha278

49 superluminous colors278

50 transparent brush tiling297

51 value range276

52 opacity mask

54 brush, filling with *See brush*

55 described166

56 elements applied to169, 170

57 example166, 167

58 of canvas63, 197

59 of glyphs100, 199

60 of path71, 198

61 Open Packaging Conventions

62 ordering

63 simple10

64 Open Packaging Conventions

65 ordering

66 interleaved10

67 Open Packaging Conventions

68 package10

69 Open Packaging Conventions

70 package

71 packaging model10

72 Open Packaging Conventions

73 package

74 relationship10

75 Open Packaging Conventions

76 physical model10

77 Open Packaging Conventions

78 interleaving

79 piece10

80 Open Packaging Conventions

81 physical model21

82 Open Packaging Conventions

83 package

84 packaging model21

85 Open Packaging Conventions

86 interleaving

87 optimization249

88 Open Packaging Conventions

89 interleaving

90 parsing head-first vs. tail first250

91 Open Packaging Conventions

92 interleaving

93 consumer considerations254

94 Open Packaging Conventions

95 digital signature *See digital signature*

96 Open Packaging Conventions

97 core properties264

98 Open Packaging Conventions

99 core properties

100 namespace *See namespace*

101 OpenType font *See font:OpenType*

102 optimization

103 for streaming consumption *See Open Packaging Conventions, interleaving*

104 of composition rules280

105 of digital signatures256

106 of glyphs119

107 of interleaving251–54

108 of named element location224

109 of pixel placement rules267

1 outline..... See document structure, outline

2 **P**

3 packaging model..... See Open Packaging Conventions

4 page

5 bleed area 55, 56

6 content area 55, 57, 265

7 height of 55, 265

8 height, advisory 53

9 layout See also layout

10 markup grouping See canvas

11 namespace See namespace

12 order of pages 52

13 orientation of 58

14 reference to 52

15 root of 55

16 scaling for print 58

17 uniqueness of 53

18 width of 55, 265

19 width, advisory 53

20 paragraph See document structure, paragraph

21 part **10**

22 part name **10**

23 parts See also XPS Document format

24 core properties See Open Packaging Conventions, Standard

25 digital signature

26 certificate

27 part See Open Packaging Conventions, Standard

28 digital signature origin See Open Packaging Conventions,

29 Standard

30 DiscardControl 24, 254

31 DocumentStructure 24, 42, 223

32 FixedDocument 23, 27

33 FixedDocumentSequence 23, 27

34 FixedPage 23, 28

35 font 23, 35–39

36 ICC profile 24

37 image 24, 28–34

38 naming recommendations 43–45

39 PrintTicket 24, 40

40 remote resource dictionary 24, 39

41 SignatureDefinitions 24, 42

42 StoryFragments 24, 43, 229

43 thumbnail 24, 34

44 thumbnail, package See Open Packaging Conventions,

45 Standard

46 XML digital signature See Open Packaging Conventions,

47 Standard

48 path

49 clipping See clipping

50 described 69, 70

51 fill brush 71

52 geometry, reference to 71, 74

53 opacity See opacity

54 opacity mask See opacity mask

55 shape See geometry

56 stroke brush 71

57 stroke control See stroke

58 transformation See transformation

59 usage described 74

60 payload **10**, See XPS Document format

61 physical imageable size **10**

62 physical media size **10**

63 physical model See Open Packaging Conventions

64 physical organization See Open Packaging Conventions,

65 interleaving; ZIP

66 pixel See layout, pixel

67 pixel snapping See stroke, line

68 PNG See image, PNG

69 positioning content See layout; transformation, matrix

70 primary fixed payload root **10**

71 printing

72 bleed area See page, bleed area

73 content area See page, content area

