XML Paper Specification

XPS Specification and Reference Guide

Working Draft 1.1

February 2008

This document is "a work in progress" and was produced by Ecma Technical Committee TC46.
# Contents

1. **Scope** .................................................................................................................. 1

2. **Conformance** ........................................................................................................ 3

3. **Language Notes** ................................................................................................... 3

4. **Software Conformance** .......................................................................................... 3

5. **Normative References** ........................................................................................... 5

4. **Definitions** .............................................................................................................. 7

5. **Notational Conventions** .......................................................................................... 11

6. **Document Conventions** .......................................................................................... 11

7. **Diagrams** ................................................................................................................ 11

6. **Acronyms and Abbreviations** .................................................................................. 13

7. **GENERAL DESCRIPTION** .................................................................................... 15

8. **XPS DOCUMENT FORMAT** .................................................................................... 17

9. **How This Specification Is Organized** ...................................................................... 17

8. **Package** .................................................................................................................. 19

9. **PARTS AND RELATIONSHIPS** ............................................................................. 21

9.1 **Fixed Payload** ..................................................................................................... 21

9.1.1 **Fixed Payload Relationships** ........................................................................... 24

9.1.2 **FixedDocumentSequence Part** ......................................................................... 25

9.1.3 **FixedDocument Part** ....................................................................................... 25

9.1.4 **FixedPage Part** ............................................................................................... 26

9.1.5 **Image Parts** ..................................................................................................... 26

9.1.6 **Thumbnail Parts** ............................................................................................. 26

9.1.7 **Font Parts** ....................................................................................................... 32

9.1.8 **Remote Resource Dictionary Parts** ................................................................. 37

9.1.9 **PrintTicket Parts** ............................................................................................. 37

9.1.10 **SignatureDefinitions Part** .............................................................................. 39

9.1.11 **DocumentStructure Part** ............................................................................... 40

9.1.12 **StoryFragments Part** ..................................................................................... 40

9.2 **Part Naming Recommendations** ........................................................................... 41

9.3 **XPS Document Markup** ....................................................................................... 43

9.3.1 **Support for Versioning and Extensibility** ......................................................... 44

9.3.2 **XML Usage** .................................................................................................... 44

9.3.3 **Markup Model** ............................................................................................... 47

9.3.4 **Whitespace** ..................................................................................................... 47

9.3.5 **Language** ....................................................................................................... 47

10. **DOCUMENTS** ....................................................................................................... 49

10.1 **<FixedDocumentSequence> Element** ................................................................. 49

10.1.1 **<DocumentReference> Element** .................................................................... 49

10.2 **<FixedDocument> Element** ................................................................................ 50

10.2.1 **<PageContent> Element** ............................................................................... 50

10.2.2 **<PageContent.LinkTargets> Element** ............................................................ 51

10.2.3 **<LinkTarget> Element** .................................................................................. 52

10.3 **<FixedPage> Element** ........................................................................................ 53

10.3.1 **BleedBox Attribute** ....................................................................................... 54
Contents

10.3.2 ContentBox Attribute ................................................................. 55
10.3.3 Page Size Terminology .................................................................... 55
10.3.4 Media Orientation and Scaling ...................................................... 56
10.4 <Canvas> Element .............................................................................. 61
10.5 <Path> Element ................................................................................... 64
10.6 <Glyphs> Element ................................................................................. 65
11. GRAPHICS ............................................................................................... 67
11.1 <Path> Element ................................................................................... 68
11.1.1 <Path.Data> Element ....................................................................... 72
11.1.2 <Path.Fill> Element ....................................................................... 72
11.1.3 <Path.Stroke> Element .................................................................... 74
11.2 Geometries and Figures ......................................................................... 75
11.2.1 Geometries ....................................................................................... 77
11.2.2 Figures ............................................................................................. 80
11.2.3 Abbreviated Geometry Syntax ......................................................... 89
12. TEXT ......................................................................................................... 97
12.1 <Glyphs> Element ................................................................................. 98
12.1.1 Glyph Metrics .................................................................................. 103
12.1.2 Mapping Code Units to Glyphs ......................................................... 104
12.1.3 Indices Attribute ............................................................................. 108
12.1.4 UnicodeString Attribute ................................................................. 110
12.1.5 StyleSimulations Attribute .............................................................. 110
12.1.6 IsSideways Attribute ....................................................................... 111
12.1.7 DeviceFontName Attribute .............................................................. 118
12.1.8 xml:lang Attribute .......................................................................... 118
12.1.9 CaretStops Attribute ....................................................................... 119
12.1.10 Optimizing Glyph Markup ............................................................ 119
12.1.11 Glyph Markup Examples ............................................................... 121
12.2 <Glyphs.Fill> Element ....................................................................... 124
13. BRUSHES ................................................................................................ 125
13.1 <SolidColorBrush> Element ................................................................. 126
13.2 <ImageBrush> Element ....................................................................... 127
13.3 <VisualBrush> Element ...................................................................... 131
13.3.1 <VisualBrush.Visual> Element .................................................... 132
13.4 Common Attributes for Tiling Brushes ............................................... 136
13.4.1 Viewbox, Viewport, ViewboxUnits, and ViewportUnits Attributes ................................................. 136
13.4.2 TileMode Attribute ......................................................................... 141
13.5 <LinearGradientBrush> Element ......................................................... 152
13.5.1 SpreadMethod Attribute ................................................................. 154
13.5.2 <LinearGradientBrush.GradientStops> Element ................................................. 157
13.6 <RadialGradientBrush> Element ......................................................... 158
13.6.1 SpreadMethod Attribute ................................................................. 162
13.6.2 <RadialGradientBrush.GradientStops> Element ................................................. 165
13.7 <GradientStop> Element .................................................................... 166
13.8 Using a Brush as an Opacity Mask ....................................................... 167
14. COMMON PROPERTIES ........................................................................... 171
14.1 Opacity .................................................................................................. 172
14.2 Resources and Resource References ................................................... 172
14.2.1 <FixedPage.Resources> Element .................................................... 173
14.2.2 <Canvas.Resources> Element ........................................................ 175
### Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>14.2.3</td>
<td><code>&lt;ResourceDictionary&gt;</code> Element</td>
<td>176</td>
</tr>
<tr>
<td>14.2.4</td>
<td>Resource References</td>
<td>180</td>
</tr>
<tr>
<td>14.2.5</td>
<td>Scoping Rules for Resolving Resource References</td>
<td>180</td>
</tr>
<tr>
<td>14.2.6</td>
<td>Support for Markup Compatibility</td>
<td>182</td>
</tr>
<tr>
<td>14.3</td>
<td>Clipping</td>
<td>183</td>
</tr>
<tr>
<td>14.3.1</td>
<td><code>&lt;Canvas.Clip&gt;</code> Element</td>
<td>183</td>
</tr>
<tr>
<td>14.3.2</td>
<td><code>&lt;Path.Clip&gt;</code> Element</td>
<td>184</td>
</tr>
<tr>
<td>14.3.3</td>
<td><code>&lt;Glyphs.Clip&gt;</code> Element</td>
<td>185</td>
</tr>
<tr>
<td>14.4</td>
<td>Positioning Content</td>
<td>186</td>
</tr>
<tr>
<td>14.4.1</td>
<td><code>&lt;MatrixTransform&gt;</code> Element</td>
<td>186</td>
</tr>
<tr>
<td>14.4.2</td>
<td><code>&lt;Canvas.RenderTransform&gt;</code> Element</td>
<td>191</td>
</tr>
<tr>
<td>14.4.3</td>
<td><code>&lt;Path.RenderTransform&gt;</code> Element</td>
<td>192</td>
</tr>
<tr>
<td>14.4.4</td>
<td><code>&lt;Glyphs.RenderTransform&gt;</code> Element</td>
<td>193</td>
</tr>
<tr>
<td>14.4.5</td>
<td><code>&lt;Geometry.Transform&gt;</code> Element</td>
<td>194</td>
</tr>
<tr>
<td>14.4.6</td>
<td><code>&lt;ImageBrush.Transform&gt;</code> Element</td>
<td>196</td>
</tr>
<tr>
<td>14.4.7</td>
<td><code>&lt;VisualBrush.Transform&gt;</code> Element</td>
<td>197</td>
</tr>
<tr>
<td>14.4.8</td>
<td><code>&lt;LinearGradientBrush.Transform&gt;</code> Element</td>
<td>200</td>
</tr>
<tr>
<td>14.4.9</td>
<td><code>&lt;RadialGradientBrush.Transform&gt;</code> Element</td>
<td>201</td>
</tr>
<tr>
<td>14.5</td>
<td>OpacityMask</td>
<td>204</td>
</tr>
<tr>
<td>14.5.1</td>
<td><code>&lt;Canvas.OpacityMask&gt;</code> Element</td>
<td>204</td>
</tr>
<tr>
<td>14.5.2</td>
<td><code>&lt;Path.OpacityMask&gt;</code> Element</td>
<td>205</td>
</tr>
<tr>
<td>14.5.3</td>
<td><code>&lt;Glyphs.OpacityMask&gt;</code> Element</td>
<td>206</td>
</tr>
<tr>
<td>15</td>
<td>Color</td>
<td>209</td>
</tr>
<tr>
<td>15.1</td>
<td>Color Support</td>
<td>209</td>
</tr>
<tr>
<td>15.1.1</td>
<td>sRGB Color Space</td>
<td>209</td>
</tr>
<tr>
<td>15.1.2</td>
<td>scRGB Color Space</td>
<td>209</td>
</tr>
<tr>
<td>15.1.3</td>
<td>Gray Color Space</td>
<td>210</td>
</tr>
<tr>
<td>15.1.4</td>
<td>CMYK Color Space</td>
<td>210</td>
</tr>
<tr>
<td>15.1.5</td>
<td>N-Channel Color Spaces</td>
<td>210</td>
</tr>
<tr>
<td>15.1.6</td>
<td>Named Color for Spot Colors and N-tone Images</td>
<td>210</td>
</tr>
<tr>
<td>15.1.7</td>
<td>Device Color Spaces</td>
<td>210</td>
</tr>
<tr>
<td>15.1.8</td>
<td>ICC Profiles</td>
<td>210</td>
</tr>
<tr>
<td>15.1.9</td>
<td>WcsProfilesTag</td>
<td>211</td>
</tr>
<tr>
<td>15.1.10</td>
<td>WCS Color Profiles</td>
<td>212</td>
</tr>
<tr>
<td>15.1.11</td>
<td>Vector Color Syntax</td>
<td>212</td>
</tr>
<tr>
<td>15.1.12</td>
<td>sRGB Color Syntax</td>
<td>213</td>
</tr>
<tr>
<td>15.1.13</td>
<td>scRGB Color Syntax</td>
<td>214</td>
</tr>
<tr>
<td>15.1.14</td>
<td>CMYK Color Syntax</td>
<td>215</td>
</tr>
<tr>
<td>15.1.15</td>
<td>N-Channel Color Syntax</td>
<td>215</td>
</tr>
<tr>
<td>15.1.16</td>
<td>Named Color Syntax</td>
<td>216</td>
</tr>
<tr>
<td>15.2</td>
<td>Rich Colors in Raster Images</td>
<td>217</td>
</tr>
<tr>
<td>15.2.1</td>
<td>sRGB Raster Images</td>
<td>217</td>
</tr>
<tr>
<td>15.2.2</td>
<td>scRGB Raster Images</td>
<td>218</td>
</tr>
<tr>
<td>15.2.3</td>
<td>Gray Raster Images</td>
<td>218</td>
</tr>
<tr>
<td>15.2.4</td>
<td>CMYK Raster Images</td>
<td>218</td>
</tr>
<tr>
<td>15.2.5</td>
<td>N-channel Raster Images</td>
<td>219</td>
</tr>
<tr>
<td>15.2.6</td>
<td>Named Color Raster Images</td>
<td>219</td>
</tr>
<tr>
<td>15.2.7</td>
<td>Device Color Raster Images</td>
<td>220</td>
</tr>
<tr>
<td>15.2.8</td>
<td>Images and Color Profile Association</td>
<td>220</td>
</tr>
<tr>
<td>15.2.9</td>
<td>Color Space Pixel Formats for Raster Images</td>
<td>220</td>
</tr>
<tr>
<td>15.3</td>
<td>Color Separation</td>
<td>221</td>
</tr>
<tr>
<td>15.4</td>
<td>Alpha and Gradient Blending with Rich Colors</td>
<td>221</td>
</tr>
<tr>
<td>15.5</td>
<td>PrintTicket Color Settings</td>
<td>222</td>
</tr>
<tr>
<td>16</td>
<td>Document Structure and Interactivity</td>
<td>227</td>
</tr>
<tr>
<td>16.1</td>
<td>Document Structure Markup</td>
<td>227</td>
</tr>
<tr>
<td>16.1.1</td>
<td>DocumentStructure Part</td>
<td>227</td>
</tr>
<tr>
<td>16.1.2</td>
<td>StoryFragments Part</td>
<td>233</td>
</tr>
</tbody>
</table>
Contents

1.1 Consistent Nominal Stroke Width ................................................................. 263
1.2 Rules for Degenerate Line and Curve Segments .............................................. 263
1.3 Optimization Guidelines ................................................................................. 264
1.4 Composition Examples .................................................................................... 265
2.6 Stroke Rendering ............................................................................................. 268
2.6.1 Stroke Edge Parallelization ....................................................................... 268
2.6.2 Phase Control .............................................................................................. 268
2.6.3 Symmetry of Stroke Drawing Algorithms ................................................ 269
2.6.4 Rules for Dash Cap Rendering .................................................................. 269
2.6.5 Rules for Line Cap Rendering ................................................................... 270
2.6.6 Line Caps for Dashed Strokes .................................................................... 270
2.6.7 Rules for Line Join Rendering .................................................................... 271
2.6.8 Rules for Degenerate Line and Curve Segments ........................................ 271
2.6.9 Stroking and Fill Rule ................................................................................ 271
2.6.10 Mixing Stroked and Non-Stroked Segments ........................................... 272
2.6.11 Stroke Behavior with Multiple Path Figures ............................................. 272
2.6.12 Consistent Nominal Stroke Width ............................................................ 272
3.7 Brushes and Images ........................................................................................ 275
3.7.1 Small Tiles ................................................................................................. 275

XPS Specification and Reference Guide

16.2 Hyperlinks ........................................................................................................ 247
16.2.1 Hyperlink Activation ............................................................................... 247
16.2.2 Hyperlink Addressing ............................................................................. 248
16.2.3 Name Attribute ....................................................................................... 248
16.2.4 FixedPage.NavigateUri Attribute ......................................................... 249
16.3 Selection .......................................................................................................... 250
16.4 Accessibility .................................................................................................... 250
16.4.1 Reading Order ......................................................................................... 250
16.4.2 Screen Reader Applications .................................................................... 251
16.4.3 Text Alternatives for Graphics and Images ............................................. 251

17. XPS DOCUMENT PACKAGE FEATURES ....................................................... 253
17.1 Interleaving Optimizations ........................................................................... 253
17.1.1 Empty PrintTicket .................................................................................. 254
17.1.2 Optimizing Interleaving Order ............................................................... 254
17.1.3 Consuming Interleaved Packages .......................................................... 257
17.1.4 Consumers with Resource Constraints .................................................. 257
17.1.5 Interleaving Optimizations and Digital Signatures .................................. 259
17.2 Digital Signatures .......................................................................................... 259
17.2.1 Signature Policy ........................................................................................ 260
17.2.2 Signature Definitions ............................................................................... 263

18. RENDERING RULES ......................................................................................... 269
18.1 Coordinate System and Rendering Placement ............................................. 269
18.1.1 Page Dimensions ..................................................................................... 269
18.1.2 Rounding of Coordinates ....................................................................... 269
18.1.3 Transforms ............................................................................................... 270
18.1.4 Pixel Center Location, Pixel Placement, and Pixel Inclusion .................. 270
18.1.5 Maximum Placement Error ..................................................................... 271
18.1.6 Pixel Placement for Glyphs ...................................................................... 271
18.1.7 Abutment of Shapes ............................................................................... 271
18.1.8 Clipping Behavior ................................................................................... 271
18.2 Implementation Limits ................................................................................... 272
18.3 Gradient Computations .................................................................................. 273
18.3.1 All Gradients ........................................................................................... 273
18.3.2 Linear Gradients ...................................................................................... 275
18.3.3 Radial Gradients ..................................................................................... 277
18.4 Opacity Computations ................................................................................... 280
18.4.1 Pre-Multiplied Alpha and Superluminous Colors .................................... 282
18.5 Composition Rules ........................................................................................ 283
18.5.1 Optimization Guidelines ......................................................................... 284
18.5.2 Composition Examples ............................................................................ 285
18.6 Stroke Rendering ............................................................................................ 289
18.6.1 Stroke Edge Parallelization ..................................................................... 289
18.6.2 Phase Control ........................................................................................... 289
18.6.3 Symmetry of Stroke Drawing Algorithms .............................................. 289
18.6.4 Rules for Dash Cap Rendering ............................................................... 290
18.6.5 Rules for Line Cap Rendering ............................................................... 293
18.6.6 Line Caps for Dashed Strokes ............................................................... 293
18.6.7 Rules for Line Join Rendering ............................................................... 294
18.6.8 Rules for Degenerate Line and Curve Segments ..................................... 298
18.6.9 Stroking and Fill Rule ............................................................................. 299
18.6.10 Mixing Stroked and Non-Stroked Segments ........................................ 299
18.6.11 Stroke Behavior with Multiple Path Figures ......................................... 299
18.6.12 Consistent Nominal Stroke Width ........................................................ 299
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>19.38 ParagraphStructure ................................................................. 327</td>
</tr>
<tr>
<td>2</td>
<td>19.39 Path .................................................................................. 328</td>
</tr>
<tr>
<td>3</td>
<td>19.40 Path.Clip ........................................................................... 333</td>
</tr>
<tr>
<td>4</td>
<td>19.41 Path.Data ........................................................................... 333</td>
</tr>
<tr>
<td>5</td>
<td>19.42 Path.Fill ........................................................................... 333</td>
</tr>
<tr>
<td>6</td>
<td>19.43 Path.OpacityMask ................................................................. 333</td>
</tr>
<tr>
<td>7</td>
<td>19.44 Path.RenderTransform ............................................................ 334</td>
</tr>
<tr>
<td>8</td>
<td>19.45 Path.Stroke ........................................................................ 334</td>
</tr>
<tr>
<td>9</td>
<td>19.46 PathFigure ....................................................................... 335</td>
</tr>
<tr>
<td>10</td>
<td>19.47 PathGeometry ................................................................. 335</td>
</tr>
<tr>
<td>11</td>
<td>19.48 PathGeometry.Transform ..................................................... 336</td>
</tr>
<tr>
<td>12</td>
<td>19.49 PolyBezierSegment ............................................................ 337</td>
</tr>
<tr>
<td>13</td>
<td>19.50 PolyLineSegment .............................................................. 337</td>
</tr>
<tr>
<td>14</td>
<td>19.51 PolyQuadraticBezierSegment .................................................. 338</td>
</tr>
<tr>
<td>15</td>
<td>19.52 RadialGradientBrush .......................................................... 338</td>
</tr>
<tr>
<td>16</td>
<td>19.53 RadialGradientBrush.GradientStops ............................................. 341</td>
</tr>
<tr>
<td>17</td>
<td>19.54 RadialGradientBrush.Transform ............................................. 341</td>
</tr>
<tr>
<td>18</td>
<td>19.55 ResourceDictionary .............................................................. 341</td>
</tr>
<tr>
<td>19</td>
<td>19.56 SectionStructure ................................................................. 342</td>
</tr>
<tr>
<td>20</td>
<td>19.57 SignBy ........................................................................... 343</td>
</tr>
<tr>
<td>21</td>
<td>19.58 SignatureDefinition .............................................................. 343</td>
</tr>
<tr>
<td>22</td>
<td>19.59 SignatureDefinitions ............................................................ 344</td>
</tr>
<tr>
<td>23</td>
<td>19.60 SigningLocation ................................................................. 344</td>
</tr>
<tr>
<td>24</td>
<td>19.61 SolidColorBrush ................................................................. 344</td>
</tr>
<tr>
<td>25</td>
<td>19.62 SpotLocation ................................................................. 345</td>
</tr>
<tr>
<td>26</td>
<td>19.63 Story ........................................................................... 345</td>
</tr>
<tr>
<td>27</td>
<td>19.64 StoryBreak ................................................................... 346</td>
</tr>
<tr>
<td>28</td>
<td>19.65 StoryFragment .................................................................. 346</td>
</tr>
<tr>
<td>29</td>
<td>19.66 StoryFragments ................................................................. 347</td>
</tr>
<tr>
<td>30</td>
<td>19.67 StoryFragmentReference ...................................................... 348</td>
</tr>
<tr>
<td>31</td>
<td>19.68 TableCellStructure ............................................................. 348</td>
</tr>
<tr>
<td>32</td>
<td>19.69 TableRowGroupStructure ....................................................... 349</td>
</tr>
<tr>
<td>33</td>
<td>19.70 TableRowStructure ............................................................. 349</td>
</tr>
<tr>
<td>34</td>
<td>19.71 TableStructure ................................................................. 350</td>
</tr>
<tr>
<td>35</td>
<td>19.72 VisualBrush ................................................................... 350</td>
</tr>
<tr>
<td>36</td>
<td>19.73 VisualBrush.Transform ........................................................ 352</td>
</tr>
<tr>
<td>37</td>
<td>19.74 VisualBrush.Visual .......................................................... 353</td>
</tr>
</tbody>
</table>
# Contents

1. **A. Signature Definitions W3C Schema** ................................................................. 355
2. **B. XPS Document W3C Schema** ........................................................................ 357
3. **C. Resource Dictionary Key W3C Schema** ......................................................... 381
4. **D. Document Structure W3C Schema** ............................................................... 383
5. **E. Discard Control W3C Schema** ....................................................................... 389
6. **F. Abbreviated Geometry Syntax Algorithm** ....................................................... 391
7. **G. scRGB Gamut Boundary Definition** .............................................................. 397
8. **H. Standard Namespaces and Content Types** ..................................................... 403
9. **H.1 XML Namespace URIs** ................................................................................. 403
10. **H.2 Content Types** .......................................................................................... 404
11. **H.3 Relationship Types** .................................................................................... 405

## I. Conformance Requirements

12. **I.1 XPS Document Format** ................................................................................ 408
13. **I.1.1 MUST Conformance Requirements** ......................................................... 408
14. **I.2 Parts and Relationships** ................................................................................ 408
15. **I.2.1 MUST Conformance Requirements** ......................................................... 408
16. **I.2.2 SHOULD Conformance Requirements** ...................................................... 414
17. **I.2.3 OPTIONAL Conformance Requirements** ................................................... 418
18. **I.3 Documents** .................................................................................................. 420
19. **I.3.1 MUST Conformance Requirements** .......................................................... 420
20. **I.3.2 SHOULD Conformance Requirements** ....................................................... 421
21. **I.4 Graphics** ...................................................................................................... 422
22. **I.4.1 MUST Conformance Requirements** ........................................................... 422
23. **I.4.2 SHOULD Conformance Requirements** ....................................................... 422
24. **I.4.3 OPTIONAL Conformance Requirements** .................................................... 422
25. **I.5 Text** .............................................................................................................. 423
26. **I.5.1 MUST Conformance Requirements** ........................................................... 423
27. **I.5.2 SHOULD Conformance Requirements** ....................................................... 425
28. **I.5.3 OPTIONAL Conformance Requirements** ..................................................... 425
29. **I.6 Brushes** ....................................................................................................... 426
30. **I.6.1 MUST Conformance Requirements** .......................................................... 426
31. **I.7 Common Properties** ..................................................................................... 426
32. **I.7.1 MUST Conformance Requirements** ........................................................... 426
33. **I.7.2 OPTIONAL Conformance Requirements** ..................................................... 427
34. **I.8 Color** ............................................................................................................. 428
35. **I.8.1 MUST Conformance Requirements** ........................................................... 428
36. **I.8.2 SHOULD Conformance Requirements** ....................................................... 430
37. **I.8.3 OPTIONAL Conformance Requirements** ..................................................... 431
38. **I.9 Document Structure and Interactivity** .......................................................... 432
39. **I.9.1 MUST Conformance Requirements** ........................................................... 432
40. **I.9.2 SHOULD Conformance Requirements** ....................................................... 433
41. **I.9.3 OPTIONAL Conformance Requirements** ..................................................... 435
42. **I.10 XPS Document Package Features** ............................................................. 436
43. **I.10.1 MUST Conformance Requirements** .......................................................... 436
44. **I.10.2 SHOULD Conformance Requirements** ....................................................... 438
45. **I.10.3 OPTIONAL Conformance Requirements** ..................................................... 440
46. **I.11 Rendering Rules** .......................................................................................... 441

Working Draft 1.1, February 2008
Contents

1.11.1 MUST Conformance Requirements .......................................................... 441
1.11.2 SHOULD Conformance Requirements ..................................................... 442
1.11.3 OPTIONAL Conformance Requirements .................................................. 445

I.12 Additional Conformance Requirements ...................................................... 447
  I.12.1 MUST Conformance Requirements ..................................................... 447