74 device fonts See font, device font

75 digital signature See digital signature

76 discard control See discard control

77 font, print and preview restricted . See font, licensing rights

78 interleaving . See Open Packaging Conventions, interleaving

79 layout See layout

80 orientation See PrintTicket keywords

81 PrintTicket See PrintTicket; PrintTicket keywords

82 resource constraints See discard control

83 scaling See PrintTicket keywords

84 PrintTicket **11**

85 empty PrintTicket, markup of 251

86 mapping content levels to parts 40

87 parts See parts

88 relationships See relationships

89 producer **11**

90 bleed size **11**

91 content size **11**

92 implementation burden 413–67

93 media size **11**

94 property See XPS Document format, properties

95 **R**

96 radial gradient See brush, radial gradient brush

97 raster graphics See image

98 reading order See accessibility, reading order

99 relationship **11**

100 StartPart **12**

101 relationships See also XPS Document format

102 core properties 26

103 digital signature 26

104 digital signature certificate 26

105 digital signature definitions 26

106 digital signature origin 26

107 DiscardControl 26

108 DocumentStructure 26

1 end71

2 flat289

3 for dashed stroke289

4 round.....289

5 square289

6 start.....71

7 triangle289

8 join

9 bevel.....291

10 miter.....292–94

11 round.....290

12 style.....71

13 miter limit71

14 pixel snapping71

15 thin stroke anti-aliasing behavior267

16 zero-length, avoided83

17 multi-figure path, behavior with295

18 phase control285

19 segments, mixed stroked and non-stroked295

20 thickness71

21 **T**

22 table *See* document structure, table, *See* document structure,

23 table

24 table of contents..... *See* document structure, outline

25 text

26 baseline *See* text, glyph, baseline

27 bidirectional 100, 115

28 bold *See* text, style simulation

29 clipping *See* clipping

30 fill brush 100, 123

31 font..... *See also* font

32 device font, reference to 100, 118

33 reference to 100

34 glyph

35 advance width 105, 106, 110, 111

36 baseline.....105

37 black box.....105

38 cluster map 106–10, 110, 111, 112, 438

39 indices

40 described 110

41 reference to non-existent glyph 110

42 restriction of length.....110

43 specifying 100

44 syntax 110

45 metrics 105

46 offset..... 106, 110, 111, 115, 438

47 origin.....105, 106

48 origin, sideways105

49 side-bearing, bottom106

50 side-bearing, left.....105

51 side-bearing, right.....105

52 side-bearing, top.....106

53 glyphs usage for99

54 italic..... *See* text, style simulation

55 markup 100

56 markup examples 120

57 markup optimization 119

58 opacity *See* opacity

59 opacity mask..... *See* opacity mask

60 position..... *See also* text, glyph, offset

61 horizontal text 100

62 vertical text 113

63 sideways (vertical) 100

64 advance width 110, 114

65 bidirectional text, intersection with 114

66 described..... 113

67 example, sideways 116

68 example, vertical 117

69 horizontal text, including 113

70 origin calculation 114

71 vertical glyphs, preference for 113

72 size 100

73 style simulation..... 100, 112

74 transformation..... *See* transformation

75 underline *See* path

76 Unicode string..... *See also* text, glyph

77 escaping open brace character 112

78 mapping code units to glyphs 106

79 normalization prohibited 112

80 specifying 100

81 Unicode control marks, inclusion of..... 112

82 Unicode scalar value, split into code units 106

83 unmappable code unit behavior 112

84 usage of 112

85 UTF-16 code units, consisting of 106

86 vertical *See* text, sideways (vertical)

87 thread..... *See* document structure, story

88 thumbnail 12

89 described 24, 34

90 formats *See* image

91 parts..... *See* parts

92 relationship..... 27

93 usage..... 34

94 TIFF *See* image, TIFF

95 transformation

96 composability 266

97 described 181, 266

98 effective transform..... *See* transformation, composability

99 elements applied to 169, 170

100 matrix

101 abbreviated syntax 182

102 abbreviated syntax example 185

103 described 181, 182

104 example 184

105 inverting x-axis 183

106 inverting y-axis 183

107 multiplying 182

108 positioning..... 183

109 rotating..... 183

110 scaling..... 183

111 skewing 183

1	specifying	182	50	element syntax	48
2	mentioned	265	51	model	47
3	non-invertible transform, rendering of elements with ...	266	52	ordering	48, 424
4	of brush	125	53	property attribute	11
5	of canvas	63, 185	54	property element	11
6	of geometry	79, 188	55	property value	11, 47
7	of glyphs	100, 187	56	property	11
8	of image brush	128, 189	57	relationship types	413
9	of linear gradient brush	151, 193	58	relationships	21, 26
10	of path	71, 186	59	RELAX NG schema	
11	of radial gradient brush	157, 194	60	XPS Document	391
12	of tiles	190	61	RELAX NG schema	
13	of tiles, example	192	62	signature definitions	389
14	of visual brush	131, 190	63	RELAX NG schema	
15	transparency	See opacity	64	resource dictionary key	393
16	TrueType collection (TTC) font	See font:TrueType collection	65	RELAX NG schema	
17	(TTC)		66	document structure	395
18	TrueType font	See font:TrueType	67	RELAX NG schema	
<hr/>					
19	V		68	discard control	397
20	vector graphics	See path	69	RELAX NG schema	
21	versioning	See markup compatibility	70	3D-Graphic Content	398
<hr/>					
22	W		71	root	51
23	whitespace	See XPS Document format, XML, whitespace	72	versioning	See markup compatibility
24	Windows Color System (WCS)	See color, color profile,	73	XML	
25	Windows Color System (WCS) profile		74	DTDs prohibited	46
26	Windows Media Photo	See image, Windows Media Photo	75	markup design	45
<hr/>					
27	X		76	markup model	47–49
28	XML	See XPS Document format, XML	77	namespaces	47, 411
29	XML namespaces	See namespace; XPS Document format,	78	Unicode encodings permitted	46
30	XML, namespaces		79	usage	46
31	XML Paper Specification		80	whitespace	49
32	document format	See XPS Document format	81	XML and XSI namespace usage	47
33	organization of	19	82	XML schema (XSD)	
34	XPS Document format	12, 19, 23	83	3D-Graphic Content	386
35	content types	412	84	characteristics	47
36	described	1, 51	85	discard control	385
37	example	25	86	document structure	379
38	extensibility	See markup compatibility	87	resource dictionary key	377
39	fixed payload	9, 23	88	signature definitions	351
40	fixed payload root	10, 23	89	validity requirement	47
41	illustrated	21	90	XPS Document	353
42	language	See language	91	XPS elements	
43	markup elements	See XPS elements	92	digital signature	
44	parts	21, 23	93	<Intent>	263, 317
45	payload	23	94	<SignatureDefinition>	261, 339
46	properties		95	<SignatureDefinitions>	261, 340
47	attribute syntax	48	96	<SignBy>	264, 339
48	composability	47, 169	97	<SigningLocation>	264, 340
49	described	47	98	<SpotLocation>	262, 341
			99	discard control	
			100	<Discard>	255, 303
			101	<DiscardControl>	255, 304
			102	document structure	
			103	<DocumentOutline>	225, 304
			104	<DocumentStructure.Outline>	225, 305
			105	<DocumentStructure>	224, 305
			106	<FigureStructure>	242, 306

1	<ListItemStructure>	242, 320	38	<GradientStop>	165, 314
2	<ListStructure>	241, 320	39	<ImageBrush.Transform>	189, 316
3	<NamedElement>	243, 321	40	<ImageBrush>	127, 315
4	<OutlineEntry>	225, 322	41	<LinearGradientBrush.GradientStops>	156, 319
5	<ParagraphStructure>	239, 323	42	<LinearGradientBrush.Transform>	193, 319
6	<SectionStructure>	239, 338	43	<LinearGradientBrush>	151, 317
7	<Story>	227, 341	44	<MatrixTransform>	182, 321
8	<StoryBreak>	239, 342	45	<Path.Clip>	179, 329
9	<StoryFragment>	235, 342	46	<Path.Data>	74, 329
10	<StoryFragmentReference>	228, 344	47	<Path.Fill>	75, 329
11	<StoryFragments>	233, 343	48	<Path.OpacityMask>	198, 329
12	<TableCellStructure>	241, 344	49	<Path.RenderTransform>	186, 330
13	<TableRowGroupStructure>	240, 345	50	<Path.Stroke>	76, 330
14	<TableRowStructure>	240, 345	51	<Path>	70, 324
15	<TableStructure>	240, 346	52	<PathFigure>	82, 331
16	document-level		53	<PathGeometry.Transform>	188, 332
17	<DocumentReference>	51, 305	54	<PathGeometry>	79, 331
18	<FixedDocument>	52, 306	55	<PolyBezierSegment>	87, 333
19	<FixedDocumentSequence>	51, 306	56	<PolyLineSegment>	88, 333
20	<LinkTarget>	54, 319	57	<PolyQuadraticBezierSegment>	89, 334
21	<PageContent.LinkTargets>	53, 323	58	<RadialGradientBrush.GradientStops>	164, 337
22	<PageContent>	52, 322	59	<RadialGradientBrush.Transform>	194, 337
23	page-level		60	<RadialGradientBrush>	157, 334
24	<ArcSegment>	83, 299	61	<ResourceDictionary>	173, 337
25	<Brush3D>	472	62	<SolidColorBrush>	126, 340
26	<Canvas.Clip>	178, 302	63	<VisualBrush.Transform>	190, 348
27	<Canvas.OpacityMask>	197, 302	64	<VisualBrush.Visual>	132, 349
28	<Canvas.RenderTransform>	185, 303	65	<VisualBrush>	131, 346
29	<Canvas.Resources>	172, 303			
30	<Canvas>	63, 300			
31	<FixedPage.Resources>	171, 308	66	Z	
32	<FixedPage>	55, 306	67	ZIP	
33	<Glyphs.Clip>	180, 313	68	archive	12, 21
34	<Glyphs.Fill>	123, 313	69	item	12
35	<Glyphs.OpacityMask>	199, 313	70	utilities	37
36	<Glyphs.RenderTransform>	187, 314			
37	<Glyphs>	100, 308			