J. BIBLIOGRAPHY ............................................................................................... 449

K. INDEX ............................................................................................................ 451

x Working Draft 1.1, February 2008
List of Figures

1. Package-based XPS Document format .......................................................... 19
2. Page regions .................................................................................................. 56
3. Matching PrintTicket and fixed page size and orientation ......................... 57
4. Matching PrintTicket and fixed page size with differing orientation ........... 57
5. Matching PrintTicket and fixed page orientation with differing size .......... 58
6. Fill using EvenOdd algorithm ..................................................................... 79
7. Fill using NonZero algorithm ..................................................................... 79
8. Arc choice A ............................................................................................... 83
9. Arc choice B ............................................................................................... 83
10. Arc choice C ............................................................................................ 83
11. Arc choice D ............................................................................................. 83
12. Glyph metrics .......................................................................................... 103
13. Upright (usually horizontal) glyph metrics ............................................... 103
14. Sideways (usually vertical) glyph metrics ................................................ 104
15. A sample signature spot ........................................................................... 266
16. Extreme curvatures and dash rendering .................................................. 289
17. Flat dash caps .......................................................................................... 290
18. Square dash caps ...................................................................................... 290
19. Round dash caps ........................................................................................ 291
20. Triangular dash caps .................................................................................. 292
21. Overlapping dash segments ..................................................................... 292
22. Flat start line cap, flat end line cap ........................................................... 293
23. Square start line cap, square end line cap ................................................ 293
24. Triangular start line cap, triangular end line cap ....................................... 293
25. Round start line cap, round end line cap ................................................ 293
26. Stroke start or end point within a dash ...................................................... 293
27. Stroke start or end point within a gap ....................................................... 294
28. Round line join with right angle ............................................................... 294
29. Round line join with acute angle .............................................................. 294
30. Round line join with obtuse angle ............................................................ 295
31. Beveled line join with right angle .............................................................. 295
32. Beveled line join with acute angle ............................................................ 295
33. Beveled line join with obtuse angle .......................................................... 296
34. Mitered line join with right angle and miter limit of 1.0 ............................. 296
35. Mitered line join with acute angle and miter limit of 1.0 ............................ 297
36. Mitered line join with obtuse angle and miter limit of 1.0 ......................... 297
37. Mitered line join with right angle and miter limit of 2.0 ............................ 297
38. Mitered line join with acute angle and miter limit of 2.0 ......................... 298
39. Mitered line join with acute angle and miter limit of 10.0 ....................... 298
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>9–1</td>
<td>XPS Document parts</td>
<td>21</td>
</tr>
<tr>
<td>9–2</td>
<td>Fixed payload relationships</td>
<td>24</td>
</tr>
<tr>
<td>9–3</td>
<td>Supported JPEG APPn markers</td>
<td>27</td>
</tr>
<tr>
<td>9–4</td>
<td>Support for ancillary PNG chunks</td>
<td>27</td>
</tr>
<tr>
<td>9–5</td>
<td>Supported TIFF 6.0 tags</td>
<td>28</td>
</tr>
<tr>
<td>9–6</td>
<td>Supported Windows Media Photo features</td>
<td>32</td>
</tr>
<tr>
<td>9–7</td>
<td>Guidelines for OpenType font embedding</td>
<td>34</td>
</tr>
<tr>
<td>9–8</td>
<td>Cmap table selection</td>
<td>36</td>
</tr>
<tr>
<td>9–9</td>
<td>Arc segment definition</td>
<td>82</td>
</tr>
<tr>
<td>9–10</td>
<td>Commands</td>
<td>90</td>
</tr>
<tr>
<td>9–12</td>
<td>Glyph specifications</td>
<td>108</td>
</tr>
<tr>
<td>9–14</td>
<td>Portions of the cluster specification</td>
<td>109</td>
</tr>
<tr>
<td>9–15</td>
<td>IsSideways and BidiLevel effects on origin placement</td>
<td>113</td>
</tr>
<tr>
<td>9–16</td>
<td>Brush types</td>
<td>125</td>
</tr>
<tr>
<td>9–17</td>
<td>Common attributes for <code>&lt;ImageBrush&gt;</code> and <code>&lt;VisualBrush&gt;</code> elements</td>
<td>136</td>
</tr>
<tr>
<td>10–1</td>
<td>Common property attributes</td>
<td>171</td>
</tr>
<tr>
<td>10–2</td>
<td>Common property elements</td>
<td>172</td>
</tr>
<tr>
<td>15–1</td>
<td>WcsProfilesTagType structure</td>
<td>211</td>
</tr>
<tr>
<td>15–2</td>
<td>Syntax summary</td>
<td>213</td>
</tr>
<tr>
<td>15–3</td>
<td>Color Space Pixel Format Defaults</td>
<td>220</td>
</tr>
<tr>
<td>15–4</td>
<td>PrintTicket color settings</td>
<td>222</td>
</tr>
<tr>
<td>16–1</td>
<td>StoryFragments part elements</td>
<td>233</td>
</tr>
<tr>
<td>16–2</td>
<td>Unicode character categories</td>
<td>249</td>
</tr>
<tr>
<td>17–1</td>
<td>JobDigitalSignatureProcessing PrintTicket settings</td>
<td>262</td>
</tr>
<tr>
<td>17–2</td>
<td>Recommended minimum processing requirements</td>
<td>272</td>
</tr>
<tr>
<td>18–2</td>
<td>Opacity computation symbols</td>
<td>280</td>
</tr>
<tr>
<td>18–1</td>
<td>Package-wide namespaces</td>
<td>403</td>
</tr>
<tr>
<td>19–1</td>
<td>XPS Document namespaces</td>
<td>403</td>
</tr>
<tr>
<td>19–3</td>
<td>Package-wide content types</td>
<td>404</td>
</tr>
<tr>
<td>19–4</td>
<td>XPS Document content types</td>
<td>404</td>
</tr>
<tr>
<td>19–5</td>
<td>Package-wide relationship types</td>
<td>405</td>
</tr>
<tr>
<td>19–6</td>
<td>XPS Document relationship types</td>
<td>405</td>
</tr>
<tr>
<td>21–1</td>
<td>XPS Document format MUST conformance requirements</td>
<td>408</td>
</tr>
<tr>
<td>21–2</td>
<td>Parts and Relationships MUST conformance requirements</td>
<td>408</td>
</tr>
<tr>
<td>21–3</td>
<td>Parts and Relationships SHOULD conformance requirements</td>
<td>414</td>
</tr>
<tr>
<td>21–4</td>
<td>Parts and Relationships OPTIONAL conformance requirements</td>
<td>418</td>
</tr>
<tr>
<td>21–5</td>
<td>Document MUST conformance requirements</td>
<td>420</td>
</tr>
<tr>
<td>21–6</td>
<td>Document SHOULD conformance requirements</td>
<td>421</td>
</tr>
<tr>
<td>21–7</td>
<td>Graphics MUST conformance requirements</td>
<td>422</td>
</tr>
<tr>
<td>21–8</td>
<td>Graphics SHOULD conformance requirements</td>
<td>422</td>
</tr>
<tr>
<td>21–9</td>
<td>Graphics OPTIONAL conformance requirements</td>
<td>422</td>
</tr>
<tr>
<td>21–10</td>
<td>Text MUST conformance requirements</td>
<td>423</td>
</tr>
<tr>
<td>21–11</td>
<td>Text SHOULD conformance requirements</td>
<td>425</td>
</tr>
<tr>
<td>21–12</td>
<td>Text OPTIONAL conformance requirements</td>
<td>425</td>
</tr>
<tr>
<td>21–13</td>
<td>Brushes MUST conformance requirements</td>
<td>426</td>
</tr>
<tr>
<td>21–14</td>
<td>Common properties MUST conformance requirements</td>
<td>426</td>
</tr>
<tr>
<td>21–15</td>
<td>Common properties OPTIONAL conformance requirements</td>
<td>427</td>
</tr>
<tr>
<td>21–16</td>
<td>Color MUST conformance requirements</td>
<td>428</td>
</tr>
<tr>
<td>21–17</td>
<td>Color SHOULD conformance requirements</td>
<td>430</td>
</tr>
<tr>
<td>21–18</td>
<td>Color OPTIONAL conformance requirements</td>
<td>431</td>
</tr>
<tr>
<td>21–19</td>
<td>Document structure MUST conformance requirements</td>
<td>432</td>
</tr>
<tr>
<td>Table</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
<td>------</td>
</tr>
<tr>
<td>1-20</td>
<td>Document structure SHOULD conformance requirements</td>
<td>433</td>
</tr>
<tr>
<td>1-21</td>
<td>Document structure OPTIONAL conformance requirements</td>
<td>435</td>
</tr>
<tr>
<td>1-22</td>
<td>XPS Document package feature MUST conformance requirements</td>
<td>436</td>
</tr>
<tr>
<td>1-23</td>
<td>XPS Document package feature SHOULD conformance requirements</td>
<td>438</td>
</tr>
<tr>
<td>1-24</td>
<td>XPS Document package feature OPTIONAL conformance requirements</td>
<td>440</td>
</tr>
<tr>
<td>1-25</td>
<td>Rendering rules MUST conformance requirements</td>
<td>441</td>
</tr>
<tr>
<td>1-26</td>
<td>Rendering rules SHOULD conformance requirements</td>
<td>442</td>
</tr>
<tr>
<td>1-27</td>
<td>Rendering rules OPTIONAL conformance requirements</td>
<td>445</td>
</tr>
<tr>
<td>1-28</td>
<td>Additional MUST conformance requirements</td>
<td>447</td>
</tr>
</tbody>
</table>
List of Examples

Example 9–1. A typical XPS Document ................................................................. 23
Example 9–2. XPS Document part naming ........................................................... 42
Example 9–3. Property attribute syntax ............................................................... 46
Example 9–4. Property element syntax ............................................................... 47
Example 10–1. &lt;FixedDocumentSequence&gt; usage ............................................... 49
Example 10–2. &lt;FixedDocument&gt; usage ............................................................... 50
Example 10–3. &lt;PageContent&gt; usage ................................................................. 51
Example 10–4. &lt;PageContent&gt;.LinkTargets&gt; usage .............................................. 52
Example 10–5. Fixed page markup ...................................................................... 54
Example 10–6. Canvas composition ..................................................................... 63
Example 11–1. &lt;Path.Data&gt; usage ...................................................................... 72
Example 11–2. &lt;Path.Fill&gt; usage ...................................................................... 74
Example 11–3. &lt;Path.Stroke&gt; usage ................................................................... 75
Example 11–4. &lt;PathGeometry&gt; usage ............................................................... 78
Example 11–5. &lt;ArcSegment&gt; usage ................................................................... 83
Example 11–6. &lt;PolyBezierSegment&gt; usage ......................................................... 85
Example 11–7. &lt;PolyLineSegment&gt; usage ............................................................ 86
Example 11–8. &lt;PolyQuadraticBezierSegment&gt; usage ....................................... 88
Example 11–9. Closed &lt;PathFigure&gt; usage ......................................................... 88
Example 11–10. A path described using abbreviated syntax .............................. 92
Example 11–11. Smooth Bézier curve ............................................................... 93
Example 11–12. Relative commands and curves .............................................. 94
Example 12–1. One-to-one cluster map ........................................................... 105
Example 12–2. Many-to-one cluster map ......................................................... 105
Example 12–3. One-to-many cluster map ......................................................... 106
Example 12–4. Many-to-many cluster map ....................................................... 107
Example 12–5. Using indices to specify advance width ................................... 109
Example 12–6. Using the Indices attribute to specify glyph replacement for a cluster 110
Example 12–7. Text with positive uOffset and vOffset Indices values .......... 114
Example 12–8. Right-to-left text (odd BidiLevel) ............................................ 114
Example 12–9. Sideways text (IsSideways set to true) ..................................... 115
Example 12–10. Vertical text .......................................................................... 115
Example 12–11. Japanese vertical text ............................................................. 116
Example 12–12. Using the CaretStops attribute to determine a valid caret stop position 119
Example 12–13. Basic italic font .................................................................... 121
Example 12–14. Italic font using StyleSimulations attribute ......................... 121
Example 12–15. Kerning ................................................................................. 122
Example 12–16. Ligatures .............................................................................. 122
Example 12–17. Cluster maps ........................................................................ 123
Example 13–1. &lt;SolidColorBrush&gt; usage ......................................................... 127
Example 13–2. &lt;ImageBrush&gt; usage ................................................................ 129
Example 13–3. &lt;VisualBrush.Visual&gt; usage ................................................... 134
Example 13–4. ViewboxUnits and ViewportUnits attribute usage ............... 137
Example 13–5. Tiling brush base image and rendering ................................... 137
Example 13–6. Tiling brush Viewport adjustments ......................................... 138
Example 13–7. Tiling brush viewbox adjustments ............................................ 139
Example 13–8. Image brush with a Viewbox larger than the image ............... 140
Example 13–9. Image brush with TileMode value of None ............................ 141
Example 13–10. Visual brush with TileMode value of None ............................ 143
Example 13–11. Image brush with a TileMode value of Tile ......................... 144
Example 13–12. Visual brush with a TileMode value of Tile ............................ 145
List of Examples

1. Example 13–13. Image brush with a TileMode value of FlipX .................................................. 146
2. Example 13–14. Visual brush with a TileMode value of FlipX .................................................. 147
3. Example 13–15. Image brush with a TileMode value of FlipY ................................................... 148
4. Example 13–16. Visual Brush with a TileMode value of FlipY .................................................. 149
5. Example 13–17. Image brush with a TileMode value of FlipXY .............................................. 150
8. Example 13–20. Linear gradient brush with a SpreadMethod value of Pad ............................ 154
9. Example 13–21. Linear gradient brush with a SpreadMethod value of Reflect ....................... 155
10. Example 13–22. Linear gradient brush with a SpreadMethod value of Repeat ...................... 156
11. Example 13–23. A radial gradient brush .............................................................................. 161
12. Example 13–24. RadialGradientBrush usage ...................................................................... 161
15. Example 13–27. Radial gradient brush with a SpreadMethod value of Repeat ...................... 164
17. Example 13–29. Opacity mask with radial gradient ............................................................. 169
22. Example 14–5. Using a resource reference to fill a brush .................................................... 180
24. Example 14–7. Canvas clip markup and rendering .......................................................... 183
28. Example 14–12. Matrix reversing the y axis ........................................................................ 188
29. Example 14–14. Matrix Rotating .......................................................................................... 188
32. Example 14–18. <Canvas.RenderTransform> usage ......................................................... 191
33. Example 14–19. <Path.RenderTransform> usage ............................................................. 192
35. Example 14–21. <PathGeometry.Transform> usage .......................................................... 194
41. Example 14–27. <Canvas.OpacityMask> usage ................................................................. 204
42. Example 14–28. <Path.OpacityMask> usage .................................................................... 205
44. Example 16–1. Document structure markup ........................................................................ 228
45. Example 16–2. Document outline markup ........................................................................... 230
47. Example 16–4. Story flowing back and forth across a page boundary .............................. 233
48. Example 16–5. Content structure spanning pages ............................................................. 235
49. Example 16–6. StoryFragments part markup ...................................................................... 240
50. Example 16–7. Story fragments markup using a fragment name ..................................... 241
51. Example 16–8. A relative, internal, named-address hyperlink ......................................... 248
52. Example 16–9. A relative internal page address hyperlink ................................................ 248
53. Example 17–1. Optimized interleaving for a single-threaded parsing architecture ................ 254

xvi Working Draft 1.1, February 2008
| Example 17–2. Optimized interleaving for a multi-threaded parsing architecture | 256 |
| Example 17–3. A DiscardControl part | 258 |
| Example 17–4. A SignatureDefinitions part | 264 |
| Example 18–1. Path opacity behavior for overlapping path figures | 285 |
| Example 18–2. Opacity behavior of path stroke intersections | 285 |
| Example 18–3. Opacity behavior of paths with stroked edges | 286 |
1. Scope

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

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

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

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

2.1 Language Notes

In this specification, 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 specification.
- **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.

[Example: One implementation may choose to include the item because a particular marketplace or scenario requires it or because it enhances the product. Another implementation may omit the same item. end example]

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

2.2 Software Conformance

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

In order for a **consumer** to be considered conformant, they **must observe** the following rules apply:

- **They/It MUST NOT** report errors when processing conforming instances of the documented formats except when forced to do so by resource exhaustion.
- **They/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, they **must observe** the following rules apply:

- **They/It MUST NOT** generate any new, non-conforming instances of a documented format.
- **They/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. Editing applications are subject to all of the above rules.

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]
For convenient reference, these rules are collected in §1 for convenient reference.
3. Normative References

The following normative documents contain provisions, which, through reference in this text, constitute provisions of this specification. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this specification 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.

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


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".


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


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


3. Normative References


http://www.rfc-editor.org


http://www.w3.org/TR/2001/REC-xmlbase-20010627/

http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/

4. Definitions

For the purposes of this specification, the following definitions apply. Terms explicitly defined in this specification 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 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, Ccompliant — A digital signature that conforms to the signing rules described in the XPS Document signing policy, regardless of signature validity.

digital signature, Iincompliant — 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 may 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.
4. Definitions

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

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

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

**FixedPage part** — The part that contains all of the visual elements to be rendered on a page.

**named color** — An industry-defined color specification that identifies a particular color in a well-defined color schema, usually for purposes of printing.

**named element** — An element in the document structure markup that refers to an element in the fixed-page markup with a specified name.

**ordering, interleaved** — The layout style of a physical package where parts are broken into pieces and “mixed-in” with pieces from other parts. When delivered, interleaved packages help improve the performance of the consumer processing the package.

**ordering, simple** — Simple ordering the parts in the package are laid out with a defined ordering. When such a package is delivered in a purely linear fashion, starting with the first byte in the package through to the last that, all of the bytes for the first part arrive first, then all of the bytes for the second part, and so on.

**package** — A logical entity that holds a collection of parts.

**package model** — Defines a package abstraction that holds a collection of parts.

**package relationship** — A relationship whose target is a part and whose source is the package as a whole. Package relationships are found in the package relationships part named “/_rels/.rels”.

**part** — A stream of bytes with a MIME content type and associated common properties. Typically corresponds to a file (as on a file system), a stream (as in a compound file), or a resource (as in an HTTP URI).

**part name** — A part name is used to refer to a part in the context of a package, typically as part of a URI. By definition, the part name is, by definition, the path component of a pack URI.

**payload** — A complete collection of interdependent parts and relationships within a package.

**physical imageable size** — Represents the area of a page that is printable by a specific device.

**physical media size** — Represents the physical media on which the content will be printed.

**physical model** — Defines the mapping between the components of the package model to the features of a particular physical format based on the ZIP specification.

**piece** — A portion of a part. Pieces of different parts may be interleaved together. The individual pieces are named using a unique mapping from the part name. The piece name grammar does not conform to the part name grammar. Pieces are not addressable in the package model.

**primary fixed payload root** — The fixed payload root that is referenced by the XPS package StartPart relationship.
PrintTicket part — A PrintTicket part provides the settings used when a package is printed. PrintTicket parts can may be attached to the entire package, or at lower levels in the structure, such as individual pages.

producer — A piece of software or a device that writes XPS packages.

producer bleed size — Represents the overflow (or “bleed”) box used by the producer for registration and layout.

producer content size — Represents the content bounding box specified by the producer.

producer media size — Represents the physical media on which the content will be printed.

property — A characteristic of a markup element, referred to as an attribute of the element.

property attribute — An XPS Document property values can be expressed as either a property attributes or a property elements.

property element — An XPS Document property value can be expressed as either a property attribute or a property element.

property value — The value of a property, expressed as an XML attribute, an XML child element, or an entry in the resource dictionary.

relationships — A relationship represents the kind of connection between a source part and a target part in a package. Relationships make the connections between parts directly discoverable without looking at the content in the parts, and without altering the parts themselves. See also, package relationship.

relationships part — A part containing an XML representation of relationships.

required part — A part, such as an image or font, that is referenced from other parts, and is required for valid processing of the referencing part.

resource definition — A shareable property value, with a name, defined within a resource dictionary. Any property value defined by fixed page markup can be held in a resource dictionary. Each resource definition has a key that is unique within the scope of the resource dictionary.

resource dictionary — A resource dictionary holds resources. Each resource in a resource dictionary carries a name. The resource’s name can be used to reference the resource from a property’s XML attribute.

resource dictionary, remote — A part containing a resource dictionary.

resource reference — An attribute whose value refers to an entry in a resource dictionary. Resource references appear in the format “{StaticResource RscName}” where RscName corresponds to a matching entry in the resource dictionary with an x:Key attribute value.

signature definition — The means by which XPS Document authors provide co-signature requirements and workflow-specific signature information.

signature spot — A visual element that indicates that a digital signature has been applied or requested.

signing rules — The set of rules that define whether a particular digital signature is compliant with the XPS Document signature policy.
4. Definitions

stories — Meaningful semantic blocks that provide structure to markup elements that appear in a FixedPage part.

story — A block of individually readable content in an XPS Document.

story fragment — A portion of a story that appears within the scope of a single fixed page.

stream — A linearly ordered sequence of bytes.

thumbnail — Images that help end-users identify parts of a package or a package as a whole.

XPS Document — A package that contains a discoverable fixed payload and is a format for storing paginated documents defined by the XPS specification.

XPS Document StartPart relationship — The specific relationship type that identifies the root of a fixed payload within an XPS Document.

ZIP Archive — A physical ZIP file that is displayed by the file system. A ZIP archive contains 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 Notes

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.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Required element" /></td>
<td>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.</td>
</tr>
<tr>
<td><img src="image" alt="Optional element" /></td>
<td>Optional element. This box represents an element that can appear zero or one times in markup when the parent element is included.</td>
</tr>
<tr>
<td><img src="image" alt="Range indicator" /></td>
<td>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.</td>
</tr>
<tr>
<td><img src="image" alt="Attribute group" /></td>
<td>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.</td>
</tr>
<tr>
<td><img src="image" alt="Sequence symbol" /></td>
<td>Sequence symbol. The element boxes connected to this symbol can appear in markup in the illustrated sequence only, from top to bottom.</td>
</tr>
<tr>
<td><img src="image" alt="Choice symbol" /></td>
<td>Choice symbol. Only one of the element boxes connected to this symbol can appear in markup.</td>
</tr>
</tbody>
</table>
Type indicator. The elements within the dashed box are of the complex type indicated.
6. Acronyms and Abbreviations

This clause is informative

The following acronyms and abbreviations are used throughout this specification:

- IEC — the International Electrotechnical Commission
- ISO — the International Organization for Standardization
- W3C — World Wide Web Consortium

End of informative text
7. General Description

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

This specification is divided into the following subdivisions:

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

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

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

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

1. Examples within narrative are indicated as follows: [Example: ... end example]
2. Examples of XML are indicated as follows: Example m.n: caption ... end example]
3. [Note: ... end note]
8. XPS Document Format

This specification describes how the XPS Document format is organized internally and rendered externally. It is built upon the principles described in the OPC specification. XPS Documents MUST observe all conformance requirements and recommendations of that specification, except where indicated otherwise [M1.1]. 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 specification that support digital signatures and core properties.

8.1 How This Specification Is Organized

This subclause is informative

<table>
<thead>
<tr>
<th>Clause</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parts and Relationships (§9)</td>
<td>This clause describes how XPS Documents use the packaging model (as described in the OPC specification) to organize data. All part and relationship types are described in detail, including how they are used and what they can contain. This clause also describes the XPS Document markup model, in particular, its parts, and how the XML markup relates to the packaging conventions and recommendations it builds on.</td>
</tr>
<tr>
<td>Documents (§10)</td>
<td>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.</td>
</tr>
<tr>
<td>Graphics (§11)</td>
<td>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.</td>
</tr>
<tr>
<td>Clause</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>Text (§12)</td>
<td>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.</td>
</tr>
<tr>
<td>Brushes (§13)</td>
<td>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.</td>
</tr>
<tr>
<td>Common Properties (§14)</td>
<td>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.</td>
</tr>
<tr>
<td>Color (§15)</td>
<td>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.</td>
</tr>
<tr>
<td>Document Structure and Interactivity (§16)</td>
<td>This clause describes the components of the XPS Document format that support assigning larger semantic meaning to individual page markings. [Example: Such markings might be tables or paragraphs. end example] 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.</td>
</tr>
<tr>
<td>XPS Document Package Features (§17)</td>
<td>This clause describes how package features (as described in the OPC specification) are used and extended in the XPS Document format. This includes interleaving, digital signatures, and core properties.</td>
</tr>
<tr>
<td>Rendering Rules (§18)</td>
<td>This clause provides precise instructions for rendering XPS Document contents to ensure a consistent result among various implementations.</td>
</tr>
<tr>
<td>Elements (§19)</td>
<td>The full list of elements described throughout the preceding clauses is assembled in this clause, in alphabetical order, for easy reference.</td>
</tr>
<tr>
<td>Signature Definitions Schema (§A)</td>
<td>This annex includes the W3C XSD schema for the Signature Definitions part.</td>
</tr>
<tr>
<td>XPS Document Schema (§B)</td>
<td>This annex includes the W3C XSD schema for the FixedDocumentSequence, FixedDocument, and FixedPage parts.</td>
</tr>
<tr>
<td>Resource Dictionary Key Schema (§C)</td>
<td>This annex provides the W3C XSD schema for the resource dictionary Key attribute, used by several elements in the XPS Document schema.</td>
</tr>
<tr>
<td>Document Structure Schema (§D)</td>
<td>This annex provides the W3C XSD schema for the DocumentStructure and StoryFragments parts.</td>
</tr>
<tr>
<td>Discard Control Schema (§E)</td>
<td>This annex includes the W3C XSD schema for the DiscardControl part for interleaving.</td>
</tr>
<tr>
<td>Abbreviated Geometry Syntax Algorithm (§F)</td>
<td>This annex provides a sample algorithm for interpreting the abbreviated geometry syntax provided to succinctly describe geometric regions in a single attribute.</td>
</tr>
</tbody>
</table>
8.2 Package

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

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

Figure 8–1. Package-based XPS Document format
9. Parts and Relationships

The packaging conventions described in the OPC specification can be used to carry any payload. A payload is a complete collection of interdependent parts and relationships within a package. This specification 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 specification is referred to as an XPS Document. Producers and consumers of XPS Documents can implement their own parsers and rendering engines based on this specification.

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. There can be more than one fixed payload in an XPS Document.

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 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 §H. Each part MUST use only the appropriate content type specified in §H [M2.2].

Table 9–1. XPS Document parts

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required/Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>FixedDocumentSequence (§9.1.2)</td>
<td>Specifies a sequence of fixed documents.</td>
<td>REQUIRED [M2.3]</td>
</tr>
<tr>
<td>FixedDocument (§9.1.3)</td>
<td>Specifies a sequence of fixed pages.</td>
<td>REQUIRED [M2.4]</td>
</tr>
<tr>
<td>FixedPage (§9.1.4)</td>
<td>Contains the description of the contents of a page.</td>
<td>REQUIRED [M2.5]</td>
</tr>
<tr>
<td>Font (§9.1.7)</td>
<td>Contains an OpenType or TrueType font.</td>
<td>REQUIRED if a &lt;Glyphs&gt; element is present [M2.6]</td>
</tr>
</tbody>
</table>
### 9. Parts and Relationships

<table>
<thead>
<tr>
<th>Part (§9.x.x)</th>
<th>Description</th>
<th>Required Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image (§9.1.5)</td>
<td>References an image file.</td>
<td>REQUIRED if an <code>&lt;ImageBrush&gt;</code> element is present [M2.7]</td>
</tr>
<tr>
<td>JPEG image (§9.1.5.1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PNG image (§9.1.5.2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TIFF image (§9.1.5.3)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Windows Media Photo image (§9.1.5.4)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Remote resource dictionary (§9.1.8)</td>
<td>Contains a resource dictionary for use by fixed page markup.</td>
<td>REQUIRED if a key it defines is referenced [M2.8]</td>
</tr>
<tr>
<td>Thumbnail (§9.1.6)</td>
<td>Contains a small JPEG or PNG image that represents the contents of the page or package.</td>
<td>OPTIONAL [O2.1]</td>
</tr>
<tr>
<td>PrintTicket (§9.1.9)</td>
<td>Provides settings to be used when printing the package.</td>
<td>OPTIONAL [O2.2]</td>
</tr>
<tr>
<td>ICC profile (§??)</td>
<td>Contains an ICC Version 2 color profile optionally containing an embedded Windows Color System (WCS) color profile.</td>
<td>OPTIONAL [O2.3]</td>
</tr>
<tr>
<td>StoryFragments (§9.1.12)</td>
<td>Contains document content structure for a fixed page.</td>
<td>OPTIONAL [O2.5]</td>
</tr>
<tr>
<td>SignatureDefinitions (§9.1.10)</td>
<td>Contains a list of digital signature spots and signature requirements.</td>
<td>OPTIONAL [O2.6]</td>
</tr>
<tr>
<td>DiscardControl (§??)</td>
<td>Contains a list of resources that are safe for consumers to discard during processing.</td>
<td>OPTIONAL [O2.7]</td>
</tr>
</tbody>
</table>
Example 9–1. A typical XPS Document

end example]
9. Parts and Relationships

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 specification.

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 §H.

Table 9-2. Fixed payload relationships

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Required/Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core Properties</td>
<td>Relationship from the package to the Core Properties part.</td>
<td>OPTIONAL [O2.8]</td>
</tr>
<tr>
<td>Digital Signature Origin</td>
<td>Relationship from the package to the Digital Signature Origin part.</td>
<td>OPTIONAL [O2.9]</td>
</tr>
<tr>
<td>Digital Signature</td>
<td>Relationship from the Digital Signature Origin part to an Digital Signature XML Signature part.</td>
<td>OPTIONAL [O2.10]</td>
</tr>
<tr>
<td>Digital Signature Certificate</td>
<td>Relationship from a Digital Signature XML Signature part to a Digital Signature Certificate part.</td>
<td>OPTIONAL [O2.11]</td>
</tr>
<tr>
<td>Digital Signature Definitions</td>
<td>Relationship from the FixedDocument part to a Digital Signature Definitions part.</td>
<td>OPTIONAL [O2.12]</td>
</tr>
<tr>
<td>DiscardControl</td>
<td>Relationship from the package to a DiscardControl part.</td>
<td>OPTIONAL [O2.13]</td>
</tr>
<tr>
<td>PrintTicket</td>
<td>Relationship from a FixedDocumentSequence part, a FixedDocument part, or a FixedPage part to a PrintTicket part.</td>
<td>OPTIONAL [O2.15]</td>
</tr>
<tr>
<td>Required Resource</td>
<td>Relationship from a FixedPage part to a required resource, including</td>
<td>REQUIRED for each resource referenced</td>
</tr>
</tbody>
</table>

24 Working Draft 1.1, February 2008
Font, Image, ColorProfile, and Remote Resource Dictionary parts. Required resources can be shared between pages.

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.

StartPart
Relationship from the package to the FixedDocumentSequence part that is the primary fixed payload root.

StoryFragments
Relationship from a FixedPage part to the StoryFragments part for the page.

Thumbnail
Relationship from the package to an Image part or from a FixedPage part to an Image part.

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

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

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

Fixed document sequence markup specifies each fixed document in the fixed payload in sequence, using <DocumentReference> elements. The order of <DocumentReference> elements determines document order and MUST be preserved by editing consumers that are also producers [M2.15]. Each <DocumentReference> element SHOULD MUST reference a FixedDocument part by relative URI [M2.8052-2]. For more information, see §10.1.

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

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

The markup in the FixedDocument part specifies the pages of a document in sequence using <PageContent> elements. The order of <PageContent> elements determines page order and
MUST be preserved by editing consumers that are also producers [M2.16]. Each
\(<\text{PageContent}>\) element MUST reference a FixedPage part by relative URI
[M2.81S2.3]. For more information, see §10.2.

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

### 9.1.4 FixedPage Part

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

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

The content type of the FixedPage part is defined in §H

### 9.1.5 Image Parts

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

Images are included in XPS Documents with an \(<\text{ImageBrush}>\) element and an ImageSource
attribute to reference a part with the appropriate content type. For more information, see §H.2.
Fixed pages MUST use a Required Resource relationship to each Image part referenced
[M2.10]. For more information, see §H.3.

XPS Documents support the following image formats:

- JPEG
- PNG
- TIFF
- Windows Media Photo

Color profiles MAY be embedded in image files [O2.18] and SHOULD be used by consumers
[S2.5]. See §15.

### 9.1.5.1 JPEG Images

It is RECOMMENDED that JPEG image part names end with the extension “.jpg” [S2.6]. JPEG
image parts MUST contain images that conform to the JPEG specification [M2.17]. Consumers
SHOULD support JPEG images that contain JFIF-specified APP0 and ICC-specified APP2
markers APP0, APP2, APP13, and APP14 markers [S2.34]. Consumers MUST support JPEG
images that contain the EXIF-specified APP1 marker and interpret the EXIF color space correctly
[M2.78].
Table 9–3. Supported JPEG APPn markers

<table>
<thead>
<tr>
<th>APPn marker</th>
<th>Originating source</th>
</tr>
</thead>
<tbody>
<tr>
<td>APP0</td>
<td>JFIF specification</td>
</tr>
<tr>
<td>APP1</td>
<td>EXIF extension defined by JEITA</td>
</tr>
<tr>
<td>APP2</td>
<td>ICC profile marker defined by the ICC specification</td>
</tr>
<tr>
<td>APP13</td>
<td>Photoshop 3.0 extension</td>
</tr>
<tr>
<td>APP14</td>
<td>Adobe DCT Filters in PostScript Level 2 extension</td>
</tr>
</tbody>
</table>

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

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

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

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

[Note: If both APP2 and APP13 markers are specified, the APP2 marker takes precedence. end note] If the JPEG image is embedded in a TIFF image, the TIFF ICC profile settings are used.

If no color profile is embedded in the JPEG image or stored in a separate part associated with the JPEG image according to the mechanisms described in §15.2.8, then the default color space MUST be treated as defined in §15.2.9 [M8.30].

9.1.5.2 PNG Images

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

Table 9–4. Support for ancillary PNG chunks

<table>
<thead>
<tr>
<th>Chunk</th>
<th>Support Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>tRNS</td>
<td>MUST Support [M2.19]</td>
</tr>
<tr>
<td>iCCP</td>
<td>MUST Support [M2.20]</td>
</tr>
<tr>
<td>sRGB</td>
<td>MUST Ignore [M2.21]</td>
</tr>
<tr>
<td>cHRM</td>
<td>MUST Ignore [M2.22]</td>
</tr>
<tr>
<td>gAMA</td>
<td>MUST Ignore [M2.23]</td>
</tr>
<tr>
<td>sBIT</td>
<td>MUST Ignore [M2.24]</td>
</tr>
</tbody>
</table>

If no color profile is embedded in the PNG image or stored in a separate part associated with the PNG image according to the mechanisms described in §15.2.8, then the default color space MUST be treated as defined in §15.2.9 [M8.30].
### 9.1.5.3 TIFF Images

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

#### Table 9–5. Supported TIFF 6.0 tags

<table>
<thead>
<tr>
<th>Image type</th>
<th>Tags</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bilevel images</td>
<td>PhotometricInterpretation (0 and 1)</td>
</tr>
<tr>
<td></td>
<td>Compression (1, 2, 3, 4, 5, or 32773)</td>
</tr>
<tr>
<td></td>
<td>ImageLength</td>
</tr>
<tr>
<td></td>
<td>ImageWidth</td>
</tr>
<tr>
<td></td>
<td>ResolutionUnit (1, 2, or 3)</td>
</tr>
<tr>
<td></td>
<td>RowsPerStrip</td>
</tr>
<tr>
<td></td>
<td>StripByteCounts</td>
</tr>
<tr>
<td></td>
<td>StripOffsets</td>
</tr>
<tr>
<td></td>
<td>XResolution</td>
</tr>
<tr>
<td></td>
<td>YResolution</td>
</tr>
<tr>
<td>Grayscale images</td>
<td>PhotometricInterpretation (0 and 1)</td>
</tr>
<tr>
<td></td>
<td>BitsPerSample (4, 8, or 16)</td>
</tr>
<tr>
<td></td>
<td>Compression (1, 5, 6, or 32773)</td>
</tr>
<tr>
<td></td>
<td>ImageLength</td>
</tr>
<tr>
<td></td>
<td>ImageWidth</td>
</tr>
<tr>
<td></td>
<td>ResolutionUnit (1, 2, or 3)</td>
</tr>
<tr>
<td></td>
<td>RowsPerStrip</td>
</tr>
<tr>
<td></td>
<td>StripByteCounts</td>
</tr>
<tr>
<td></td>
<td>StripOffsets</td>
</tr>
<tr>
<td></td>
<td>XResolution</td>
</tr>
<tr>
<td></td>
<td>YResolution</td>
</tr>
<tr>
<td>Palette color images</td>
<td>BitsPerSample (1, 4, or 8)</td>
</tr>
<tr>
<td></td>
<td>ColorMap</td>
</tr>
<tr>
<td></td>
<td>Compression (1, 5, or 32773)</td>
</tr>
<tr>
<td></td>
<td>ImageLength</td>
</tr>
<tr>
<td></td>
<td>ImageWidth</td>
</tr>
<tr>
<td></td>
<td>PhotometricInterpretation (3)</td>
</tr>
<tr>
<td></td>
<td>ResolutionUnit (1, 2, or 3)</td>
</tr>
<tr>
<td></td>
<td>RowsPerStrip</td>
</tr>
<tr>
<td></td>
<td>StripByteCounts</td>
</tr>
<tr>
<td></td>
<td>StripOffsets</td>
</tr>
<tr>
<td></td>
<td>XResolution</td>
</tr>
<tr>
<td></td>
<td>YResolution</td>
</tr>
<tr>
<td>RGB images</td>
<td>BitsPerSample (8,8,8 or 16,16,16; or if ExtraSamples=1 or 2: 8,8,8,8</td>
</tr>
</tbody>
</table>
or 16,16,16,16)
Compression (1, 5, 6, or 32773)
ExtraSamples (0, 1, or 2)
ICC Color Profile [tag 34675]
ImageLength
ImageWidth
PhotometricInterpretation (2)
PlanarConfiguration (1)
ResolutionUnit (1, 2, or 3)
RowsPerStrip
SamplesPerPixel (3; or if ExtraSamples=1 or 2: 4)
StripByteCounts
StripOffsets
XResolution
YResolution

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

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

2. If the ResolutionUnit tag is set to 1 (no units), XResolution and YResolution are interpreted in the same manner as if the ResolutionUnit was set to 2 (inches).

3. If no color profile is embedded in the TIFF image or stored in a separate part associated with the TIFF image according to the mechanisms described in §15.2.8, then the default color space MUST be treated as defined in §15.2.9 [M8.30].
The following features of the TIFF specification MUST be supported in addition to the tags described in Table 9–5:

- Baseline TIFF (Sections 1–10) with the exception of the following tags [M2.26]:
  - CellLength
  - CellWidth
  - GrayResponseCurve
  - GrayResponseUnit
  - MaxSampleValue
  - MinSampleValue
  - Orientation
  - Thresholding

- CCITT bilevel encodings (Section 11) [M2.28]

- CMYK images (Section 16) [M2.29]
• Associated alpha data (Section 18) [M2.30]
  o ExtraSamples tag value of 1: Treat alpha as pre-multiplied alpha (see §18.4.1 for details)
  o ExtraSamples tag value of 2: Treat alpha as non-pre-multiplied alpha
• LZW compression (Section 13) [M2.31]
• Differencing predictors (Section 14) [M2.32]
• JPEG compression (Section 22)
  o Only compression mode 6 MUST be supported [M2.33]
• Embedded ICC Profile (described in the ICC specification) [M2.34]
• EXIF IFD (tag 34665) as described in the EXIF specification. The EXIF color space MUST be interpreted correctly [M2.79].

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

[Note: Many TIFF images in circulation today deviate from the TIFF Specification. Over time, TIFF-consuming implementations have developed a certain tolerance for such deviations by attempting to deduce the intent of the TIFF image author and correct for apparent errors or deviations. end note] Therefore, XPS Document consumers SHOULD do the following [S2.11]:

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

end note]

9.1.5.4 Windows Media Photo Images

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

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

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

If no color profile is embedded in the Windows Media Photo image or stored in a separate part associated with the Windows Media Photo image according to the mechanisms described in §15.2.8, then the default color space MUST be treated as defined in §15.2.9 [M8.30].

**9.1.6 Thumbnail Parts**

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

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

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

For more information about the relationship type for thumbnail parts, see §H.3.
9.1.7 Font Parts

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

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

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

\[
\text{fontface} = \*\text{DIGIT}
\]

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

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

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

9.1.7.1 Subsetting Fonts

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

9.1.7.2 OpenType Font Embedding

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

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

For further details on licensing rights of OpenType fonts, see the description of the OS/2 and Windows Metrics.”
Table 9–7. Guidelines for OpenType font embedding

<table>
<thead>
<tr>
<th>Bit/mask</th>
<th>Licensing right intent</th>
<th>Producer rules</th>
<th>Consumer rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 / 0x0000</td>
<td>Installable embedding.</td>
<td>SHOULD do embedded font obfuscation [S2.16] (see §9.1.7.3 for details).</td>
<td>SHOULD NOT extract or install permanently (see below) [S2.17].</td>
</tr>
<tr>
<td>0x0001</td>
<td>Reserved, must be 0.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 / 0x0002</td>
<td>Restricted license embedding. If only this bit is set, the font MUST NOT be modified, embedded or exchanged in any manner without obtaining permission from the legal owner.</td>
<td>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 &lt;Path&gt; element [S2.18].</td>
<td>Render embedded images.</td>
</tr>
<tr>
<td>2 / 0x0004</td>
<td>For preview and print embedding, font can be embedded and temporarily used on remote systems. However, documents containing any preview and print fonts MUST NOT be modified or edited [M2.43].</td>
<td>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 §H.3 for details.</td>
<td>MUST NOT extract or install permanently [M2.45]. MUST NOT modify or edit the XPS Document markup or hierarchical structure starting from the &lt;FixedDocument&gt; element [M2.43].</td>
</tr>
<tr>
<td>3 / 0x0008</td>
<td>Editable embedding.</td>
<td>MUST do embedded font obfuscation [M2.46] (see §9.1.7.3).</td>
<td>MUST NOT extract or install permanently [M2.47].</td>
</tr>
<tr>
<td>4–7</td>
<td>Reserved, must be 0.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8 / 0x0100</td>
<td>No subsetting.</td>
<td>MUST do embedded font obfuscation (see §9.1.7.3) [M2.48]. MUST NOT subset font before embedding.</td>
<td>MUST NOT extract or install permanently [M2.50].</td>
</tr>
<tr>
<td>9 / 0x0200</td>
<td>Bitmap embedding only.</td>
<td>MUST do embedded font obfuscation [M2.51] (see §9.1.7.3). MUST embed only bitmap characters contained in the font [M2.51]. If no bitmap characters are present in the font,</td>
<td>MUST NOT extract or install permanently [M2.52].</td>
</tr>
</tbody>
</table>
9.1.7.3 Embedded Font Obfuscation

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

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

Embedded font obfuscation achieves the following goals:

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

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

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 [S2.19]. [Note: Microsoft implementations for XPS Documents always perform obfuscated font embedding and do not extract or permanently install the font [S2.19]. end note] However, there are vertical solutions in which implementations might benefit from un-obfuscated font embedding. In these cases, implementations could omit obfuscation or extract and install the embedded font.

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

1. Generate a 128-bit GUID (Globally Unique Identifier) for the font to be obfuscated. Instead of a true GUID, a 128-bit random number MAY be used [O2.21]. The 16 bytes of the 128-bit GUID are referred to in the following text by the placeholder names B_{00}, B_{01}, B_{02}, B_{03}; B_{10}, B_{11}; B_{20}, B_{21}; B_{30}, B_{31}, B_{32}, B_{33}, B_{34}, B_{35}, B_{36}, and B_{37}. The order in which bytes are assigned to these placeholders does not matter, as long as it is consistent for obfuscation and de-obfuscation.

2. Generate a part name for the obfuscated font using the GUID. The last segment of the part name MUST be of the form “B_{03}B_{02}B_{01}B_{00}” or “B_{03}B_{02}B_{01}B_{00}” where each B_{x} represents a placeholder for one byte of the GUID, represented as two hex digits [M2.54]. The part name MAY have an arbitrary extension (identified by the placeholder “.ext”) [O2.22]. It is RECOMMENDED that the extension for TrueType fonts be “.odttf” and for TrueType collections be “.odttc” [S2.20].

3. The content type for the part containing the obfuscated font MUST match the definition in H, “Standard Namespaces and Content Types” [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_{29}, B_{28}, B_{27}, B_{26}, B_{25}, B_{24}, B_{23}, B_{22}, B_{21}, B_{20}, B_{19}, B_{18}, B_{17}, B_{16}, B_{15}, B_{14}, B_{13}, B_{12}, B_{11}, B_{10}, B_{09}, B_{08}, B_{07}, B_{06}, B_{05}, B_{04}, B_{03}, B_{02}, B_{01}, B_{00}.

**Note:** The content type for the part containing the obfuscated font MUST NOT embed the font [M2.51].

10–15 Reserved, must be 0.
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 H, process the font without any de-obfuscation steps.

2. For font parts with the obfuscated font content type as specified in H, 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<sub>37</sub>, B<sub>36</sub>, B<sub>35</sub>, B<sub>34</sub>, B<sub>33</sub>, B<sub>32</sub>, B<sub>31</sub>, B<sub>30</sub>, B<sub>20</sub>, B<sub>21</sub>, B<sub>10</sub>, B<sub>11</sub>, B<sub>00</sub>, B<sub>01</sub>, B<sub>02</sub>, and B<sub>03</sub>, 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].

Editing 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, editing 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].

Printing and display only 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]:

<table>
<thead>
<tr>
<th>Platform ID</th>
<th>Encoding ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>10</td>
<td>Unicode with surrogates</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>Unicode without surrogates</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>Wansung</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>Big5</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>Prc</td>
</tr>
</tbody>
</table>
All further processing for that font MUST use the selected cmap table \([M2.55]\).

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

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

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

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

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

- Search is unpredictable
- Copy and paste functionality is unpredictable

Producers \textit{are discouraged} \textit{should not} from using this non-standard encoding and they \textit{should} write PUA codepoints to the UnicodeString attribute \([S2.15]\).

### 9.1.8 Remote Resource Dictionary Parts

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

### 9.1.9 PrintTicket Parts

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 Mapping PrintTicket Parts to Fixed Payload Parts

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].

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

- Consumers MUST process job-level, document-level and page-level settings of PrintTicket parts associated with FixedDocumentSequence parts [M2.60].
- Producers SHOULD only attach PrintTicket parts containing only document-level and page-level settings with FixedDocument parts [S2.22].
- 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 [M2.61].
- Producers SHOULD only attach PrintTicket parts containing only page-level settings with FixedPage parts [S2.23].
- 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 [M2.62].

PrintTicket parts are associated with parts via the PrintTicket relationship defined in H, “Standard Namespaces and Content Types.”

9.1.9.3 Processing PrintTicket Parts

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

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

Second, consumers MUST validate the PrintTicket according to the methods defined in the PrintTicket Validation Checklist of the Print Schema documentation [M2.64]. Following validation, the printing consumer MUST properly interpret the print settings according to the following rules for merging two PrintTicket parts [M2.65].
Print settings are expressed by scoped Print Schema elements. Elements can interact between different levels in the PrintTicket hierarchy. Elements that interact between levels MUST be specified at the root of each level ticket [M2.59]. A keyword merge conflict between PrintTicket settings is defined as the same root-level Print Schema element denoted by the same name attribute value appearing in multiple level tickets. There are two options for interactions:

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

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

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

1. Validate the job-level PrintTicket associated with the fixed document sequence by merging and validating the PrintTicket with the default PrintTicket for the print consumer. The default PrintTicket represents the default configuration of the print consumer’s state; in the case of a print consumer, the state is the current device and print driver configuration. Call the resulting ticket the “validated job-level PrintTicket.” If no job-level PrintTicket is supplied, use the default PrintTicket.

2. For each FixedDocument\(X\) referenced by the fixed document sequence, perform the following steps:
   a. Merge and validate the document-level PrintTicket associated with FixedDocument\(X\) with the validated job-level PrintTicket from Step 1.
   b. Call the resulting ticket the “validated Document\(X\)-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 FixedPage\(Y\) referenced by each FixedDocument\(X\) in the fixed document sequence, perform the following steps:
   a. Merge and validate the page-level PrintTicket associated with FixedPage\(Y\) with the validated Document\(X\)-level PrintTicket from Step 2.
   b. Call the resulting ticket the “validated Page\(XY\)-level PrintTicket.” If no page-level PrintTicket is supplied for the current fixed page, the validated Document\(X\)-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 §H.

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


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

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

9.1.12 StoryFragments Part

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


The StoryFragments part is referenced by relationship from its associated FixedPage part. The content type of the StoryFragments part is specified in §H.2.
9.2 Part Naming Recommendations

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

By choosing part names according to a well-defined, human-readable convention, the resulting package is easier to browse and specific parts are more easily located. Part names MUST still conform to the syntax specified in the OPC specification [M1.1]. It is RECOMMENDED that producers of XPS Documents use the following part naming convention:

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

```
/FixedDocSeq.fdseq
/Documents/1/FixedDocument.fdoc
/Documents/1/Pages/1.fpage
/Documents/1/Pages/2.fpage
/Documents/1/Resources/Fonts/FontA.ttf
/Documents/1/Resources/Images/ImageB.jpg
/Documents/1/Metadata/Document_PT.xml
```

Example 9–2. XPS Document part naming
9.3 XPS Document Markup

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

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

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

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

1. XPS Document markup should be parsimonious; that is, it should include only the minimum set of primitive operations and markup constructs necessary to render text and graphics with full fidelity. Redundancy in the specification increases the opportunity for independent implementations, such as printer-resident raster image processors (RIPs), viewers, and interactive applications, to introduce accidental incompatibilities. Redundancy also increases the cost of implementation and testing, and, typically, the required memory footprint.

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

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

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

- DocumentStructure
- FixedDocument
- FixedDocumentSequence
- FixedPage
- Relationships
- Remote Resource Dictionary
- SignatureDefinitions
- StoryFragments

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

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

9.3.2 XML Usage

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

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

2. The XML 1.0 specification allows for the usage of Data Type Definitions (DTDs), which enable Denial of Service attacks, typically through the use of an internal entity expansion technique. As mitigation for this potential threat, DTD content MUST NOT be used in the XML markup defined in this specification, and consumers MUST treat the presence of DTD content as an error [M2.71].

3. If the XML content contains the Markup Compatibility and Extensibility namespace, as described in the Markup Compatibility and Extensibility specification, it MUST be processed to remove Markup Compatibility and Extensibility elements and attributes, ignorable namespace declarations, and ignored elements and attributes before applying further validation rules below [M2.69].

4. XML content MUST be valid against the corresponding W3C XSD schema defined in this specification. In particular, the XML content MUST NOT contain elements or attributes drawn from namespaces that are not explicitly defined in the corresponding XSD unless
the XSD allows elements or attributes drawn from any namespace to be present in particular locations in the XML markup [M2.72].

5. 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 specification [M2.73].

9.3.3 Markup Model

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

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

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

9.3.3.1 Namespaces

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

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

9.3.3.2 Properties

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

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

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

9.3.3.2.1 Composable Property Values

Some fixed page properties are composable, meaning that the page marking effect is determined by combining the property value of a given element with that of its parent and ancestor elements. [Example: A <Path> element with an Opacity value of 0.5 nested inside a <Canvas> element with an Opacity value of 0.5 results in an effective 25% opacity of the <Path> element when rendered. end example]
The coordinate space used to render page marking elements is also composable. By default, elements are rendered in a coordinate space with units of 1/96". The effective coordinate space for a particular element is created by sequentially applying each parent and ancestor element’s affine matrix transformation, specified with the Transform or RenderTransform properties, from outermost to innermost, including the element’s own affine matrix transformation.

For more information, see §18.1.3, and §18.518.4.1.

### 9.3.3.2.2 Property Attribute Syntax

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

#### Example 9-3. Property attribute syntax

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

```xml
<.SolidColorBrush Color="#FF0000" />
```

#### 9.3.3.2.3 Property Element Syntax

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

The order of child property elements is significant: they MUST occur before any contents of the parent element and they MUST appear in the sequence specified in the schema [M2.72].
Example 9-4. Property element syntax

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

```xml
<Canvas>
  <!-- First, the property-related child elements -->
  <Canvas.RenderTransform>
    <MatrixTransform Matrix="1,0,0,1,0,0" />
  </Canvas.RenderTransform>
  <Canvas.Clip>
    <PathGeometry>
      ...
    </PathGeometry>
  </Canvas.Clip>
  <!-- Then, the "contents" -->
  <Path ...
    ...
  </Path>
  <Glyphs ... />
</Canvas>
```

end example]

9.3.4 White space

XPS Documents allow flexible whitespace usage in markup. Wherever a single whitespace character is allowed, multiple whitespace characters MAY be used [O2.30]. Attributes that specify comma-delimited attribute values MAY, unless specified otherwise, OPTIONALLY include whitespace characters preceding or following the comma [O2.31]. XPS Document markup MUST NOT use the xml:space attribute [M2.75]. Additionally, 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 Specification [O2.32].

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

9.3.5 Language

Language information supports the following features:

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

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

9. Parts and Relationships

9.3.5.1 xml:lang Attribute

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 [M2.76]. This attribute is defined in the W3C XML specification.

xml:lang is REQUIRED for <FixedPage> elements and MAY be used with <Canvas>, <Path>, and <Glyphs> elements; it is not valid on any other fixed page markup element [M2.72]. xml:lang is also REQUIRED for the <DocumentOutline> element for document structure and OPTIONAL for the <OutlineEntry> element [M2.72]. When the language of the contents is unknown and is 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

  <DocumentReference Source="Documents/1/FixedDocument.fdoc" />
  <DocumentReference Source="Documents/2/FixedDocument.fdoc" />
</FixedDocumentSequence>

10.1.1 <DocumentReference> Element

element DocumentReference

diagram

attributes

<table>
<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>
10. Documents

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

### 10.2 `<FixedDocument>` Element

**element FixedDocument**

  <PageContent Source="Pages/1.fpage" />
  <PageContent Source="Pages/2.fpage" />
</FixedDocument>

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

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

**10.2.1 `<PageContent>` Element**

**element PageContent**

<table>
<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></td>
</tr>
</tbody>
</table>

Specifies a URI that refers to the page content, held in a distinct part within the package. The content identified MUST be a FixedPage part within the XPS Document [M3.5].
Width ST_GEOne

Typical width of pages contained in the page content.

Height ST_GEOne

Typical height of pages contained in the page content.

**annotation**

Defines a reference from a fixed document to a part that contains a `<FixedPage>` element.

Each `<PageContent>` element refers to the source of the content for a single page. The number of pages in the document can be determined by counting the number of `<PageContent>` elements.

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

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

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 [M3.6].

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

```xml
  <PageContent Source="Pages/1.fpage" Height="1056" Width="816" />
  <PageContent Source="Pages/2.fpage" Height="1056" Width="816" />
</FixedDocument>
```

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

**element** `<PageContent.LinkTargets>`

**diaagram**

```
PageContent.LinkTargets -- [LinkTarget (1..*)]
```

**annotation**

Contains a collection of `<LinkTarget>` elements, each of which is addressable via hyperlink.

The `<PageContent.LinkTargets>` element defines the list of link targets that specify each named element on the page that can be addressed by hyperlink.
Example 10–4. `<PageContent.LinkTargets>` usage

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

```xml
  <PageContent Source="Pages/1.fpage" Height="1056" Width="816" />
  <PageContent Source="Pages/2.fpage" Height="1056" Width="816">
    <PageContent.LinkTargets>
      <LinkTarget Name="Anchor1" />
      <LinkTarget Name="Anchor2" />
    </PageContent.LinkTargets>
  </PageContent>
</FixedDocument>
```

end example]

10.2.3 `<LinkTarget>` Element

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

The <FixedPage> element...

**Diagram**

<table>
<thead>
<tr>
<th>attributes</th>
<th>Name</th>
<th>Type</th>
<th>Use</th>
<th>Default</th>
<th>Fixed</th>
<th>Annotation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Width</td>
<td>ST_GEOOne</td>
<td>required</td>
<td></td>
<td></td>
<td>Width of the page, expressed as a real number in units of the effective coordinate space.</td>
</tr>
<tr>
<td></td>
<td>Height</td>
<td>ST_GEOOne</td>
<td>required</td>
<td></td>
<td></td>
<td>Height of the page, expressed as a real number in units of the effective coordinate space.</td>
</tr>
<tr>
<td></td>
<td>ContentBox</td>
<td>ST_ContentBox</td>
<td></td>
<td></td>
<td></td>
<td>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).</td>
</tr>
<tr>
<td></td>
<td>BleedBox</td>
<td>ST_BleedBox</td>
<td></td>
<td></td>
<td></td>
<td>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).</td>
</tr>
<tr>
<td></td>
<td>xml:lang</td>
<td></td>
<td>required</td>
<td></td>
<td></td>
<td>Specifies the default language used for the current element and for any child or descendant elements.</td>
</tr>
</tbody>
</table>
The \textlt{<FixedPage>} element contains the contents of a page and is the root element of a
FixedPage part. The fixed page contains the elements that together form the basis for all
markings rendered on the page: \textlt{<Paths>}, \textlt{<Glyphs>}, and the optional \textlt{<Canvas>} grouping
element.

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

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

Example 10–5. Fixed page markup

\begin{verbatim}
<FixedPage Height="1056" Width="816" xml:lang="en-US"
     xmlns="http://schemas.microsoft.com/xps/2005/06">
  <Glyphs
    OriginX="96"
    OriginY="96"
    UnicodeString="This is Page 1!"
    FontUri="../Resources/Fonts/Times.TTF"
    FontRenderingEmSize="16" />
</FixedPage>
\end{verbatim}

end example]

10.3.1 BleedBox Attribute

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

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

\begin{itemize}
  \item The \textlt{BleedBox BleedOriginX} value MUST be less than or equal to 0 \([M3.7]\).
  \item The \textlt{BleedBox BleedOriginY} value MUST be less than or equal to 0 \([M3.8]\).
  \item The \textlt{BleedBox BleedWidth} value MUST be greater than or equal to the fixed page Width
        attribute value plus the absolute value of the \textlt{BleedBox BleedOriginX} value \([M3.9]\).
  \item The \textlt{BleedBox BleedHeight} value MUST be greater than or equal to the fixed page Height
        attribute value plus the absolute value of the \textlt{BleedBox BleedOriginY} value \([M3.10]\).
\end{itemize}

If the \textlt{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.
10.3.4 Media Orientation and Scaling

When rendering a fixed page for printing, 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.]
Figure 10–2. Matching PrintTicket and fixed page size and orientation

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

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

Finally, if the PrintTicket specifies the PageMediaSize value as 4x6 and the PageOrientation value as Portrait, the bottom and right portions of the fixed page content are clipped, as shown below.
Figure 10-4. Matching PrintTicket and fixed page orientation with differing size

end example]

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

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

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

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

- FitApplicationBleedSizeToPageImageableSize
- FitApplicationContentSizeToPageImageableSize
- FitApplicationMediaSizeToPageImageableSize
- FitApplicationMediaSizeToPageMediaSize

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

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

10.3.4.2 FitApplicationContentSizeToPageImageableSize

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

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

10.3.4.4 FitApplicationMediaSizeToPageMediaSize

Consumers MUST scale the height and width (producer media size) to the PageMediaSize, preserving the aspect ratio [M3.19]. See the Print Schema PageScaling definition.
10.4 <Canvas> Element

element Canvas

<table>
<thead>
<tr>
<th>attributes</th>
<th>Name</th>
<th>Type</th>
<th>Use</th>
<th>Default</th>
<th>Fixed</th>
<th>Annotation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RenderTransform</td>
<td>ST_RscRefMatrix</td>
<td></td>
<td></td>
<td></td>
<td>Establishes a new coordinate frame for the child and descendant elements of the canvas, such as another canvas. Also affects clip and opacity mask.</td>
</tr>
<tr>
<td></td>
<td>Clip</td>
<td>ST_RscRefAbbrGeomF</td>
<td></td>
<td></td>
<td></td>
<td>Limits the rendered region of the element.</td>
</tr>
<tr>
<td></td>
<td>Opacity</td>
<td>ST_ZeroOne</td>
<td>1.0</td>
<td></td>
<td></td>
<td>Defines the uniform</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OpaqueMask</td>
<td>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.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Contains a string value that identifies the current element as a named, addressable point in the document for the purpose of hyperlinking.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RenderOptions.EdgeMode</td>
<td>Controls how edges of paths within the canvas are rendered. The only valid value is Alased. Omitting this attribute causes the edges to be rendered in the consumer’s default manner.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FixedPage.NavigateUri</td>
<td>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.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xml:lang</td>
<td>Specifies the default language used for the current element and for any child or descendant elements. The language is specified according to RFC 3066.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x:Key</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 [M3.20].</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The `<Canvas>` element groups elements together. [Example: `<Glyphs>` and `<Path>` elements can be grouped in a canvas in order to be identified as a unit (as a hyperlink destination) or to apply a composed property value to each child and ancestor element. end example]

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

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

Example 10–6. Canvas composition

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

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

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

```xml
<Path>
  <Path.Fill>
    <SolidColorBrush Color="#808080" />
  </Path.Fill>
  <Path.Data>
    <PathGeometry>
      <PathFigure StartPoint="0,0" IsClosed="true">
        <PolyLineSegment Points="200,0 200,100 0,100 0,0" />
      </PathFigure>
    </PathGeometry>
  </Path.Data>
</Path>
```
10. Documents

This markup is rendered as follows:

```
<Canvas Opacity="0.5" RenderTransform="0.75,0,0,0.75,25,46">
  <Path>
    <Path.Fill>
      <SolidColorBrush Color="#0000FF" />
    </Path.Fill>
    <Path.Data>
      <PathGeometry>
        <PathFigure StartPoint="0,0" IsClosed="true">
          <PolyLineSegment Points="200,0 200,100 0,100 0,0" />
        </PathFigure>
      </PathGeometry>
    </Path.Data>
  </Path>
</Canvas>
```

This markup is rendered as follows:

end example]

10.5 <Path> Element

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

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

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

element Path
## 11. Graphics

<table>
<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>Data</td>
<td>ST_RscRefAbbrGeomF</td>
<td></td>
<td></td>
<td></td>
<td>Describes the geometry of the path.</td>
</tr>
<tr>
<td>Fill</td>
<td>ST_RscRefColor</td>
<td></td>
<td></td>
<td></td>
<td>Describes the brush used to paint the geometry specified by the Data property of the path.</td>
</tr>
<tr>
<td>RenderTransform</td>
<td>ST_RscRefMatrix</td>
<td></td>
<td></td>
<td></td>
<td>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 &lt;Path.Data&gt; property element.</td>
</tr>
<tr>
<td>Clip</td>
<td>ST_RscRefAbbrGeomF</td>
<td></td>
<td></td>
<td></td>
<td>Limits the rendered region of the element.</td>
</tr>
<tr>
<td>Opacity</td>
<td>ST_ZeroOne</td>
<td></td>
<td>1.0</td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>OpacityMask</td>
<td>ST_RscRef</td>
<td></td>
<td></td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>Stroke</td>
<td>ST_RscRefColor</td>
<td></td>
<td></td>
<td></td>
<td>Specifies the brush used to draw the stroke.</td>
</tr>
<tr>
<td>StrokeDashArray</td>
<td>ST_EvenArrayPos</td>
<td></td>
<td></td>
<td></td>
<td>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</td>
</tr>
<tr>
<td>StrokeDashCap</td>
<td>ST_DashCap</td>
<td>Flat</td>
<td>Specifies how the ends of each dash are drawn. Valid values are Flat, Round, Square, and Triangle.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------------</td>
<td>------------</td>
<td>-----------</td>
<td>--------------------------------------------------------------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>StrokeDashOffset</td>
<td>ST_Double</td>
<td>0.0</td>
<td>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.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>StrokeEndLineCap</td>
<td>ST_LineCap</td>
<td>Flat</td>
<td>Defines the shape of the end of the last dash in a stroke. Valid values are Flat, Square, Round, and Triangle.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>StrokeStartLineCap</td>
<td>ST_LineCap</td>
<td>Flat</td>
<td>Defines the shape of the beginning of the first dash in a stroke. Valid values are Flat, Square, Round, and Triangle.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>StrokeLineJoin</td>
<td>ST_LineJoin</td>
<td>Miter</td>
<td>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.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>StrokeMiterLimit</td>
<td>ST_GEOne</td>
<td>10.0</td>
<td>The ratio between the maximum miter length and half of the stroke thickness. This value is significant only if the StrokeLineJoin attribute specifies Miter.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| 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.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Contains a string value that identifies the current element as a named, addressable point in the document for the purpose of hyperlinking.</td>
</tr>
<tr>
<td>FixedPage.NavigateUri</td>
<td>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.</td>
</tr>
<tr>
<td>xml:lang</td>
<td>Specifies the default language used for the current element and for any child or descendant elements. The language is specified according to RFC 3066.</td>
</tr>
<tr>
<td>x:Key</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 [M4.1].</td>
</tr>
<tr>
<td>AutomationProperties.Name</td>
<td>A brief description of the &lt;Path&gt; for accessibility purposes, particularly if filled with an &lt;ImageBrush&gt;.</td>
</tr>
<tr>
<td>AutomationProperties.HelpText</td>
<td>A detailed description of the &lt;Path&gt; for accessibility purposes, particularly if filled with an &lt;ImageBrush&gt;.</td>
</tr>
<tr>
<td>SnapsToDevicePixels</td>
<td>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].</td>
</tr>
<tr>
<td>annotation</td>
<td>Defines a single graphical effect to be rendered to the page. It paints a geometry with a brush and draws a stroke around it.</td>
</tr>
</tbody>
</table>
The `<Path>` element is the sole means of adding vector graphics and images to a fixed page. It defines a single vector graphic to be rendered on a page. Some properties of the `<Path>` element are composable, meaning that the markings rendered to the page are determined by a combination of the property and all of the like-named properties of its parent and ancestor elements.

The Data property contains a geometric description of the area on which to apply a given effect. This description can take one of two forms: verbose or abbreviated. In the verbose form, the geometry is described in the `<Path.Data>` property element using the elements described in §11.2, “Geometries and Figures,” on page 75. In abbreviated form, it is described using abbreviated syntax in the Data attribute. For more information, see §11.2.3.

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

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

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

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

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 [O4.1].

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

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

### 11.1.1 `<Path.Data>` Element

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

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

```xml
<Path Stroke="#000000" StrokeThickness="1">
  <Path.Data>
    <PathGeometry/>
  </Path.Data>
</Path>
```
This markup produces the following results:

```
<PathFigure StartPoint="50,50" IsClosed="true">
  <PolyLineSegment Points="250,50 150,250" />
</PathFigure>
</PathGeometry>
</Path.Data>
</Path>

end example]

11.1.2 <Path.Fill> Element

element Path.Fill

The <Path.Fill> property element specifies the brush that is used to fill the region described by the Data property. This can be a solid color, an image, a gradient, or a vector drawing pattern.
Example 11-2. <Path.Fill> usage

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

```xml
<Path>
  <Path.Fill>
    <SolidColorBrush Color="#0000FF" />
  </Path.Fill>
  <Path.Data>
    <PathGeometry>
      <PathFigure StartPoint="10,10" IsClosed="true">
        <PolyLineSegment Points="50,200 100,40 150,200 200,10 100,105" />
      </PathFigure>
    </PathGeometry>
  </Path.Data>
</Path>
```

This markup produces the following result:

![Image of a filled blue W shape]

end example]

11.1.3 <Path.Stroke> Element

element Path.Stroke

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

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

For more information, see §18.6.

Example 11–3. <Path.Stroke> usage

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

```xml
<Path StrokeThickness="10" Data="M 20,20 L 170,20 L 170,170 L 20,170 Z">
  <LinearGradientBrush MappingMode="Absolute"
    StartPoint="0,0" EndPoint="0,5" SpreadMethod="Reflect">
    <LinearGradientBrush.GradientStops>
      <GradientStop Color="#9999FF" Offset="0.0" />  
      <GradientStop Color="#333366" Offset="1.0" />
    </LinearGradientBrush.GradientStops>
  </LinearGradientBrush>
</Path.Stroke>
</Path>
```

This markup produces the following results:

![Image of a box with a gradient border]

end example]

11.2 Geometries and Figures

Geometries are used to build visual representations of geometric shapes.

The smallest atomic unit in a geometry is a segment. Segments can be lines or curves. One or more segments are combined into a path figure definition. A path figure is a single shape comprised of continuous segments. One or more path figures collectively define an entire path geometry. A path geometry MAY define the fill algorithm to be used on the component path figures [M2.72].
A single path geometry can be used in the Data property of the `<Path>` element to describe its overall geometry. A path geometry can also be used in the Clip property of the `<Canvas>`, `<Path>`, or `<Glyphs>` elements to describe a clipping region.
11.2.1 Geometries

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

11.2.1.1 `<PathGeometry>` Element

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 [M4.3].

The union of the path figures defines the interior of the path geometry according to the FillRule attribute as described in §11.2.1.2.
Example 11-4. <PathGeometry> usage

```
<Path Stroke="#000000">
  <Path.Data>
    <PathGeometry>
      <PathFigure StartPoint="25,75">
        <PolyLineSegment Points="150,75 50,75" />
      </PathFigure>
      <PathFigure StartPoint="50,75" IsClosed="true">
        <ArcSegment
          Size="60,60"
          RotationAngle="0"
          IsLargeArc="true"
          SweepDirection="Counterclockwise"
          Point="125,75" />
      </PathFigure>
      <PathFigure StartPoint="50,75" IsClosed="true">
        <PolyLineSegment Points="25,25 150,25 125,75" />
      </PathFigure>
    </PathGeometry>
  </Path.Data>
</Path>
```

This markup produces the following results:

```
+----+
|    |
|    |
|    |
|    |
|    |
|    |
+----+
```

end example]

11.2.1.2 FillRule Attribute

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

11.2.1.2.1 EvenOdd Fill Algorithm

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

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

11.2.1.3 Figures Attribute

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

11.2.2.1 <PathFigure> Element

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

### Attributes

<table>
<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><code>IsClosed</code></td>
<td><code>ST_Boolean</code></td>
<td>required</td>
<td>false</td>
<td></td>
<td>Specifies whether the path is closed. If set to true, the stroke is drawn &quot;closed,&quot; that is, the last point in the last segment of the path figure is connected with the point specified in the <code>StartPoint</code> attribute, otherwise the stroke is drawn &quot;open,&quot; and the last point is not connected to the start point. Only applicable if the path figure is used in a <code>&lt;Path&gt;</code> element that specifies a stroke.</td>
</tr>
<tr>
<td><code>StartPoint</code></td>
<td><code>ST_Point</code></td>
<td>required</td>
<td></td>
<td></td>
<td>Specifies the starting point for the first segment of the path figure.</td>
</tr>
<tr>
<td><code>IsFilled</code></td>
<td><code>ST_Boolean</code></td>
<td></td>
<td>true</td>
<td></td>
<td>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.</td>
</tr>
</tbody>
</table>

#### Annotation

- Specifies a set of one or more segment elements defining a closed region.
Segment elements are:

- `<ArcSegment>`
- `<PolyBezierSegment>`
- `<PolyLineSegment>`
- `<PolyQuadraticBezierSegment>`

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

### 11.2.2.2 `<ArcSegment>` Element

#### Element ArcSegment

<table>
<thead>
<tr>
<th>attributes</th>
<th>Name</th>
<th>Type</th>
<th>Use</th>
<th>Default</th>
<th>Fixed</th>
<th>Annotation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<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></td>
<td>Size</td>
<td>ST_PointGEO</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></td>
<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></td>
<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></td>
<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></td>
<td>IsStroked</td>
<td>ST_Boolean</td>
<td>true</td>
<td></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>
The `<ArcSegment>` element describes an elliptical arc. It is geometrically defined by the intersection of two ellipses that have the same x radius and y radius. The ellipses intersect at the starting and ending points of the arc.

### Table 11–1. Arc segment definition

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Starting Point</td>
<td>Implicitly defined by the previous point in the path figure definition.</td>
</tr>
<tr>
<td>Ending Point</td>
<td>Specified by the Point attribute.</td>
</tr>
<tr>
<td>Arc Size</td>
<td>Defined by the Size attribute. This value consists of the comma-delimited x and y radii of the ellipses that will be used to define the arc. [Example: &quot;100,50&quot; end example]</td>
</tr>
<tr>
<td>Rotation Angle</td>
<td>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.</td>
</tr>
<tr>
<td>Large Arc Flag</td>
<td>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 &gt;= 180°), and when it is false it uses the smaller arcs (arc length &lt; 180°).</td>
</tr>
<tr>
<td>Sweep Direction</td>
<td>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.</td>
</tr>
</tbody>
</table>
1.1. Working Draft 1.1, February 2008

1. Figure 11-3. Arc choice A
   IsLargeArc = false; SweepDirection = Counterclockwise

2. Figure 11-4. Arc choice B
   IsLargeArc = false; SweepDirection = Clockwise

3. Figure 11-5. Arc choice C
   IsLargeArc = true; SweepDirection = Counterclockwise

4. Figure 11-6. Arc choice D
   IsLargeArc = true; SweepDirection = Clockwise

5. Example 11-5. <ArcSegment> usage
   <Path Stroke="#000000" StrokeThickness="1">
     <Path.Data>
       ...
     </Path.Data>
   </Path>
11.2.2.2 Out-of-Range Attributes

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

- If the arc is impossible to render given the combination of radii specified in the Size
  attribute and the angle of rotation specified in the RotationAngle attribute, the ellipses are
  scaled equally until there is exactly one solution that satisfies the arc requirements to
  pass through the specified Point attribute.

- If the Point attribute is the same as the previous point in the path figure, the segment is
  omitted.

- If either the x or y radius in the Size attribute is 0, the segment is rendered as a poly line
  segment with a single line segment to the x,y coordinates specified by the Point attribute.

- The x or y radius in the Size attribute MUST NOT be negative [M2.72].

- If the RotationAngle value is greater than 360, it is replaced by the value of the
  RotationAngle modulo 360. If it is less than 0, it is replaced with a value normalized to the
  range 0–360.
11.2.2.3 `<PolyBezierSegment>` Element

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

Example 11.6. `<PolyBezierSegment>` usage

```xml
<Path Stroke="#000000" StrokeThickness="1">
  <Path.Data>
    <PathGeometry>
      <PathFigure StartPoint="20,80">
        <PolyBezierSegment Points="70,0 120,160 170,80 120,0 70,160 20,80"/>
      </PathFigure>
    </PathGeometry>
  </Path.Data>
</Path>
```
This markup generates the following results:

![Infinity symbol](image)

end example]

11.2.2.4 `<PolyLineSegment>` Element

element `PolyLineSegment`

diagram

<table>
<thead>
<tr>
<th>attributes</th>
<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></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>
<tr>
<td>IsStroked</td>
<td>ST_Boolean</td>
<td>true</td>
<td></td>
<td></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>

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

Example 11-7. `<PolyLineSegment>` usage

```xml
<Path Stroke="#000000" StrokeThickness="1">
  <Path.Data>
    <PathGeometry>
      <PathFigure StartPoint="10,10">
        <PolyLineSegment Points="140,10 140,55 95,55 65,85 95,115 140,115 140,160 10,160" />
      </PathFigure>
    </PathGeometry>
  </Path.Data>
</Path>
```
This markup produces the following figure:

```
<Path></Path>
```

end example[

### 11.2.2.5 `<PolyQuadraticBezierSegment>` Element

element *PolyQuadraticBezierSegment*

<table>
<thead>
<tr>
<th>attributes</th>
<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></td>
<td>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.</td>
</tr>
<tr>
<td>IsStroked</td>
<td>ST_Boolean</td>
<td>true</td>
<td></td>
<td></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>

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

```xml
<Path Stroke="#000000" StrokeThickness="1">
  <Path.Data>
    <PathGeometry>
      <PathFigure StartPoint="10,10">
        <PolyQuadraticBezierSegment Points="50,200 100,40 150,200 200,10" />
      </PathFigure>
    </PathGeometry>
  </Path.Data>
</Path>
```

This markup produces the following curve:

![Diagram of a curve produced by the example XML markup]

end example]

11.2.2.6 Closed `<PathFigure>`

If the IsClosed attribute of the `<PathFigure>` element is set to true, a straight line is drawn from the last point in the last segment of the `<PathFigure>` element to the StartPoint attribute of the `<PathFigure>` element. If the IsClosed attribute is omitted, its default setting is “false”.

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

Example 11-9. Closed `<PathFigure>` usage

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

```xml
<Path Stroke="#000000" Stroke Thickness="1">
  <Path.Data>
    <PathGeometry>
      <PathFigure StartPoint="10,10" IsClosed="true">
        <ArcSegment
          Size="100,50"
          RotationAngle="45"
          IsLargeArc="true"
          SweepDirection="Counterclockwise"
          Point="200,100" />
      </PathFigure>
    </PathGeometry>
  </Path.Data>
</Path>
```
This markup generates the following figure:

```
end example]
```

### 11.2.3 Abbreviated Geometry Syntax

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

- Line
- Horizontal Line
- Vertical Line
- Cubic Bézier Curve
- Quadratic Bézier Curve
- Smooth Cubic Bézier Curve
- Elliptical Arc

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

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

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

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

Values specifying coordinates can be real numbers.

For more information, see §F.
<table>
<thead>
<tr>
<th>Name</th>
<th>Syntax</th>
<th>Description</th>
<th>Non-Abbreviated Equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>FillRule</td>
<td>F fillRule</td>
<td>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.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>This command MUST appear only as the first command in the abbreviated geometry syntax [M2.72]. This command MUST NOT be specified in the value of the Figures attribute of the &lt;PathGeometry&gt; element [M2.72]. [Example: F 0 end example]</td>
<td></td>
</tr>
<tr>
<td>Move</td>
<td>M x,y</td>
<td>Establishes a new current endpoint. Every geometry MAY specify one or more figures, and MAY be preceded by a FillRule command where allowed [M2.72]. The first figure in a geometry MUST begin with a Move command [M2.72]. Subsequent Move commands indicate the start of a new figure but MAY be omitted, indicating the current endpoint for the subsequent figure is the same as the end point of the previous figure [M2.72]. [Example: M 1.0,1.5 end example]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>or m x,y</td>
<td></td>
<td>&lt;PathFigure&gt; StartPoint attribute</td>
</tr>
<tr>
<td>Line</td>
<td>L x,y</td>
<td>Draws a straight line from the current point to the specified point. [Example: L 20,30 end example]</td>
<td>&lt;PolyLineSegment&gt; element</td>
</tr>
<tr>
<td></td>
<td>or l x,y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Horizontal Line</td>
<td>H x</td>
<td>Draws a horizontal line from the current endpoint to the specified x coordinate. [Example: H 90 end example]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>or h x</td>
<td></td>
<td>&lt;PolyLineSegment&gt; element</td>
</tr>
<tr>
<td>Vertical Line</td>
<td>v y</td>
<td>Draws a vertical line from the current endpoint to the specified y coordinate.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>or v y</td>
<td></td>
<td>&lt;PolyLineSegment&gt; element</td>
</tr>
<tr>
<td>Name</td>
<td>Syntax</td>
<td>Description</td>
<td>Non-Abbreviated Equivalent</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>Cubic Bézier Curve</td>
<td>C $x_1,y_1 \ x_2,y_2 \ x_3,y_3$ or c $x_1,y_1 \ x_2,y_2 \ x_3,y_3$</td>
<td>Draws a cubic Bézier curve from the current endpoint to the specified point ($x_3,y_3$) using the two specified control points ($x_1,y_1$ and $x_2,y_2$). The first control point determines the initial direction (tangent) of the curve, and the second determines the terminating direction (tangent) of the curve. [Example: C 100,200 200,400 300,200 end example]</td>
<td>&lt;PolyBezierSegment&gt; element</td>
</tr>
<tr>
<td>Quadratic Bézier Curve</td>
<td>Q $x_1,y_1 \ x_2,y_2$ or q $x_1,y_1 \ x_2,y_2$</td>
<td>Draws a quadratic Bézier curve from the current endpoint to the specified point ($x_2,y_2$) using the specified control point ($x_1,y_1$). [Example: q 100,200 300,200 end example]</td>
<td>&lt;PolyQuadraticBezierSegment&gt; element</td>
</tr>
<tr>
<td>Smooth Cubic Bézier Curve</td>
<td>S $x_1,y_1 \ x_2,y_2$ or s $x_1,y_1 \ x_2,y_2$</td>
<td>Draws a cubic Bézier curve from the current endpoint to the specified point ($x_2,y_2$). The first control point is assumed to be the reflection of the second control point of the previous command, relative to the current endpoint. If there is no previous command or if the previous command was not a Cubic Bézier Curve command or Smooth Cubic Bézier Curve command, the first control point is assumed to be coincident with the current endpoint. The second control point is specified by $x_1,y_1$. [Example: S 100,200 200,300 end example]</td>
<td>&lt;PolyBezierSegment&gt; element</td>
</tr>
<tr>
<td>Elliptical Arc</td>
<td>A $x_r,y_r \ r_x fArc fSweep x,y$ or a $x_r,y_r \ r_x fArc fSweep x,y$</td>
<td>Draws an elliptical arc from the current endpoint to the specified point ($x,y$). The size and orientation of the ellipse are defined by $x_r,y_r$. $r_x$ defines the x radius, $y_r$ defines the y radius, and $r_x$ defines the x-axis rotation in degrees, which indicates how the</td>
<td>&lt;ArcSegment&gt; element</td>
</tr>
</tbody>
</table>

[Example: v 90 end example]
## 11. Graphics

<table>
<thead>
<tr>
<th>Name</th>
<th>Syntax</th>
<th>Description</th>
<th>Non-Abbreviated Equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>ellipse is rotated relative to the current coordinate system. The center of the ellipse is calculated automatically. In most situations, four different arcs satisfy the specified constraints. $f_{Arc}$ and $f_{Sweep}$ indicate which arc to use. 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°. 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. 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]</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| Close | Z or z | Draws a straight line from the current endpoint to the first point of the current figure and then ends the figure. 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. <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 syntax:
Smooth Bézier curves specified with the abbreviated geometry syntax are basic cubic Bézier curves with an implied first control point. This control point is coincident with the endpoint of the previous segment unless the previous segment is also a Bézier curve. In this case, the first control point of the smooth Bézier curve is a reflection of the second control point of the previous curve segment around the start point of the smooth Bezier curve segment, as shown below.

Example 11–11. Smooth Bézier curve

In the following example, C₁ and C₂ represent the first and second control points of the first cubic Bézier curve segment, respectively. S₁ represents the implied first control point of the smooth Bézier curve segment. S₂ represents the specified control point of the smooth Bézier curve segment. I represents the inflection point around which control point S₁ is derived from control point C₂.

The above diagram is generated with the following markup:
11.2.3.2 Relative Commands and Curve Control Points

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

Example 11–12. Relative commands and curves

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

```xml
<Path Stroke="#000000" Data="M 50,20 L 150,20 C 250,75 170,130 120,100 C 70,70 90,110 130,160 Q 0,150 50,20" />
```
This markup describes the same shape, using relative commands:

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

```
end example]
```
12. Text

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

**element** Glyphs

<table>
<thead>
<tr>
<th>attributes</th>
<th>Name</th>
<th>Type</th>
<th>Use</th>
<th>Default</th>
<th>Fixed</th>
<th>Annotation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BidiLevel</td>
<td></td>
<td></td>
<td>0</td>
<td></td>
<td>Specifies the Unicode algorithm bidirectional nesting level. Even values imply left-to-right layout,</td>
</tr>
<tr>
<td>Property</td>
<td>Type</td>
<td>Required</td>
<td>Description</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------------</td>
<td>-----------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CaretStops</td>
<td>ST_CaretStops</td>
<td></td>
<td>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].</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DeviceFontName</td>
<td>ST_UnicodeString</td>
<td></td>
<td>Uniquely identifies a specific device font. The identifier is typically defined by a hardware vendor or font vendor.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fill</td>
<td>ST_RscRefColor</td>
<td></td>
<td>Describes the brush used to fill the shape of the rendered glyphs.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FontRenderingEmSize</td>
<td>ST_GEZero</td>
<td>required</td>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FontUri</td>
<td>xs:anyURI</td>
<td>required</td>
<td>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), 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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parameter</td>
<td>Type</td>
<td>Required</td>
<td>Description</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>--------------------</td>
<td>----------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OriginX</td>
<td>ST_Double</td>
<td>required</td>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OriginY</td>
<td>ST_Double</td>
<td>required</td>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IsSideways</td>
<td>ST_Boolean</td>
<td>false</td>
<td>Indicates that a glyph is turned on its side, with the origin being defined as the top center of the unturned glyph.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Indices</td>
<td>ST_Indices</td>
<td></td>
<td>Specifies a series of glyph indices and their attributes used for rendering the glyph run. If the UnicodeString attribute specifies an empty string (&quot;&quot; or &quot;{}&quot;) and the Indices attribute is not specified or is also empty, a consumer MUST generate an error [M5.2].</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UnicodeString</td>
<td>ST_UnicodeString</td>
<td></td>
<td>Contains the string of text rendered by the &lt;Glyphs&gt; element. The text is specified as Unicode code points.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>StyleSimulations</td>
<td>ST_StyleSimulations</td>
<td>None</td>
<td>Specifies a style simulation. Valid values are None, ItalicSimulation, BoldSimulation, and BoldItalicSimulation.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RenderTransform</td>
<td>ST_RscRefMatrix</td>
<td></td>
<td>Establishes a new coordinate frame for the glyph run specified by the &lt;Glyphs&gt; element. The render transform affects clip, opacity</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Property</td>
<td>Type</td>
<td>Value</td>
<td>Description</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------------</td>
<td>-------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Clip</td>
<td>ST_RscRefAbbrGeomF</td>
<td></td>
<td>Limits the rendered region of the element. Only portions of the &lt;Glyphs&gt; element that fall within the clip region (even partially clipped characters) produce marks on the page.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Opacity</td>
<td>ST_ZeroOne</td>
<td>1.0</td>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OpacityMask</td>
<td>ST_RscRef</td>
<td></td>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>ST_Name</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 hyperlinking.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FixedPage.NavigateUri</td>
<td>xs:anyURI</td>
<td></td>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>xml:lang</td>
<td></td>
<td></td>
<td>Specifies the default language used for the current element. The language is specified according to RFC 3066.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x:Key</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 [M5.3].</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The `<Glyphs>` element represents a run of uniformly-formatted text from a single font. It provides information necessary for accurate rendering and supports search and selection features in viewing consumers.

If the Fill property is not specified, the `<Glyphs>` element has no visible effect.

Some properties of the `<Glyphs>` element are composable, meaning that the markings rendered to the page are determined by a combination of the property and all the like-named properties of the `<Glyphs>` element’s parent and ancestor elements. For details, see §14.
12.1.1 Glyph Metrics

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

Figure 12–1. Glyph metrics

Figure 12–2. Upright (usually horizontal) glyph metrics
In general, glyphs within a font are either base glyphs or combining marks that can be attached to base glyphs. Base glyphs usually have an advance width that is non-zero, and a 0,0 offset vector. Combining marks usually have a zero advance width. The offset vector can be used to adjust the position of a combining mark and, therefore, can have a non-0,0 value for combining marks.

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

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

### 12.1.2 Mapping Code Units to Glyphs

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

The cluster map contains one entry for each UTF-16 code unit in the UnicodeString attribute. Each entry specifies the offset of the first glyph that represents the cluster of UTF-16 code units.
### 12.1.2.1 One-to-One Mappings

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

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

Each character in the word “file” is represented by a single glyph.

$$\begin{array}{c|c|c|c|c} & f & i & l & e \\
\text{UnicodeString} & 0066 & 0069 & 006c & 0065 \\
\text{ClusterMap} & 0 & 1 & 2 & 3 \\
\text{GlyphIndices} & 0049 & 004c & 004f & 0048 \\
\end{array}$$

*end example]*

### 12.1.2.2 Many-to-One Mappings

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

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

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

$$\begin{array}{c|c|c|c|c} & f & i & l & e \\
\text{UnicodeString} & 0066 & 0069 & 006c & 0065 \\
\text{ClusterMap} & 0 & 0 & 1 & 2 \\
\text{GlyphIndices} & 00bf & 004f & 0048 \\
\end{array}$$

*end example]*
12.1.2.3 One-to-Many Mappings

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

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

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

```
UnicodeString 0E20 0E31 0E33 0021
ClusterMap 0 1 2 4
GlyphIndices 0099 006A 007C 00C6 0004
(BrowalliaUPC font)

The markup appears as follows:

```xml
<Glyphs FontUri="../Resources/Fonts/browau.ttf" UnicodeString="ภ้ำ!
Indices="153;106,,16;(1:2)124;198;4"
OriginX="10" OriginY="60"
FontRenderingEmSize="70"
Fill="#000000"/>
```

The markup above is rendered as follows:

```
ก้า!
```

*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.

The markup appears as follows:

```xml
<Glyphs
  FontUri="../Resources/Fonts/latha.ttf"
  UnicodeString="போது"
  Indices="(2:3)94;76;88;(2:1)162"
  OriginX="10" OriginY="120"
  FontRenderingEmSize="40"
  Fill="#000000"/>
```

The markup above is rendered as follows:

![Glyphs](../Resources/Fonts/latha.ttf)

end example]
12.1.3 Indices Attribute

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

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

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

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

Table 12–3. Glyph specifications

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GlyphIndex</td>
<td>Index of the glyph (16-bit) in the physical font. The entry MAY be empty [M2.72], in which case the glyph index is determined by looking up the UTF-16 code unit in the font character map table. If there is not a one-to-one mapping between code units and the glyph indices, this entry MUST be specified [M5.5]. In cases where character-to-glyph mappings are not one-to-one, a cluster mapping specification precedes the glyph index (further described below).</td>
</tr>
<tr>
<td>AdvanceWidth</td>
<td>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</td>
</tr>
</tbody>
</table>
So that rounding errors do not accumulate, the advance MUST be calculated as the exact unrounded origin of the subsequent glyph minus the sum of the calculated (that is, rounded) advance widths of the preceding glyphs [M5.6].

The advance MUST be 0 or greater [M2.72]. The right-to-left writing direction can be specified using the 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. uOffset and vOffset are real numbers. Base glyphs generally have a glyph offset of 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. |

Example 12–5. Using indices to specify advance width

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

Indices = ";;;;;;;40"

end example]

12.1.3.1 Specifying Character-to-Glyph Mappings

A cluster map specification MAY precede the glyph specification for the first glyph of the cluster [M2.72]. Empty Indices attribute values indicate that the corresponding UTF-16 code unit within the Unicode string has a one-to-one relationship with the glyph index as specified by the character mapping table within the font.

Cluster maps that specify 0:n or n:0 mappings are invalid. See the glyph specification syntax above for details of how to specify cluster maps.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClusterCodeUnitCount</td>
<td>Number of UTF-16 code units that combine to form this cluster. One or more code units can be specified. Default value is 1.</td>
</tr>
<tr>
<td>ClusterGlyphCount</td>
<td>Number of glyph indices that combine to form this cluster. One or more indices can be specified. Default value is 1.</td>
</tr>
</tbody>
</table>
Example 12-6. Using the Indices attribute to specify glyph replacement for a cluster

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

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

12.1.4 UnicodeString Attribute

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

The standard XML escaping mechanisms are used to specify XML-reserved characters. An additional mechanism MUST be used to escape a UnicodeString attribute value that begins with an open brace (“{”) [M5.7]. In order to use an open brace at the beginning of the Unicode string, it MUST be escaped with a prefix of “{"” [M5.7]. If the UnicodeString attribute value starts with “{"”, consumers MUST ignore those first two characters in processing the Unicode string and in calculating index positions for the characters of the Unicode string [M5.7].

If the UnicodeString attribute specifies an empty string (“” or “{"”), and the Indices attribute is missing or is also empty, it MUST be treated as an error [M5.2]. 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 [M5.9].

Producers MAY include Unicode control marks in the Unicode string [O5.1]. 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 specification. 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 [S5.2]. In the absence of such information, consumers MUST treat Unicode control marks like ordinary characters and render the glyphs to which the Unicode control marks are mapped in the CMAP table [M5.10]. The resulting glyphs might produce an inappropriate rendering of the original Unicode string.

Producers MAY choose to generate UnicodeString attribute values that are not normalized by any Unicode-defined algorithm [O5.2]. 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 [M5.11]. See §9.1.7.5 for details and exceptions.

12.1.5 StyleSimulations Attribute

Synthetic style simulations can be applied to the shape of the glyphs by using the StyleSimulations attribute. Style simulations can be applied in addition to the designed style of a font. The default value for the StyleSimulations attribute is None, in which case the shapes of glyphs are not modified from their original design.
When the StyleSimulations value is specified as BoldSimulation, synthetic emboldening is applied by geometrically widening the strokes of glyphs by 1% of the em size, so that the centers of strokes remain at the same position. This leaves the baseline origin unmodified. The black box grows 1% all around for a total of 2% horizontal and 2% vertical. As a result, the character height and the advance width of each glyph are increased by 2% of the em size. Producers MUST lay out algorithmically emboldened glyphs using advance widths that are 2% of the em size larger than when not algorithmically emboldened [M5.12].

Consumers MUST implement the effect of algorithmic emboldening such that the black box of the glyph grows by 2% of the em size [M5.13]. 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 [M5.13].

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

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

12.1.6 IsSideways Attribute

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

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

Text from horizontal writing systems can be included in the column by using <Glyphs> elements without specifying IsSideways, or using a value of false for it. The rotated coordinate system makes them appear top to bottom on the page, but with the glyphs rotated to the right. If alternate vertical character representations are available in the font, the producer SHOULD use those and provide their glyph indices in the Indices attribute [S5.3].

12.1.6.1 Calculating Sideways Text Origin and Advance Width

The formulas below describe the method used to calculate each glyph’s nominal origin, which is used for positioning the glyphs on the fixed page and for calculating the default advance width for each glyph.
The origin is the top center of the unturned glyph. The \( x \) origin of the unturned glyph is calculated to be exactly one-half the advance width of the glyph, as specified in the horizontal metrics table of the font. This formula is expressed as follows (in pseudocode):

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

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

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

or:

\[
\text{TopOriginY} = \text{vorg.defaultVertOriginY}
\]

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

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

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

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

\[
\text{TopOriginY} = \text{os/2.sTypoAscender}
\]

\[
\text{Descender} = \text{abs}(\text{os/2.typoDescender})
\]

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

\[
\text{TopOriginY} = \text{hhea.Ascender}
\]

\[
\text{Descender} = \text{abs}(\text{hhea.Descender})
\]

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

\[
\text{AdvanceWidth} = \text{TopOriginY} + \text{Descender}
\]

12.1.6.2 \textbf{IsSideways and BidiLevel Effects on Glyph Positioning}

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

Producers MUST NOT specify text that is both right-to-left (BidiLevel attribute value of 1) and vertical (IsSideways attribute set to true) [M5.15].
### Table 12-5. *IsSideways* and *BidiLevel* effects on origin placement

<table>
<thead>
<tr>
<th>IsSideways</th>
<th>BidiLevel</th>
<th>Glyph origin</th>
<th>Direction of advance width and positive uOffset</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizontal</td>
<td>Left-to-right (0)</td>
<td>Left end of horizontal advance vector along Latin baseline</td>
<td>To the right</td>
</tr>
<tr>
<td>(false)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Horizontal</td>
<td>Right-to-left (1)</td>
<td>Right end of horizontal advance vector along Latin baseline</td>
<td>To the left</td>
</tr>
<tr>
<td>(false)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vertical</td>
<td>Left-to-right (0)</td>
<td>Top end of vertical advance vector through the glyph centerline</td>
<td>To the right</td>
</tr>
<tr>
<td>(true)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vertical</td>
<td>Right-to-left (1)</td>
<td>Invalid combination</td>
<td></td>
</tr>
<tr>
<td>(true)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example 12–7. Text with positive uOffset and vOffset Indices values

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

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

indices Adjustment Only

end example]

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

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

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

Right-to-Left Text

end example]
Example 12-9. Sideways text (IsSideways set to true)

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

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

end example]

Example 12-10. Vertical text

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

```xml
<Glyphs Fill="#000000" FontRenderingEmSize="48"
  OriginX="100" OriginY="100"
  UnicodeString="AFQ"
  Indices=";100,30,10;"
  IsSideways="true"
  FontUri="../Resources/Fonts/Arial.ttf"
  RenderTransform="0,1,-1,0,200,0" />
```
This example demonstrates a real-world usage of vertical text. Japanese text is shown below where the text is read down each column, from right to left across the page. The IsSideways attribute is set to true, thus rotating the each glyph 90° counter-clockwise. Then, the RenderTransform attribute (see §14.4) rotates the overall block of text 90° clockwise to achieve the final result of columns of text.

```xml
<Glyphs Fill="#000000" FontRenderingEmSize="24" OriginX="10" OriginY="10"
  UnicodeString="これは、縦書きの日本語テキストが"
  FontUri="../Resources/Fonts/msmincho.ttf" IsSideways="true"
  RenderTransform="0,1,-1,0,145,0"/>

<Glyphs Fill="#000000" FontRenderingEmSize="24" OriginX="10" OriginY="45"
  UnicodeString="どのように列で書かれるかの例です。"
  FontUri="../Resources/Fonts/msmincho.ttf" IsSideways="true"
  RenderTransform="0,1,-1,0,145,0"/>

<Glyphs Fill="#000000" FontRenderingEmSize="24" OriginX="10" OriginY="80"
  UnicodeString="テキストは縦に読み、一行ずつ進みます。"
  FontUri="../Resources/Fonts/msmincho.ttf" IsSideways="true"
  RenderTransform="0,1,-1,0,145,0"/>

<Glyphs Fill="#000000" FontRenderingEmSize="24" OriginX="10" OriginY="115"
  UnicodeString="他の言語も縦書きで書かれます。"
  FontUri="../Resources/Fonts/msmincho.ttf" IsSideways="true"
  RenderTransform="0,1,-1,0,145,0"/>
```
This markup is rendered as follows:

1. これは、縦書きの日本語テキストがどのように列で書かれるかの例です。
2. 他の言語も縦書きで書かれます。
12.1.7 DeviceFontName Attribute

Printer device fonts are specified by the DeviceFontName attribute. Device manufacturers define the values for this attribute. 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 [S5.4].

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

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

For producers, a <Glyphs> element with a specified device font name MUST have exactly one Indices glyph per code unit 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 and MUST render the embedded version of the font.
[M5.19]. This means that a device font cannot be used for characters outside the basic multilingual plane (BMP).

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

12.1.8 xml:lang Attribute

XPS Document consumers might need to override the default language for a specific run of glyphs, particularly in multilingual documents. 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 [M2.72]. For larger blocks of text, the producer MAY specify the xml:lang attribute on the <Canvas> element [M2.72].

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

12.1.9 CaretStops Attribute

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

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

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

Example 12–12. Using the CaretStops attribute to determine a valid caret stop position

Given the following attributes, the m in “example” is not a valid caret stop position:

 end example]

12.1.10 Optimizing Glyph Markup

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

12.1.10.1 Optimizing Glyph Indices Markup

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

[Example 12–12: Using the CaretStops attribute to determine a valid caret stop position]
1. 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.

2. 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 section §12.1.6.1, “Calculating Sideways Text Origin and Advance Width” (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 section §12.1.6.1, “Calculating Sideways Text Origin and Advance Width” (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

```xml
<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

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

This text is rendered as follows:
Simulated italic font...
end example]

Example 12–15. Kerning

<Canvas>

<!- "WAVE" without kerning -->

<Glyphs
  OriginX="35"
  OriginY="35"
  UnicodeString="WAVE (no kerning)"
  FontUri="../Resources/Fonts/Times.ttf"
  FontRenderingEmSize="20"
  Fill="#009900" />

<!- "WAVE" with kerning -->

<Glyphs
  OriginX="35"
  OriginY="70"
  UnicodeString="WAVE (with kerning)"
  Indices="88;59"
  FontUri="../Resources/Fonts/Times.ttf"
  FontRenderingEmSize="20"
  Fill="#009900" />

</Canvas>

This text is rendered as follows:

WAVE (no kerning)

WAVE (with kerning)

end example]

Example 12–16. Ligatures

<Canvas>

<!- "Open file" without "fi" ligature -->

<Glyphs
  OriginX="35"
  OriginY="35"
  UnicodeString="Open file (no ligature)"
  FontUri="../Resources/Fonts/Times.ttf"
  FontRenderingEmSize="20"
  Fill="#009900" />
This text is rendered as follows:

```
<Canvas>
<!-- "Open file" with "fi" ligature -->
<Glyphs
  OriginX="35"
  OriginY="70"
  UnicodeString="Open file (with ligature)"
  Indices=";;;;;(2:1)191"
  FontUri="../Resources/Fonts/Times.ttf"
  FontRenderingEmSize="20"
  Fill="#009900" />
</Canvas>
```

Open file (no ligature)

Open file (with ligature)

```
end example]
```

Example 12–17. Cluster maps

```
<Canvas>
<!-- "ёжик в тумане" using pre-composed "ё" -->
<Glyphs
  OriginX="35"
  OriginY="35"
  xml:lang="ru-RU"
  UnicodeString="ёжик в тумане"
  FontUri="../Resources/Fonts/Times.ttf"
  FontRenderingEmSize="20"
  Fill="#009900" />

<!-- "ёжик в тумане" using composition of "е" and diaeresis -->
<Glyphs
  OriginX="35"
  OriginY="70"
  xml:lang="ru-RU"
  UnicodeString="ёжик в тумане"
  Indices="(1:2)72;142,0,-40"
  FontUri="../Resources/Fonts/Times.ttf"
  FontRenderingEmSize="20"
  Fill="#009900" />

<!-- "ёжик в тумане" Forced rendering right-to-left showing
combining mark in logical order -->
<Glyphs
```
This text is rendered as follows:

ёжик в тумане
ёжик в тумане
енамут в кижё

end example]

12.2 <Glyphs.Fill> Element

element Glyphs.Fill

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

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

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solid color brush</td>
<td>Fills a region with a solid color</td>
</tr>
<tr>
<td>Image brush</td>
<td>Fills a region with an image</td>
</tr>
<tr>
<td>Visual brush</td>
<td>Fills a region with a drawing</td>
</tr>
<tr>
<td>Linear gradient brush</td>
<td>Fills a region with a linear gradient</td>
</tr>
<tr>
<td>Radial gradient brush</td>
<td>Fills a region with a radial gradient</td>
</tr>
</tbody>
</table>
13.1 <SolidColorBrush> Element

element SolidColorBrush

<table>
<thead>
<tr>
<th>attribute</th>
<th>Name</th>
<th>Type</th>
<th>Use</th>
<th>Default</th>
<th>Fixed</th>
<th>Annotation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Opacity</td>
<td>ST_ZeroOne</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></td>
<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>
<tr>
<td></td>
<td>Color</td>
<td>ST_Color</td>
<td>required</td>
<td></td>
<td></td>
<td>Specifies the color for filled elements. An sRGB color value specified as a 6-digit hexadecimal number (#RRGGBB) or an extended color.</td>
</tr>
</tbody>
</table>

Fills defined geometric regions with a solid color.

The <SolidColorBrush> element is used to fill defined geometric regions with a solid color. If there is an alpha component of the color, it is combined in a multiplicative way with the corresponding Opacity attribute.
Example 13–1. `<SolidColorBrush>` usage

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

```xml
<Path Stroke="#000000">
  <Path.Fill>
    <SolidColorBrush Color="#00FFFF" />
  </Path.Fill>
  <Path.Data>
    <PathGeometry>
      <PathFigure StartPoint="20,20" IsClosed="true">
        <PolyLineSegment Points="250,20 135,150" />
      </PathFigure>
    </PathGeometry>
  </Path.Data>
</Path>
```

This markup is rendered as follows:

![Triangle filled with light blue color]

end example]}

13.2 `<ImageBrush>` Element

element `ImageBrush`

13. Brushes

<table>
<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>ST_ZeroOne</td>
<td></td>
<td>1.0</td>
<td></td>
<td>Defines the uniform transparency of the brush fill. Values range from 0 (fully transparent) to 1 (fully opaque), inclusive. Values outside of this range are invalid.</td>
</tr>
<tr>
<td>x:Key</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Specifies a name for a resource in a resource dictionary. x:Key MUST be present when the current element is defined in a resource dictionary. x:Key MUST NOT be specified outside of a resource dictionary [M6.2].</td>
</tr>
<tr>
<td>Transform</td>
<td>ST_RscRefMatrix</td>
<td></td>
<td></td>
<td></td>
<td>Describes the matrix transformation applied to the coordinate space of the brush. The Transform property is concatenated with the current effective render transform to yield an effective render transform local to the brush. The viewport for the brush is transformed using the local effective render transform.</td>
</tr>
<tr>
<td>Viewbox</td>
<td>ST_ViewBox</td>
<td>required</td>
<td></td>
<td></td>
<td>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.</td>
</tr>
</tbody>
</table>
The `<ImageBrush>` element is used to fill a region with an image. The image is defined in a coordinate space specified by the resolution of the image. The image MUST refer to a JPEG, PNG, TIFF 6.0, or Windows Media Photo image part within the XPS Document package [M6.3]. For more information, see §9.1.5. A URI part name for the image is specified using the `ImageSource` attribute.

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

*Example 13–2. `<ImageBrush>` usage*

The following markup describes an image on a canvas.

```
<Canvas>
  <Path Stroke="#008000">
    <Path.Fill>
      <ImageBrush
        ImageSource="dog.jpg"
        TileMode="None"
        Viewbox="0,0,270,423"
        ViewboxUnits="Absolute"
        Viewport="25,25,125,185"
      />
  </Path.Fill>
</Path>
```
This markup produces the following results:
13.3 `<VisualBrush>` Element

element `VisualBrush`

<table>
<thead>
<tr>
<th>attributes</th>
<th>Name</th>
<th>Type</th>
<th>Use</th>
<th>Default</th>
<th>Fixed</th>
<th>Annotation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Opacity</td>
<td><code>ST_ZeroOne</code></td>
<td></td>
<td>1.0</td>
<td></td>
<td>Defines the uniform transparency of the brush fill. Values range from 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(fully transparent) to 1 (fully opaque), inclusive. Values outside of this</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>range are invalid.</td>
</tr>
<tr>
<td></td>
<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</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>present when the current element is defined in a resource dictionary. x:Key</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MUST NOT be specified outside of a resource dictionary [M6.4].</td>
</tr>
<tr>
<td></td>
<td>Transform</td>
<td><code>ST_RscRefMatrix</code></td>
<td></td>
<td></td>
<td></td>
<td>Describes the matrix transformation applied to the coordinate space of the</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>brush. The Transform property is concatenated with the current effective</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>render transform to yield an effective render transform local to the brush.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>The viewport for the brush is transformed using that local effective render</td>
</tr>
<tr>
<td></td>
<td>Viewbox</td>
<td><code>ST_Viewport</code></td>
<td>required</td>
<td></td>
<td></td>
<td>Specifies the position and dimensions of the brush’s source content.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Specifies four comma-separated real numbers (x, y, Width, Height),</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>depending on the viewbox units.</td>
</tr>
</tbody>
</table>
The `<VisualBrush>` element is used to fill a region with a drawing. The drawing can be specified as either a `<VisualBrush.Visual>` property element or as a resource reference. Drawing content can include exactly one `<Canvas>`, `<Path>`, or `<Glyphs>` element and that element’s child and descendant elements.

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

### 13.3.1 `<VisualBrush.Visual>` Element

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

```xml
<Path>
  <Path.Fill>
    <VisualBrush
      Viewbox="0,0,1,1"
      Viewport="50,50,100,100"
      ViewportUnits="Absolute"
      ViewboxUnits="Absolute"
      TileMode="Tile">
      <VisualBrush.Visual>
        <Path>
          <Path.Fill>
            <SolidColorBrush Color="#FF0000" />  
          </Path.Fill>
          <Path.Data>
            <PathGeometry>
              <PathFigure StartPoint="0,0.5" IsClosed="true">
                <PolyLineSegment Points="0.5,0 1.0,0.5 0.5,1.0" />
              </PathFigure>
            </PathGeometry>
          </Path.Data>
        </Path>
      </VisualBrush.Visual>
    </VisualBrush>
  </Path.Fill>
  <Path.Data>
    <PathGeometry>
      <PathFigure StartPoint="50,50" IsClosed="true">
        <PolyLineSegment Points="350,50 350,350 50,350 50,350" />
      </PathFigure>
    </PathGeometry>
  </Path.Data>
</Path>
```
This markup produces the following result:

```
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

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

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:

\[
\text{SourceLeft} = \text{HorizontalImageResolution} \times \text{Viewbox.Left} / 96 \\
\text{SourceTop} = \text{VerticalImageResolution} \times \text{Viewbox.Top} / 96 \\
\text{SourceWidth} = \text{HorizontalImageResolution} \times \text{Viewbox.Width} / 96 \\
\text{SourceHeight} = \text{VerticalImageResolution} \times \text{Viewbox.Height} / 96
\]

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.
Example 13–4. ViewboxUnits and ViewportUnits attribute usage

The following markup contains an image brush:

```
13.4.1.1 Viewbox and Viewport Examples

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

Example 13–5. Tiling brush base image and rendering
```

```
<Example>

The following markup contains an image brush:

```
</Example>

```
This markup is rendered as follows:

```xml
<Path Fill="#CCCC66" Data="M 150,0 L 300,150 L 150,300 L 0,150 Z" />
<Path Data="M 150,0 L 300,150 L 150,300 L 0,150 Z">
  <Path.Fill>
    <VisualBrush Viewbox="0,0,1,1" Viewport="125,125,50,50" ViewboxUnits="Absolute" ViewportUnits="Absolute" TileMode="Tile">
      <VisualBrush.Visual>
        <Canvas>
          <Path Fill="#333399" Data="M 0.1,0.1 L 0.9,0.1 L 0.9,0.9 L 0.1,0.9 Z" />
          <Path Fill="#FFFF00" Data="M 0.1,0.35 L 0.35,0.1 L 0.6,0.35 L 0.35,0.6 Z" />
        </Canvas>
      </VisualBrush.Visual>
    </VisualBrush>
  </Path.Fill>
</Path>
```

Example 13–6. Tiling brush Viewport adjustments

By adjusting the viewport, the position of the tiles within the image can be changed:
This markup is rendered as follows:

```
<Path Fill="#CCCC66" Data="M 150,0 L 300,150 L 150,300 L 0,150 Z" />
<Path Data="M 150,0 L 300,150 L 150,300 L 0,150 Z">
  <Path.Fill>
    <VisualBrush Viewbox="0.25,0.25,0.75,0.75" Viewport="150,75,50,50" ViewboxUnits="Absolute" ViewportUnits="Absolute" TileMode="Tile">
      <VisualBrush.Visual>
        <Canvas>
          <Path Fill="#333399" Data="M 0.1,0.1 L 0.9,0.1 L 0.9,0.9 L 0.1,0.9 Z" />
          <Path Fill="#FFFF00" Data="M 0.1,0.35 L 0.35,0.1 L 0.6,0.35 L 0.35,0.6 Z" />
        </Canvas>
      </VisualBrush.Visual>
    </VisualBrush>
  </Path.Fill>
</Path>
```

Example 13–7. Tiling brush viewbox adjustments

The following markup uses a smaller window on the viewbox to zoom in on each tile:
Example 13-8. Image brush with a Viewbox larger than the image

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

```xml
<Path Fill="#CCCCCC" Data="M 10,10 L 265,10 L 265,125 L 10,125 Z" />
<Path Stroke="#803333" StrokeThickness="3"
   Data="M 25,25 L 250,25 L 250,200 L 25,200 Z">
   <Path.Fill>
      <ImageBrush ImageSource="../Resources/Images/dog.jpg"
         TileMode="Tile"
         Viewbox="-10,-10,290,443" ViewboxUnits="Absolute"
         Viewport="50,50,90,125" ViewportUnits="Absolute"/>
   </Path.Fill>
</Path>
```

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

```
<!-- Draw background diamond to show where fill affects background -->
<Path Fill="#CCCC66" Data="M 200,0 L 400,200 L 200,400 L 0,200 Z" />
<Path Data="M 200,0 L 400,200 L 200,400 L 0,200 Z">
  <Path.Fill>
    <ImageBrush ImageSource="newspaper.png"
      Viewbox="0,0,350,284"
      Viewport="200,100,87,71"
      ViewportUnits="Absolute"
      ViewboxUnits="Absolute"
      TileMode="None" />
  </Path.Fill>
</Path>
```

### 13.4.2 TileMode Attribute

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

#### 13.4.2.1 None

In this mode, only the single base tile is drawn. The remaining area is left transparent.
This markup is rendered as follows:

end example]
Example 13-10. Visual brush with TileMode value of None

```xml
<!-- Draw background diamond to show where fill affects background -->
<Path Fill="#CCCC66" Data="M 200,0 L 400,200 L 200,400 L 0,200 Z" />
<Path Data="M 200,0 L 400,200 L 200,400 L 0,200 Z">
  <Path.Fill>
    <VisualBrush Viewbox="0,0,1,1"
                 Viewport="200,133,67,67"
                 ViewboxUnits="Absolute"
                 ViewportUnits="Absolute"
                 TileMode="None">
      <VisualBrush.Visual>
        <Canvas>
          <Path Fill="#333399" Data="M 0.1,0.1 L 0.9,0.1 L 0.9,0.9 L 0.1,0.9 Z" />
          <Path Fill="#FFFF00" Data="M 0.1,0.35 L 0.35,0.1 L 0.6,0.35 L 0.35,0.6 Z" />
        </Canvas>
      </VisualBrush.Visual>
    </VisualBrush>
  </Path.Fill>
</Path>
```

This markup is rendered as follows:

![Visual brush with TileMode value of None](image)

end example]
13.4.2.2 Tile

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

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

<!-- Draw background diamond to show where fill affects background -->
<Path Fill="#CCCC66" Data="M 200,0 L 400,200 L 200,400 L 0,200 Z" />
<Path Data="M 200,0 L 400,200 L 200,400 L 0,200 Z">
  <Path.Fill>
    <ImageBrush>
      ImageSource="newspaper.png"
      Viewbox="0,0,350,284"
      Viewport="200,100,87,71"
      ViewportUnits="Absolute"
      ViewboxUnits="Absolute"
      TileMode="Tile" />
  </Path.Fill>
</Path>

This markup is rendered as follows:

end example]
Example 13-12. Visual brush with a TileMode value of Tile

<!-- Draw background diamond to show where fill affects background -->
<Path Fill="#CCCCCC" Data="M 200,0 L 400,200 L 200,400 L 0,200 Z" />
<Path Data="M 200,0 L 400,200 L 200,400 L 0,200 Z">
<Path.Fill>
<VisualBrush Viewbox="0,0,1,1"
Viewport="200,133,67,67"
ViewportUnits="Absolute"
ViewboxUnits="Absolute"
TileMode="Tile">
<VisualBrush.Visual>
<Canvas>
<Path Fill="#333399" Data="M 0.1,0.1 L 0.9,0.1 L 0.9,0.9 L 0.1,0.9 Z" />
<Path Fill="#FFFF00" Data="M 0.1,0.35 L 0.35,0.1 L 0.6,0.35 L 0.35,0.6 Z" />
</Canvas>
</VisualBrush.Visual>
</VisualBrush>
</Path.Fill>
</Path>

This markup is rendered as follows:

[end example]
13. Brushes

13.4.2.3 FlipX

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

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

<!-- Draw background diamond to show where fill affects background -->
<Path Fill="#CCCC66" Data="M 200,0 L 400,200 L 200,400 L 0,200 Z" />
<Path Data="M 200,0 L 400,200 L 200,400 L 0,200 Z">
  <Path.Fill>
    <ImageBrush ImageSource="newspaper.png"
                Viewbox="0,0,350,284"
                Viewport="200,100,87,71"
                ViewportUnits="Absolute"
                ViewboxUnits="Absolute"
                TileMode="FlipX" />
  </Path.Fill>
</Path>

This markup is rendered as follows:

end example]
Example 13-14. Visual brush with a TileMode value of FlipX

```xml
<!-- Draw background diamond to show where fill affects background -->
<Path Fill="#CCCC66" Data="M 200,0 L 400,200 L 200,400 L 0,200 Z" />
<Path Data="M 200,0 L 400,200 L 200,400 L 0,200 Z">
  <Path.Fill>
    <VisualBrush
      Viewbox="0,0,1,1"
      Viewport="200,133,67,67"
      ViewboxUnits="Absolute"
      ViewportUnits="Absolute"
      TileMode="FlipX">
      <VisualBrush.Visual>
        <Canvas>
          <Path Fill="#333399" Data="M 0.1,0.1 L 0.9,0.1 L 0.9,0.9 L 0.1,0.9 Z" />
          <Path Fill="#FFFF00" Data="M 0.1,0.35 L 0.35,0.1 L 0.6,0.35 L 0.35,0.6 Z" />
        </Canvas>
      </VisualBrush.Visual>
    </VisualBrush>
  </Path.Fill>
</Path>
```

This markup is rendered as follows:

![Visual brush with a TileMode value of FlipX](image)

end example]
13. Brushes

13.4.2.4 FlipY

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

Example 13–15. Image brush with a TileMode value of FlipY

<!-- Draw background diamond to show where fill affects background -->
<Path Fill="#CCCC66" Data="M 200,0 L 400,200 L 200,400 L 0,200 Z" />
<Path Data="M 200,0 L 400,200 L 200,400 L 0,200 Z">
  <Path.Fill>
    <ImageBrush
      ImageSource="newspaper.png"
      Viewbox="0,0,350,284"
      Viewport="200,100,87,71"
      ViewportUnits="Absolute"
      ViewboxUnits="Absolute"
      TileMode="FlipY" />
  </Path.Fill>
</Path>

This markup is rendered as follows:
Example 13-16. Visual Brush with a TileMode value of FlipY

<!-- Draw background diamond to show where fill affects background -->
<Path Fill="#CCCC66" Data="M 200,0 L 400,200 L 200,400 L 0,200 Z" />
<Path Data="M 200,0 L 400,200 L 200,400 L 0,200 Z"/>
<Path.Fill>
  <VisualBrush
    Viewbox="0,0,1,1"
    Viewport="200,133,67,67"
    ViewboxUnits="Absolute"
    ViewportUnits="Absolute"
    TileMode="FlipY">
    <VisualBrush.Visual>
      <Canvas>
        <Path Fill="#333399" Data="M 0.1,0.1 L 0.9,0.1 L 0.9,0.9 L 0.1,0.9 Z" />
        <Path Fill="#FFFF00" Data="M 0.1,0.35 L 0.35,0.1 L 0.6,0.35 L 0.35,0.6 Z" />
      </Canvas>
    </VisualBrush.Visual>
  </VisualBrush>
</Path.Fill>
</Path>

This markup is rendered as follows:

![Visual Brush Example]
13.4.2.5 FlipXY

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

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

```xml
<!-- Draw background diamond to show where fill affects background -->
<Path Fill="#CCCC66" Data="M 200,0 L 400,200 L 200,400 L 0,200 Z" />
<Path Data="M 200,0 L 400,200 L 200,400 L 0,200 Z">
  <Path.Fill>
    <ImageBrush ImageSource="newspaper.png" Viewbox="0,0,350,284" Viewport="200,100,87,71" ViewportUnits="Absolute" ViewboxUnits="Absolute" TileMode="FlipXY" />
  </Path.Fill>
</Path>
```

This markup is rendered as follows:
Example 13–18. Visual brush with a TileMode value of FlipXY

```xml
<!-- Draw background diamond to show where fill affects background -->
<Path Fill="#CCCC66" Data="M 200,0 L 400,200 L 200,400 L 0,200 Z" />
<Path Data="M 200,0 L 400,200 L 200,400 L 0,200 Z">
  <Path.Fill>
    <VisualBrush
      Viewbox="0,0,1,1"
      Viewport="200,133,67,67"
      ViewboxUnits="Absolute"
      ViewportUnits="Absolute"
      TileMode="FlipXY">
      <VisualBrush.Visual>
        <Canvas>
          <Path Fill="#333399" Data="M 0.1,0.1 L 0.9,0.1 L 0.9,0.9 L 0.1,0.9 Z" />
          <Path Fill="#FFFF00" Data="M 0.1,0.35 L 0.35,0.1 L 0.6,0.35 L 0.35,0.6 Z" />
        </Canvas>
      </VisualBrush.Visual>
    </VisualBrush>
  </Path.Fill>
</Path>
```

This markup is rendered as follows:
13.5 `<LinearGradientBrush>` Element

**element** `LinearGradientBrush`

**attributes**

<table>
<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><code>ST_ZeroOne</code></td>
<td></td>
<td>1.0</td>
<td></td>
<td>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.</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 [M6.5].</td>
</tr>
<tr>
<td>ColorInterpolationMode</td>
<td><code>ST_ClrIntMode</code></td>
<td></td>
<td><code>SRgbLinearInterpolation</code></td>
<td></td>
<td>Specifies the gamma function for color interpolation. The gamma adjustment should not be applied to the alpha component, if specified. Valid values are <code>SRgbLinearInterpolation</code> and <code>ScRgbLinearInterpolation</code>.</td>
</tr>
<tr>
<td>SpreadMethod</td>
<td>ST_SpreadMethod</td>
<td>Pad</td>
<td>Describes how the brush should fill the content area outside of the primary, initial gradient area. Valid values are Pad, Reflect and Repeat.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------</td>
<td>--------------</td>
<td>--------------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MappingMode</td>
<td>ST_MappingMode</td>
<td>required</td>
<td>Specifies that the start point and end point are defined in the effective coordinate space (includes the Transform attribute of the brush).</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transform</td>
<td>ST_RscRefMatrix</td>
<td></td>
<td>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.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>StartPoint</td>
<td>ST_Point</td>
<td>required</td>
<td>Specifies the starting point of the linear gradient.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EndPoint</td>
<td>ST_Point</td>
<td>required</td>
<td>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.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Annotation:** Fills a region with a linear gradient.

1. The `<LinearGradientBrush>` element is used to specify a linear gradient brush along a vector. For details about computing a linear gradient, see Section 18.3.

2. Example 13–19. `<LinearGradientBrush>` usage

3. The following markup describes a page with a rectangular path that is filled with a linear gradient:

   ```xml
   <Path>
     <Path.Fill>
   ```
13. Brushes

This markup is rendered as follows:

```xml
<LinearGradientBrush
    MappingMode="Absolute"
    StartPoint="0,0"
    EndPoint="300,300">
    <LinearGradientBrush.GradientStops>
        <GradientStop Color="#FFFF00" Offset="0" />
        <GradientStop Color="#0000FF" Offset="1" />
    </LinearGradientBrush.GradientStops>
</LinearGradientBrush>
</Path.Fill>
<path data="M 150,0 L 300,150 L 150,300 L 0,150 Z">
    <Path.Fill>
        <LinearGradientBrush MappingMode="Absolute"
            StartPoint="0,0"
            EndPoint="300,300">
            <LinearGradientBrush.GradientStops>
                <GradientStop Color="#FFFF00" Offset="0" />
                <GradientStop Color="#0000FF" Offset="1" />
            </LinearGradientBrush.GradientStops>
        </LinearGradientBrush>
    </Path.Fill>
</Path>
```

end example]

### 13.5.1 SpreadMethod Attribute

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

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

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

```xml
<path data="M 150,0 L 300,150 L 150,300 L 0,150 Z">
    <Path.Fill>
        <LinearGradientBrush MappingMode="Absolute"
            StartPoint="0,0"
            EndPoint="300,300">
            <LinearGradientBrush.GradientStops>
                <GradientStop Color="#FFFF00" Offset="0" />
                <GradientStop Color="#0000FF" Offset="1" />
            </LinearGradientBrush.GradientStops>
        </LinearGradientBrush>
    </Path.Fill>
</Path>
```
Example 13–21. Linear gradient brush with a SpreadMethod value of Reflect

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

```xml
<Path Data="M 150,0 L 300,150 L 150,300 L 0,150 Z">
  <LinearGradientBrush MappingMode="Absolute"
    StartPoint="120,0"
    EndPoint="180,0"
    SpreadMethod="Reflect">
    <LinearGradientBrush.GradientStops>
      <GradientStop Color="#FFFF00" Offset="0.0" />
      <GradientStop Color="#0000FF" Offset="1.0" />
    </LinearGradientBrush.GradientStops>
  </LinearGradientBrush>
</Path.Fill>
</Path>
```

This markup is rendered as follows:

![Rendered Example Image]

end example]
This markup is rendered as follows:

![Example](image)

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

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

```xml
<Path Data="M 150,0 L 300,150 L 150,300 L 0,150 Z">
  <Path.Fill>
    <LinearGradientBrush MappingMode="Absolute"
      StartPoint="120,0"
      EndPoint="180,0"
      SpreadMethod="Repeat">
      <LinearGradientBrush.GradientStops>
        <GradientStop Color="#FFFF00" Offset="0.0" />
        <GradientStop Color="#0000FF" Offset="1.0" />
      </LinearGradientBrush.GradientStops>
    </LinearGradientBrush>
  </Path.Fill>
</Path>
```
13.5.2 <LinearGradientBrush.GradientStops> Element

The `<LinearGradientBrush.GradientStops>` property element specifies a collection of gradient stops that comprise the linear gradient. For more information, see §13.7.
## 13.6 `<RadialGradientBrush>` Element

**element** `RadialGradientBrush`

---

### Attributes

<table>
<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><code>ST_ZeroOne</code></td>
<td>1.0</td>
<td></td>
<td></td>
<td>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.</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 [M6.6].</td>
</tr>
<tr>
<td>ColorInterpolationMode</td>
<td><code>ST_ClrIntMode</code></td>
<td><code>SRgbLinear Interpolation</code></td>
<td></td>
<td></td>
<td>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</td>
</tr>
<tr>
<td>Property</td>
<td>Type</td>
<td>Required</td>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------</td>
<td>----------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SpreadMethod</td>
<td>ST_SpreadMethod</td>
<td>Pad</td>
<td>Describes how the brush should fill the content area outside of the primary, initial gradient area. Valid values are Pad, Reflect and Repeat.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MappingMode</td>
<td>ST_MappingMode</td>
<td>Absolute</td>
<td>Specifies that center, x radius, and y radius are defined in the effective coordinate space (includes the Transform attribute of the brush).</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transform</td>
<td>ST_RscRefMatrix</td>
<td></td>
<td>Describes the matrix transformation applied to the coordinate space of the brush. The Transform property is concatenated with the current effective render transform to yield an effective render transform local to the brush. The ellipse defined by the center, gradient origin, x radius, and y radius values is transformed using the local effective render transform.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Center</td>
<td>ST_Point</td>
<td>required</td>
<td>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.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GradientOrigin</td>
<td>ST_Point</td>
<td>required</td>
<td>Specifies the origin point of the radial gradient.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RadiusX</td>
<td>ST_GEZero</td>
<td>required</td>
<td>Specifies the radius in the x dimension of the ellipse which defines the radial gradient.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RadiusY</td>
<td>ST_GEZero</td>
<td>required</td>
<td>Specifies the radius in the y dimension of the ellipse which defines the radial gradient.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Annotation:* Fills a region with a radial gradient.
Radial gradient brushes are similar to linear gradient brushes. However, whereas a linear gradient brush has a start point and end point to define the gradient vector, a radial gradient brush has an ellipse (defined by the center, x radius, and y radius) and a gradient origin. The ellipse defines the end point of the gradient. In other words, a gradient stop with an offset at 1.0 defines the color at the circumference of the ellipse. The gradient origin defines the center of the gradient. A gradient stop with an offset at 0.0 defines the color at the gradient origin.

For details about computing a radial gradient, see §18.3.3.
Example 13–23. A radial gradient brush

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

Example 13–24. RadialGradientBrush usage

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

```xml
<Path>
  <Path.Fill>
    <RadialGradientBrush MappingMode="Absolute"
      Center="30,150"
      GradientOrigin="30,150"
      RadiusX="250"
      RadiusY="250">
      <RadialGradientBrush.GradientStops>
        <GradientStop Color="#FFFF00" Offset="0" />
        <GradientStop Color="#0000FF" Offset="1" />
      </RadialGradientBrush.GradientStops>
    </RadialGradientBrush>
  </Path.Fill>
  <Path.Data>
    <PathGeometry>
      <PathFigure StartPoint="0,0" IsClosed="true">
        <PolyLineSegment Points="300,0 300,300 0,300" />
      </PathFigure>
    </PathGeometry>
  </Path.Data>
</Path>
```
13.6.1 SpreadMethod Attribute

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

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

In the following markup, the last color is used to cover the fill area outside the ellipse.
This markup is rendered as follows:

```xml
<Path Data="M 150,0 L 300,150 L 150,300 L 0,150 Z">
  <Path.Fill>
    <RadialGradientBrush MappingMode="Absolute"
      Center="150,150"
      GradientOrigin="125,125"
      RadiusX="60"
      RadiusY="60"
      SpreadMethod="Reflect">
      <RadialGradientBrush.GradientStops>
        <GradientStop Color="#FFFF00" Offset="0.0" />
        <GradientStop Color="#0000FF" Offset="1.0" />
      </RadialGradientBrush.GradientStops>
    </RadialGradientBrush>
  </Path.Fill>
</Path>
```

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

In the following markup, the gradient stops are replayed in reverse order repeatedly to cover the fill area.
This markup is rendered as follows:

```xml
<Path Data="M 150,0 L 300,150 L 150,300 L 0,150 Z">
  <Path.Fill>
    <RadialGradientBrush MappingMode="Absolute"
      Center="150,150"
      GradientOrigin="125,125"
      RadiusX="60"
      RadiusY="60"
      SpreadMethod="Repeat">
      <RadialGradientBrush.GradientStops>
        <GradientStop Color="#FFFF00" Offset="0.0" />
        <GradientStop Color="#0000FF" Offset="1.0" />
      </RadialGradientBrush.GradientStops>
    </RadialGradientBrush>
  </Path.Fill>
</Path>
```

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

In the following markup, the gradient stops are repeated in order until the fill area is covered.
This markup is rendered as follows:

```
end example]
```

### 13.6.2 `<RadialGradientBrush.GradientStops>` Element

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

The `<GradientStop>` element is used by both the `<LinearGradientBrush>` and `<RadialGradientBrush>` elements to define the location and range of color progression for rendering a gradient.

For linear gradient brushes, the offset value of 0.0 is mapped to the start point of the gradient, and the offset value of 1.0 is mapped to the end point. Intermediate offset values are interpolated between these two points to determine their location.

For radial gradient brushes, the offset value of 0.0 is mapped to the gradient origin location. The offset value of 1.0 is mapped to the circumference of the ellipse as determined by the center, x radius, and y radius. Offsets between 0.0 and 1.0 are positioned at a location interpolated between these points.

For full details of rendering of gradient brushes, including handling of offsets, please see §18.3.
### 13.8 Using a Brush as an Opacity Mask

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

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

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

An opacity mask always has a brush child element.

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

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

```xml
<FixedPage Height="1056" Width="816" xml:lang="en-US">
  <Glyphs OriginX="25" OriginY="50" UnicodeString="This is a fading text example."
    FontUri="../Resources/Fonts/Times.TTF" FontRenderingEmSize="32">
    <Glyphs.OpacityMask>
      <LinearGradientBrush StartPoint="25,0" EndPoint="450,0"
        MappingMode="Absolute">
        <LinearGradientBrush.GradientStops>
          <GradientStop Color="#FF000000" Offset="0" />
          <GradientStop Color="#00000000" Offset="1" />
        </LinearGradientBrush.GradientStops>
      </LinearGradientBrush>
    </Glyphs.OpacityMask>
    <Glyphs.Fill>
      <SolidColorBrush Color="#000000" />
    </Glyphs.Fill>
  </Glyphs>
</FixedPage>
```
This markup is rendered as follows:

This is a fading text example.

end example]
Example 13–29. Opacity mask with radial gradient

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

```xml
<FixedPage Width="816" Height="1056" xml:lang="en-US">
  <Path>
    <Path.OpacityMask>
      <RadialGradientBrush MappingMode="Absolute"
        Center="200,300"
        GradientOrigin="200,300"
        RadiusX="200"
        RadiusY="300">
        <RadialGradientBrush.GradientStops>
          <GradientStop Color="#FF000000" Offset="0" />
          <GradientStop Color="#20000000" Offset="1" />
        </RadialGradientBrush.GradientStops>
      </RadialGradientBrush>
    </Path.OpacityMask>
    <Path.Fill>
      <ImageBrush Viewbox="0,0,400,600"
        ViewboxUnits="Absolute"
        Viewport="0,0,400,600"
        ViewportUnits="Absolute"
        TileMode="None"
        ImageSource="images/jpeg3.jpg" />
    </Path.Fill>
    <Path.Data>
      <PathGeometry>
        <PathFigure StartPoint="0,0" IsClosed="true">
          <PolyLineSegment Points="400,0 400,600 0,600" />
        </PathFigure>
      </PathGeometry>
    </Path.Data>
  </Path>
</FixedPage>
```
This markup is rendered as follows:

\end{example}
# 14. Common Properties

Several XPS Document elements share property attributes and elements as summarized in Table 14–1 and Table 14–2 and detailed in the following sections. Other than the Name, FixedPage.NavigateUri, and xml:lang attributes, these properties compose their results from parent to child, as described in §18.4.1.

<table>
<thead>
<tr>
<th>Name</th>
<th>Applies to</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clip</td>
<td>&lt;Canvas&gt; &lt;Glyphs&gt; &lt;Path&gt;</td>
<td>Restricts the region to which a brush can be applied.</td>
</tr>
<tr>
<td>Opacity</td>
<td>&lt;Canvas&gt; &lt;Glyphs&gt; &lt;ImageBrush&gt; &lt;LinearGradientBrush&gt; &lt;Path&gt; &lt;RadialGradientBrush&gt; &lt;SolidColorBrush&gt; &lt;VisualBrush&gt;</td>
<td>Defines the uniform transparency of the element.</td>
</tr>
<tr>
<td>OpacityMask</td>
<td>&lt;Canvas&gt; &lt;Glyphs&gt; &lt;Path&gt;</td>
<td>Specifies a mask of alpha values.</td>
</tr>
<tr>
<td>RenderTransform</td>
<td>&lt;Canvas&gt; &lt;Glyphs&gt; &lt;Path&gt;</td>
<td>Establishes a new coordinate space through the use of an affine matrix transformation. For more information, see §14.4.</td>
</tr>
<tr>
<td>Transform</td>
<td>&lt;ImageBrush&gt; &lt;LinearGradientBrush&gt; &lt;PathGeometry&gt; &lt;RadialGradientBrush&gt; &lt;VisualBrush&gt;</td>
<td>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.</td>
</tr>
<tr>
<td>Name</td>
<td>&lt;Canvas&gt; &lt;FixedPage&gt; &lt;Glyphs&gt; &lt;Path&gt;</td>
<td>Defines a hyperlink target or identifies an element uniquely for document structure markup to reference. For more information, see §16.2.</td>
</tr>
<tr>
<td>FixedPage.NavigateUri</td>
<td>&lt;Canvas&gt; &lt;Glyphs&gt; &lt;Path&gt;</td>
<td>Defines a hyperlink source. For more information, see §16.2.</td>
</tr>
</tbody>
</table>
### 14. Common Properties

<table>
<thead>
<tr>
<th>xml:lang</th>
<th><code>&lt;Canvas&gt;</code></th>
<th>Specifies a language.</th>
</tr>
</thead>
<tbody>
<tr>
<td>---</td>
<td><code>&lt;FixedPage&gt;</code></td>
<td>---</td>
</tr>
<tr>
<td>---</td>
<td><code>&lt;Glyphs&gt;</code></td>
<td>---</td>
</tr>
<tr>
<td>---</td>
<td><code>&lt;Path&gt;</code></td>
<td>---</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;Canvas.Resources&gt;</code></td>
<td>Contains elements that can be reused by reference throughout the markup of the <code>&lt;FixedPage&gt;</code> or <code>&lt;Canvas&gt;</code> child or descendant elements.</td>
</tr>
<tr>
<td><code>&lt;FixedPage.Resources&gt;</code></td>
<td>---</td>
</tr>
<tr>
<td><code>&lt;Canvas.Clip&gt;</code></td>
<td>Restricts the region to which a brush can be applied.</td>
</tr>
<tr>
<td><code>&lt;Glyphs.Clip&gt;</code></td>
<td>---</td>
</tr>
<tr>
<td><code>&lt;Path.Clip&gt;</code></td>
<td>---</td>
</tr>
<tr>
<td><code>&lt;Canvas.RenderTransform&gt;</code></td>
<td>Establishes a new coordinate space through the use of an affine matrix transformation. For more information, see §14.4.</td>
</tr>
<tr>
<td><code>&lt;Glyphs.RenderTransform&gt;</code></td>
<td>---</td>
</tr>
<tr>
<td><code>&lt;Path.RenderTransform&gt;</code></td>
<td>---</td>
</tr>
<tr>
<td><code>&lt;ImageBrush.Transform&gt;</code></td>
<td>Establishes a new effective coordinate space through the use of an affine matrix transformation.</td>
</tr>
<tr>
<td><code>&lt;LinearGradientBrush.Transform&gt;</code></td>
<td>---</td>
</tr>
<tr>
<td><code>&lt;PathGeometry.Transform&gt;</code></td>
<td>---</td>
</tr>
<tr>
<td><code>&lt;RadialGradientBrush.Transform&gt;</code></td>
<td>Path geometry transformations ( <code>&lt;PathGeometry.Transform&gt;</code> ) are applied before brushes. The results are concatenated with any containing effective render transformation.</td>
</tr>
<tr>
<td><code>&lt;VisualBrush.Transform&gt;</code></td>
<td>---</td>
</tr>
<tr>
<td><code>&lt;Canvas.OpacityMask&gt;</code></td>
<td>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.</td>
</tr>
<tr>
<td><code>&lt;Glyphs.OpacityMask&gt;</code></td>
<td>---</td>
</tr>
<tr>
<td><code>&lt;Path.OpacityMask&gt;</code></td>
<td>---</td>
</tr>
</tbody>
</table>

#### 14.1 Opacity

The Opacity property attribute is used to transparently blend the current element with previously specified elements, also known as alpha blending. The opacity value MUST fall within the 0 (fully transparent) to 1 (fully opaque) range, inclusive [M2.72].

For more information, see §18.4.

#### 14.2 Resources and Resource References

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

Each resource in a resource dictionary has a key. Any property that specifies its value by referencing a resource key in a resource dictionary is called a resource reference.
The `<Canvas>` and `<FixedPage>` elements can carry a resource dictionary. A resource
dictionary is expressed in markup by the `<FixedPage.Resources>` or `<Canvas.Resources>`
property element. Individual resource values MUST be specified within a resource dictionary
[M7.1].

The `<Canvas.Resources>` or `<FixedPage.Resources>` property elements MUST precede any
property elements of the `<Canvas>` or `<FixedPage>` elements [M2.72]. Likewise, they MUST
precede any path, glyphs, or canvas children of the `<Canvas>` or `<FixedPage>` elements
[M2.72].

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

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

### 14.2.1 `<FixedPage.Resources>` Element

**Diagram:**

- `<FixedPage.Resources>`
- `<ResourceDictionary>`
  - `<PathGeometry x:Key="Rectangle">`
    - `<PathFigure StartPoint="20,20" IsClosed="true">`
      - `<PolyLineSegment Points="120,20 120,70 20,70" />`
    - `</PathFigure>`
  - `</PathGeometry>`
- `</ResourceDictionary>`

**Annotation:**

Contains the resource dictionary for the `<FixedPage>` element.

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

```
<FixedPage Width="816" Height="1056" xml:lang="en-US"
  xmlns="http://schemas.microsoft.com/xps/2005/06"
  xmlns:x="http://schemas.microsoft.com/xps/2005/06/resourcedictionary-
  key">
  <FixedPage.Resources>
    <ResourceDictionary>
      <PathGeometry x:Key="Rectangle">
        <PathFigure StartPoint="20,20" IsClosed="true">
          <PolyLineSegment Points="120,20 120,70 20,70" />
        </PathFigure>
      </PathGeometry>
    </ResourceDictionary>
  </FixedPage.Resources>
  <Path Stroke="#000000"
    StrokeThickness="1"
    Data="{StaticResource Rectangle}" />
</FixedPage>
```
This markup is rendered as follows:

```

end example]
```
### 14.2.2 `<Canvas.Resources>` Element

#### Diagram

The diagram shows the relationship between the `<Canvas.Resources>` element and the `ResourceDictionary` element.

#### Annotation

Contains the resource dictionary for the `<Canvas>` element.

#### Example 14–2. `<Canvas.Resources>` usage

```xml
<Canvas
  xmlns:x="http://schemas.microsoft.com/xps/2005/06/resourcedictionary-key">
  <Canvas.Resources>
    <ResourceDictionary>
      <PathGeometry x:Key="Circle">
        <PathFigure StartPoint="20,70">
          <ArcSegment
            Point="120,70"
            Size="50,50"
            RotationAngle="0"
            IsLargeArc="true"
            SweepDirection="Clockwise"/>
          <ArcSegment
            Point="20,70"
            Size="50,50"
            RotationAngle="0"
            IsLargeArc="true"
            SweepDirection="Clockwise"/>
        </PathFigure>
      </PathGeometry>
    </ResourceDictionary>
  </Canvas.Resources>
  <Path Stroke="#000000"
        StrokeThickness="1"
        Data="{StaticResource Circle}"/>
</Canvas>
```

This markup is rendered as follows:

```xml
end example]```
14.2.3 <ResourceDictionary> Element

element ResourceDictionary

<table>
<thead>
<tr>
<th>attributes</th>
<th>Name</th>
<th>Type</th>
<th>Use</th>
<th>Default</th>
<th>Fixed</th>
<th>Annotation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source</td>
<td>xs:anyURI</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Specifies the URI of a part containing markup for a resource dictionary. The URI MUST refer to a part in the package [M2.1].</td>
</tr>
</tbody>
</table>

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

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

A resource definition MAY reference another resource defined previously in the same resource dictionary [O7.2]. 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 [O7.3].
Namespace prefixes in resource definitions MUST apply in the context of the definition, rather than in the context of the resource reference [M7.2]. An xml:lang attribute within a resource definition MUST be interpreted in the context of the resource reference, not the resource definition [M7.3].

Example 14–3. Resource dictionary markup

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

```xml
<Canvas xmlns:x="http://schemas.microsoft.com/xps/2005/06/resourcedictionary-key">
  <Canvas.Resources>
    <ResourceDictionary>
      <PathGeometry x:Key="Rectangle">
        <PathFigure StartPoint="20,20" IsClosed="true">
          <PolyLineSegment Points="120,20 120,70 20,70" />
        </PathFigure>
      </PathGeometry>
      <PathGeometry x:Key="Circle">
        <PathFigure StartPoint="20,70">
          <ArcSegment
            Point="120,70"
            Size="50,50"
            RotationAngle="0"
            IsLargeArc="true"
            SweepDirection="Clockwise" />
        </PathFigure>
      </PathGeometry>
    </ResourceDictionary>
  </Canvas.Resources>
  <Path Data="{StaticResource Rectangle}" Fill="#FF0000" />
</Canvas>
```

end example]

14.2.3.1 Remote Resource Dictionaries

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

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

A remote Resource Dictionary part MUST be added as a Required Resource relationship from the FixedPage part that references it [M2.10]. In addition, producers MUST add each resource such as fonts or images referenced in the Resource Dictionary part as a Required Resource relationship from the FixedPage part (not the Resource Dictionary part) to the indirectly required resource, even if the particular fixed page does not reference the resource [M2.10].

For more information, see §H.3.

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 [M7.7].

**Example 14–4. A remote resource dictionary and reference**

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

```xml
  <PathGeometry x:Key="Rectangle">
    <PathFigure StartPoint="20,20" IsClosed="true">
      <PolyLineSegment Points="120,20 120,70 20,70" />
    </PathFigure>
  </PathGeometry>
  <PathGeometry x:Key="Circle">
    <PathFigure StartPoint="20,70">
      <ArcSegment
        Point="120,70"
        Size="50,50"
        RotationAngle="0"
        IsLargeArc="true"
        SweepDirection="Clockwise" />
      <ArcSegment
        Point="20,70"
        Size="50,50"
        RotationAngle="0"
        IsLargeArc="true"
        SweepDirection="Clockwise" />
    </PathFigure>
  </PathGeometry>
</ResourceDictionary>
```

The following markup references the previously defined resource dictionary:

```xml
<Canvas xmlns:x="http://schemas.microsoft.com/xps/2005/06/resourcedictionary-key">
  <Canvas.Resources>
    <ResourceDictionary Source="/resource.xaml"/>
  </Canvas.Resources>
</Canvas>
```
<Path Data="{StaticResource Rectangle}">
  <Path.Fill>
    <SolidColorBrush Color="#FF0000" />
  </Path.Fill>
</Path>
</Canvas>

end example]
14. Common Properties

14.2.4 Resource References

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

```xml
{StaticResource key}
```

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

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

It is considered an error if a static resource reference cannot be resolved, or if it can be resolved but the resource type does not match the usage at the location of reference.

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

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

```xml
<Canvas
  xmlns:x="http://schemas.microsoft.com/xps/2005/06/resourcedictionary-key">
  <Canvas.Resources>
    <ResourceDictionary>
      <PathGeometry x:Key="Rectangle">
        <PathFigure StartPoint="20,20" IsClosed="true">
          <PolyLineSegment Points="120,20 120,70 20,70" />
        </PathFigure>
      </PathGeometry>
    </ResourceDictionary>
  </Canvas.Resources>
  <Path Data="{StaticResource Rectangle}"
        Fill="<SolidColorBrush Color="#FF0000"/>">
    </Path>
  </Canvas>
```

end example]

14.2.5 Scoping Rules for Resolving Resource References

The value of the `x:Key` attribute MUST be unique within the resource dictionary [M2.72].

However, the resource dictionary of a `<Canvas>` element MAY re-use an `x:Key` value defined in the resource dictionary of a parent or ancestor `<Canvas>` or `<FixedPage>` element [O7.5]. Resource references are resolved from the innermost to the outermost resource dictionary.

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

A resource definition MAY reference another resource defined prior to the point of reference, including a resource previously defined within the same resource dictionary [O7.2]. 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 [M7.9].

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

Example 14–6. Using scoping rules

```xml
<FixedPage
  xmlns="http://schemas.microsoft.com/xps/2005/06"
  xmlns:x="http://schemas.microsoft.com/xps/2005/06/resourcedictionary-key"
  Height="1056" Width="816" xml:lang="en-US">
  <FixedPage.Resources>
    <ResourceDictionary>
      <SolidColorBrush x:Key="FavoriteColorFill" Color="#808080" />
    </ResourceDictionary>
  </FixedPage.Resources>
  <Canvas>
    <Canvas.Resources>
      <ResourceDictionary>
        <SolidColorBrush x:Key="FavoriteColorFill"
                         Color="#000000" />
      </ResourceDictionary>
    </Canvas.Resources>
    <Canvas>
      <Path Fill="{StaticResource FavoriteColorFill}">
        <Path.Data>
          ...
        </Path.Data>
      </Path>
    </Canvas>
    <Canvas>
      <Path Fill="{StaticResource FavoriteColorFill}">
        <Path.Data>
          ...
        </Path.Data>
      </Path>
    </Canvas>
    <Canvas>
      <Path Fill="{StaticResource FavoriteColorFill}">
        <Path.Data>
          ...
        </Path.Data>
      </Path>
    </Canvas>
  </Canvas>
</FixedPage>
```

end example]
14. Common Properties

14.2.6 Support for Markup Compatibility

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 [M2.10].
14.3 Clipping

The Clip property specifies a geometric area that restricts the rendered region of an element area to be filled with a brush.

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

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

14.3.1 <Canvas.Clip> Element

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

Example 14–7. Canvas clip markup and rendering

```xml
<Canvas>
  <Canvas.Clip>
    <PathGeometry>
      <PathFigure StartPoint="25,25" IsClosed="true">
        <PolylineSegment Points="60,25 70,60 80,25 115,25 115,115 80,115 70,80 60,115 25,115" />
      </PathFigure>
    </PathGeometry>
  </Canvas.Clip>
  <Path Fill="#9999CC">
    <Path.Data>
      <PathGeometry>
        <PathFigure StartPoint="20,70">
          <ArcSegment
            Point="120,70"
            Size="50,50"
            RotationAngle="0"
            IsLargeArc="true"
            SweepDirection="Clockwise" />
        </PathFigure>
      </PathGeometry>
    </Path.Data>
  </Path>
</Canvas>
```
14. Common Properties

This markup is rendered as follows:

```xml
<Canvas>
  <Path Fill="#9999CC">
    <Path.Clip>
      <PathGeometry>
        <PathFigure StartPoint="25,25" IsClosed="true">
          <PolyLineSegment Points="115,25 115,115 25,115" />
        </PathFigure>
        <PathFigure StartPoint="55,55" IsClosed="true">
          <PolyLineSegment Points="85,55 85,85 55,85" />
        </PathFigure>
      </PathGeometry>
    </Path.Clip>
    <Path.Data>
      <PathGeometry>
        <PathFigure StartPoint="20,70">
          <ArcSegment
            Point="120,70"
            Size="50,50"
            RotationAngle="0"
            IsLargeArc="true"
            SweepDirection="Clockwise" />
        </PathFigure>
      </PathGeometry>
    </Path.Data>
  </Path>
</Canvas>
```

end example]

### 14.3.2 `<Path.Clip>` Element

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

**Example 14–8. `<Path.Clip>` usage**

The following markup describes a complex clipping behavior:

```xml
<Canvas>
  <Path Fill="#9999CC">
    <Path.Clip>
      <PathGeometry>
        <PathFigure StartPoint="25,25" IsClosed="true">
          <PolyLineSegment Points="115,25 115,115 25,115" />
        </PathFigure>
        <PathFigure StartPoint="55,55" IsClosed="true">
          <PolyLineSegment Points="85,55 85,85 55,85" />
        </PathFigure>
      </PathGeometry>
    </Path.Clip>
    <Path.Data>
      <PathGeometry>
        <PathFigure StartPoint="20,70">
          <ArcSegment
            Point="120,70"
            Size="50,50"
            RotationAngle="0"
            IsLargeArc="true"
            SweepDirection="Clockwise" />
        </PathFigure>
      </PathGeometry>
    </Path.Data>
  </Path>
</Canvas>
```
This markup is rendered as follows:

```
<Path Data="M 0,0 L 180,0 L 180,140 L 0,140 Z M 20,60 L 140,60 L 140,80 L 20,80 Z"
  OriginX="20"
  OriginY="130"
  UnicodeString="N"
  FontRenderingEmSize="170"
  FontUri="../Resources/Fonts/Timesbd.ttf" />
```


The following markup uses abbreviated geometry syntax to define the clipping region:
14.4 Positioning Content

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

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

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

The RenderTransform and Transform properties both specify an affine matrix transformation to the local coordinate space, using the <MatrixTransform> element as their value. An abbreviated matrix transformation syntax MAY be used to specify a RenderTransform or Transform attribute value [M2.72].

### 14.4.1 <MatrixTransform> Element

<table>
<thead>
<tr>
<th>attributes</th>
<th>Name</th>
<th>Type</th>
<th>Use</th>
<th>Default</th>
<th>Fixed</th>
<th>Annotation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Matrix</td>
<td>ST_Matrix</td>
<td>required</td>
<td></td>
<td></td>
<td>Specifies the matrix structure that defines the transformation.</td>
</tr>
<tr>
<td></td>
<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...</td>
</tr>
</tbody>
</table>
The <MatrixTransform> element defines an arbitrary affine matrix transformation used to manipulate the coordinate systems of elements. A 3x3 matrix is used for transformations in an x,y plane. Affine transformation matrices can be multiplied to form any number of linear transformations, such as rotation and skew (shear), followed by translation. An affine transformation matrix has its final column equal to 0,0,1, so only the members in the first two columns are specified.

\[
\begin{bmatrix}
M_{11} & M_{12} & 0 \\
M_{21} & M_{22} & 0 \\
\text{OffsetX} & \text{OffsetY} & 1
\end{bmatrix}
\]

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

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

\[M11,M12,M21,M22,\text{OffsetX},\text{OffsetY}\]

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

**Example 14–10. Matrix scaling**

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

**Example 14–11. Matrix reversing the x axis**

\[
\begin{bmatrix}
-1 & 0 & 0
\end{bmatrix}
\]
Example 14–12. Matrix reversing the y axis
\[
\begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & 0 \\
-1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}
\]
end example]

Example 14–13. Matrix skewing
\[
\begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & -1 & 0 \\
0 & 0 & 1
\end{bmatrix}
\]
end example]

Example 14–14. Matrix rotating
\[
\begin{bmatrix}
\cos \theta & \sin \theta & 0 \\
-\sin \theta & \cos \theta & 0 \\
0 & 0 & 1
\end{bmatrix}
\]
end example]
Example 14–15. Matrix positioning

\[
\begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
OffsetX & OffsetY & 1
\end{bmatrix}
\]

Example 14–16. <MatrixTransform> usage

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

```xml
<Path
    Stroke="#000000"
    Fill="#9999BB"
    Data="M 0,0 L 60,0 L 60,25 L 60,0 L 120,0 L 120,120 L 0,120 Z">
    <Path.RenderTransform>
        <MatrixTransform Matrix="0,1,-1,0,170,50" />
    </Path.RenderTransform>
</Path>
```

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

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

```
end example]
```
Example 14–17. Using abbreviated matrix transformation syntax

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

```xml
<Path
  Stroke="#000000"
  Fill="#9999BB"
  Data="M 0,0 L 60,0 L 60,25 L 60,0 L 120,0 L 120,120 L 0,120 Z"
  RenderTransform="0,1,-1,0,170,50" />
```

end example]

14.4.2 `<Canvas.RenderTransform>` Element

element `Canvas.RenderTransform`

```xml
<Canvas.RenderTransform>
  <MatrixTransform Matrix="1,0,0,1,25,50"/>
</Canvas.RenderTransform>
```

The following markup positions child elements of the canvas by the render transformation:

```
<Canvas Resources>
  <ResourceDictionary>
    <PathGeometry x:Key="StarFish">
      <PathFigure StartPoint="50,0" IsClosed="true">
        <PolyLineSegment Points="55,45 100,25 55,50 80,100 50,55
                         20,100 45,50 0,25 45,45"/>
      </PathFigure>
    </PathGeometry>
  </ResourceDictionary>
  <Canvas.RenderTransform>
    <MatrixTransform Matrix="1,0,0,1,25,50"/>
  </Canvas.RenderTransform>
  <Path Data="{StaticResource StarFish}" Fill="#00FF00"/>
</Canvas>
```

```
<!-- Draw a green starfish shifted 25 to the right and 50 down -->
<Canvas Resources>
  <ResourceDictionary>
    <PathGeometry x:Key="StarFish">
      <PathFigure StartPoint="50,0" IsClosed="true">
        <PolyLineSegment Points="55,45 100,25 55,50 80,100 50,55
                         20,100 45,50 0,25 45,45"/>
      </PathFigure>
    </PathGeometry>
  </ResourceDictionary>
  <Canvas.RenderTransform>
    <MatrixTransform Matrix="1,0,0,1,25,50"/>
  </Canvas.RenderTransform>
  <Path Data="{StaticResource StarFish}" Fill="#FF0000"/>
</Canvas>
```

```
<!-- Draw a red starfish shifted 100 to the right and 150 down -->
<Canvas>
  <Canvas.RenderTransform>
    <MatrixTransform Matrix="1,0,0,1,100,150"/>
  </Canvas.RenderTransform>
  <Path Data="{StaticResource StarFish}" Fill="#00FF00"/>
</Canvas>
```
14. Common Properties

This markup is rendered as follows:

```
<Canvas.RenderTransform>
  <MatrixTransform Matrix="1,0,0,1,100,150" />
</Canvas.RenderTransform>
<Path Data="{StaticResource StarFish}">
  <Path.Fill>
    <SolidColorBrush Color="#FF0000" />
  </Path.Fill>
</Path>
</Canvas>
```

Example 14–19. `<Path.RenderTransform>` usage

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

```
<Path
  Fill="#999999"
  Stroke="#000000"
  Data="M 20,70 A 50,50 0 1 1 120,70 L 100,70 L 120,70 A 50,50 0 1 1 20,70 Z" />
```
14.4.4 `<Glyphs.RenderTransform>` Element

**element** `Glyphs.RenderTransform`

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.

**Example 14–20. `<Glyphs.RenderTransform>` usage**

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

```xml
<Glyphs RenderTransform="1,0,0,-1,0,150" />
<MatrixTransform Matrix="1,0,0,-1,0,150" />
<MatrixTransform Matrix="1,0,0,1,0,0" />
<Path.RenderTransform>
    <MatrixTransform Matrix="1,0.5,0,1,0,0" />
</Path.RenderTransform>
</Path>
```

This markup is rendered as follows:

![Glyphs RenderTransform Example](image)

end example]
This markup is rendered as follows:

![Diagram](image)

*end example]*

### 14.4.5 `<PathGeometry.Transform>` Element

#### element `PathGeometry.Transform`

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.

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

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

```xml
<Path StrokeThickness="2" Stroke="#000000">
  <Path.Data>
    <PathGeometry Transform="1.5,0,0,1.5,20,20">
      <PathGeometry>
        <PathGeometry.Transform>
          <MatrixTransform Matrix="1.5,0,0,1.5,20,20" />
        </PathGeometry.Transform>
      </PathGeometry>
      <PathFigure StartPoint="50,0" IsClosed="true">
        <PolyLineSegment Points="150,0 200,100 0,100" />
      </PathFigure>
    </PathGeometry>
  </Path.Data>
</Path>
```
This markup is rendered as follows. The pre-transform path is indicated in light gray. Note that the stroke thickness did not change. If this transformation had been applied to the entire Path, the stroke thickness would also have increased by 150%.

end example]
14. Common Properties

14.4.6 `<ImageBrush.Transform>` Element

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

Example 14–22. `<ImageBrush.Transform>` usage

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

```xml
<Path
    StrokeThickness="5"
    Stroke="#996666"
    StrokeLineJoin="Round"
    <Path.Fill>
        <ImageBrush
            ImageSource="dog.jpg"
            Transform=".939,.342,-.342,.939,0,-80"
            TileMode="Tile"
            Viewbox="0,0,270,423"
            ViewboxUnits="Absolute"
            Viewport="75,75,90,125"
            ViewportUnits="Absolute" />
        <ImageBrush.Transform>
            <MatrixTransform Matrix=".939,.342,-.342,.939,0,-80" />
        </ImageBrush.Transform>
    </ImageBrush>
</Path.Fill>
</Path>
```
This markup is rendered as follows:

```
<VisualBrush.Transform> element

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.

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


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

```xml
<Path
    StrokeThickness="5"
    Stroke="#336666"
    StrokeLineJoin="Round"
    Data="M 25,25 L 365,25 L 365,250 L 25,250 Z M 70,70 L 320,70 L 320,205 L 70,205 Z">
    <Path.Fill>
        <VisualBrush
            Transform="0.707,0.707,-0.707,0.707,0,0"/>
    </Path.Fill>
</Path>
```
14. Common Properties

The markup is rendered as follows:

```xml
<Path Fill="#99CCCC" Data="M 0,0 L 60,0 L 60,100 L 0,100 Z" />
<Path Stroke="#336666" Data="M 0,0 L 0,100 M 20,0 L 20,100 M 40,0 L 40,100 M 60,0 L 60,100 M 80,0 L 80,100" />
```

end example]
Example 14–24. `<VisualBrush.Transform>` usage with tiling behavior

This example demonstrates tile rendering behavior when applying a transform.

```xml
<!-- Draw background diamond to show where fill affects background -->
<Path Fill="#CCCC66" Data="M 200,0 L 400,200 L 200,400 L 0,200 Z" />
<Path Data="M 200,0 L 400,200 L 200,400 L 0,200 Z"/>
<Path.Fill>
  <VisualBrush
    Viewbox="0,0,1,1"
    Viewport="200,133,67,67"
    ViewboxUnits="Absolute"
    ViewportUnits="Absolute"
    Transform="1,0,1,1,0,0"
    TileMode="FlipY">
    <MatrixTransform Matrix="1,0,1,1,0,0"/>
  </VisualBrush.Transform>
  <VisualBrush.Visual>
    <Canvas>
      <Path Fill="#333399" Data="M 0.1,0.1 L 0.9,0.1 L 0.9,0.9 L 0.1,0.9 Z"/>
      <Path Fill="#FFFF00" Data="M 0.1,0.35 L 0.35,0.1 L 0.6,0.35 L 0.35,0.6 Z"/>
    </Canvas>
  </VisualBrush.Visual>
</Path.Fill>
</Path>
```

This markup is rendered as follows:
14.4.8 `<LinearGradientBrush.Transform>` Element

**Element** `LinearGradientBrush.Transform`

- **Diagram**
  - `CT_CP_Transform` (concatenated Transform property)
  - `MatrixTransform` (current effective render 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 start point and end point are transformed using the local effective render transform.

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

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

```xml
<Path Stroke="#000000" StrokeThickness="2" Data="M 20,50 L 170,50 L 170,200 L 20,200 Z M 120,20 L 270,20 L 270,170 120,170 Z">
  <Path.Fill>
    <LinearGradientBrush MappingMode="Absolute" Transform=".707,.707,-.707,.707,150,-30">
      <!-- Gradient details -->
    </LinearGradientBrush>
  </Path.Fill>
</Path>
```

*end example*
This markup is rendered as follows:

```
<LinearGradientBrush.Transform>
  <MatrixTransform Matrix="-0.707,0.707,0.707,-0.707,150,-30" />  
</LinearGradientBrush.Transform>
<LinearGradientBrush.GradientStops>
  <GradientStop Color="#9999FF" Offset="0.0"/>
  <GradientStop Color="#333366" Offset="1.0"/>
</LinearGradientBrush.GradientStops>
</LinearGradientBrush>
</Path.Fill>
</Path>
```

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

```
<LinearGradientBrush.Transform>
  <MatrixTransform Matrix="-0.707,0.707,0.707,-0.707,150,-30" />  
</LinearGradientBrush.Transform>
<LinearGradientBrush.GradientStops>
  <GradientStop Color="#9999FF" Offset="0.0"/>
  <GradientStop Color="#333366" Offset="1.0"/>
</LinearGradientBrush.GradientStops>
</LinearGradientBrush>
```

14.4.9 `<RadialGradientBrush.Transform>` Element

Element `<RadialGradientBrush.Transform>`

Diagram:

```
RadialGradientBrush.Transform  MatrixTransform
```

end example]
14. Common Properties

Example 14-26. `<RadialGradientBrush.Transform>` usage

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

```xml
<Path
    Stroke="#000000"
    StrokeThickness="2"
    Data="M 20,20 L 270,20 L 270,200 L 20,200 Z">
    <Path.Fill>
        <RadialGradientBrush MappinMode="Absolute" Transform=".707,.707,-.707,.707,150,-10"
            Center="80,90"
            RadiusX="50"
            RadiusY="80"
            GradientOrigin="70,15"
            SpreadMethod="Reflect">
            <RadialGradientBrush.Transform>
                <MatrixTransform Matrix=".707,.707,-.707,.707,150,-10" />
            </RadialGradientBrush.Transform>
            <RadialGradientBrush.GradientStops>
                <GradientStop Color="#9999FF" Offset="0.0" />
                <GradientStop Color="#333366" Offset="1.0" />
            </RadialGradientBrush.GradientStops>
        </RadialGradientBrush>
    </Path.Fill>
</Path>
```

This markup is rendered as follows:

Without the Transform property, this markup is rendered as follows:
end example]
14.5 OpacityMask

The OpacityMask property defines a variable alpha mask for the parent element. The alpha for areas not marked by the brush is 0.0.

14.5.1 <Canvas.OpacityMask> Element

The diagram illustrates the various brush types, including LinearGradientBrush, RadialGradientBrush, SolidColorBrush, and VisualBrush, which are used with the OpacityMask property.

Example 14–27. <Canvas.OpacityMask> usage

In the following markup, the contents of the canvas are opaque with respect to each other, but both elements are blended with the background triangle:

```xml
<Path Fill="#CCCC66" Data="M 10,10 L 300,80 L 180,240 Z" />
<Canvas>
  <Canvas.OpacityMask>
    <LinearGradientBrush MappingMode="Absolute"
                         StartPoint="0,150"
                         EndPoint="0,175"
                         SpreadMethod="Pad">
      <LinearGradientBrush.GradientStops>
        <GradientStop Color="#40000000" Offset="0.0" />
        <GradientStop Color="#FF000000" Offset="1.0" />
      </LinearGradientBrush.GradientStops>
    </LinearGradientBrush>
  </Canvas.OpacityMask>
  <Path Stroke="#000000"
       StrokeThickness="2"
       Fill="#333399"
       Data="M 20,40 L 270,40 L 270,200 L 20,200 Z" />
  <Glyphs OriginX="30"
           OriginY="180"
           UnicodeString="EXAMPLE"
           FontUri="../Resources/Fonts/Timesbd.ttf"
```
This markup is rendered as follows:

```
<Canvas>
  <Path Stroke="#000000" StrokeThickness="2" Fill="#FFFFFF" />
</Canvas>
```

end example]

14.5.2 `<Path.OpacityMask>` Element

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

```
<Path
  Stroke="#000000"
  StrokeThickness="2"
  Fill="#CCCC66"
  Data="M 135,10 L 270,250 L 20,250 Z" />
```

14. Common Properties

This markup is rendered as follows:

```
<Path.StrokeThickness="2"
Data="M 20,40 L 270,40 L 270,200 L 20,200 Z">
  <Path.OpacityMask>
    <LinearGradientBrush
      MappingMode="Absolute"
      StartPoint="0,60"
      EndPoint="0,180"
      SpreadMethod="Pad">
      <LinearGradientBrush.GradientStops>
        <GradientStop Color="#FF000000" Offset="0.0" />
        <GradientStop Color="#60000000" Offset="1.0" />
      </LinearGradientBrush.GradientStops>
    </LinearGradientBrush>
  </Path.OpacityMask>
  <Path.Fill>
    <SolidColorBrush Color="#9999FF" />
  </Path.Fill>
</Path>
```

end example]

14.5.3 `<Glyphs.OpacityMask>` Element

element `<Glyphs.OpacityMask>`
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.

The following markup demonstrates the use of an opacity mask to create a tile effect:

```xml
<Path Fill="#CCCC66" Data="M 40,40 L 480,40 L 260,120 Z" />
<Glyphs
  OriginX="20"
  OriginY="95"
  UnicodeString="EXAMPLE"
  FontUri="../Resources/Fonts/Timesbd.ttf"
  FontRenderingEmSize="100"
  Fill="#000080">
  <Glyphs.OpacityMask>
    <VisualBrush
      Viewbox="0,0,2,2"
      ViewboxUnits="Absolute"
      Viewport="0,0,6,6"
      ViewportUnits="Absolute"
      TileMode="Tile">
      <VisualBrush.Visual>
        <Path
          Fill="#CC000000"
          Data="M 0,0 L 1.5,0 L 1.5,1.5 L 0,1.5 Z" />
      </VisualBrush.Visual>
    </VisualBrush>
  </Glyphs.OpacityMask>
</Glyphs>
```

This markup is rendered as follows:

![Example Image]

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

- 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]
- ICC Version 2 profiles for 3-, 4-, 5-, 6-, 7-, and 8-channel color [M8.9]
- ICC Version 2 profiles with a Windows Color System (WCS) profile embedded as a private tag [M8.10]

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, but rather MAY convert data specified in a rich color space to sRGB at an early stage [O8.1]. Consumers that do not handle rich colors natively can experience reduced fidelity.

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. See §G for the scRGB gamut definition.
15. Color

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. 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 for images and spot colors through the use of color management transformations from an ICC profile.

15.1.7 Device Color Spaces

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 §15.5, "PrintTicket Color Settings", Table 15–4, to indicate to the device 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 [M8.11].

It is the responsibility of the consumer to identify the ICC profile as one that correctly approximates the native colors of the device. When colors in an XPS file are encoded in the native colors of the device, then the colors MUST NOT be color-managed according to the included profile unless forced to do so for transparency effects or gradient blending. [M8.11].

15.1.8 ICC Profiles

XPS Documents MAY include ICC profile parts [O2.3]. 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 [M8.12], for color spaces other than sRGB and scRGB. Optionally, XPS producers and consumers MAY provide color management using ICC profiles conforming to the requirements of ISO 15076-1, "Image technology colour management — Architecture, profile format, and data structure — Part 1: Based on ICC.1:2004-10" [O8.9].

Producers SHOULD restrict 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 [S8.15]. ICC profiles MUST conform to the requirements in the ICC Color Profile specification, Version 3.4 [M8.12].

Consumers MUST support color profiles as specified in the ICC specification [M8.9]. Supported profiles include Monochrome Input Profiles, Monochrome Display Profiles, Monochrome Output Profiles, Three-component Matrix-based Input Profiles, and RGB Display Profiles. However, the set of usable N-component LUT-based profiles is limited to 3-, 4-, 5-, 6-, 7-, or 8-color channels.
All ICC profiles used in XPS Documents MUST be one of the following [M8.13]:

- Input
- Output
- Monitor (RGB)
- ColorSpace Conversion

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

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

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

An ICC profile MAY contain the private tag, "MS00", which specifies an embedded Windows Color System (WCS) profile [O8.2].

15.1.9 WcsProfilesTag

The WcsProfilesTag (Windows color space profiles tag) is a private Microsoft ICC profile tag that is used in ICC profiles created by WcsCreateIccProfile to contain input WCS profiles. This tag conforms to ICC profile requirements for profile tags. In particular, the tag header is in big-endian byte ordering, but the embedded WCS XML profiles remain in their native byte order. Furthermore, the tag data must be aligned on a 4-byte boundary (measured from the start of the ICC profile). The structure of the tag is defined by the WcsProfilesTagType below.

The WcsProfilesTag signature is "MS00". This is the tag signature that will appear in the ICC profiles tag table for the WcsProfilesTag.

<table>
<thead>
<tr>
<th>Byte offset</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>0–3</td>
<td>MS10-type signature.</td>
</tr>
<tr>
<td>4–7</td>
<td>Reserved, must be set to 0 (according to ICC convention).</td>
</tr>
<tr>
<td>8–11</td>
<td>Byte offset from the beginning of the tag to the CDMP data.</td>
</tr>
<tr>
<td>12–15</td>
<td>Size of the CDMP data in bytes.</td>
</tr>
<tr>
<td>16–19</td>
<td>Byte offset from the beginning of the tag to the CAMP data.</td>
</tr>
<tr>
<td>20–23</td>
<td>Size of the CAMP data in bytes.</td>
</tr>
<tr>
<td>24–27</td>
<td>Byte offset from the beginning of the tag to the GMMP data.</td>
</tr>
<tr>
<td>28–31</td>
<td>Size of the GMMP data in bytes.</td>
</tr>
<tr>
<td>32–n</td>
<td>A sequence of bytes of length element_size-32, where element_size is the tag size recorded in the ICC profile tag table entry for this tag. These are the WCS XML profiles that were used to create this ICC profile via WcsCreateIccProfileFromWcsProfileHandle. The WCS</td>
</tr>
</tbody>
</table>
profiles are ordered as follows: the CDMP (required) first, followed by the CAMP (if present), followed by the GMMP (if present).

15.1.10 WCS Color Profiles

XPS Documents include only WCS color profiles embedded in ICC color profiles, as described above.

15.1.11 Vector Color Syntax

Vector colors can be specified in XPS Document markup in the following locations:

- The Color attribute of the <SolidColorBrush> element
- The Color attribute of the <GradientStop> element
- The Fill attribute of the <Path> element
- The Fill attribute of the <Glyphs> element
- The Stroke attribute of the <Path> element

The last three locations are an abbreviated syntax for expressing a solid color brush with the specified color.
## Table 15–2. Syntax summary

<table>
<thead>
<tr>
<th>Color type</th>
<th>Syntax</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>sRGB w/o alpha</td>
<td>Color=&quot;#RRGGBB&quot;</td>
<td>Color=&quot;#FFFFFF&quot;</td>
</tr>
<tr>
<td>sRGB with alpha</td>
<td>Color=&quot;#AARRGGBB&quot;</td>
<td>Color=&quot;#80FFFFFF&quot;</td>
</tr>
<tr>
<td>scRGB w/o alpha</td>
<td>Color=&quot;sc#RedFloat, GreenFloat,BlueFloat&quot;</td>
<td>Color=&quot;sc#1.0,0.5,1.0&quot;</td>
</tr>
<tr>
<td>scRGB with alpha</td>
<td>Color=&quot;sc#AlphaFloat,RedFloat, GreenFloat,BlueFloat&quot;</td>
<td>Color=&quot;sc#0.3,1.0,0.5,1.0&quot;</td>
</tr>
<tr>
<td>CMYK with alpha</td>
<td>Color=&quot;ContextColor ProfileURI AlphaFloat, Chan0Float, Chan1Float, Chan2Float, Chan3Float&quot;</td>
<td>Color=&quot;ContextColor ProfileURI AlphaFloat, Chan0Float, Chan1Float, Chan2Float, Chan3Float&quot;</td>
</tr>
<tr>
<td>N-Channel with alpha</td>
<td>Color=&quot;ContextColor ProfileURI AlphaFloat, Chan0Float, ... ChanN-1Float&quot;</td>
<td>Color=&quot;ContextColor ProfileURI AlphaFloat, Chan0Float, ... ChanN-1Float&quot;</td>
</tr>
<tr>
<td>Named color with alpha</td>
<td>Color=&quot;ContextColor ProfileURI AlphaFloat, TintFloat, 0, 0&quot;</td>
<td>Color=&quot;ContextColor ProfileURI AlphaFloat, TintFloat, 0, 0&quot;</td>
</tr>
</tbody>
</table>

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

**It is the responsibility of the consumer to determine profile usability.** A consumer incapable of supporting a particular optional ICC profile tag should treat this tag as a user-defined custom tag, and therefore ignore it. If an ICC profile contains optional tags or invalid tag type signatures such that the profile is unusable by a consumer, then the default pixel formats for each vector color space MUST be treated as follows:

- **Three-component integer default for vector data is sRGB.**
- **Three-component float default for vector data is scRGB.**
- **The specific CMYK to be used as the four component data default for vector data shall be determined by the consumer.**

A consumer might choose to abort the job.

### 15.1.12 sRGB Color Syntax

The sRGB color syntax is the same as that used in HTML, with the red, green, and blue channels represented by two hexadecimal digits. XPS Documents can specify an sRGB color either with or without an alpha channel value, which is also expressed as two hexadecimal digits.

The syntax is as follows (without alpha):

```
#RRGGBB
```

or (with alpha):
15. Color

1. #AARRGGBB

2. When an sRGB color is specified without an alpha value, an alpha of "FF" is implied.

3. **15.1.13 scRGB Color Syntax**

4. The scRGB color syntax allows XPS Document producers to specify a color using the full scRGB color space, which is much larger than the sRGB color space and can represent the entire range of colors perceivable by the human eye.
This syntax is expressed either as:

\[
\text{sc#RedFloat,GreenFloat,BlueFloat}
\]

or:

\[
\text{sc#AlphaFloat,RedFloat,GreenFloat,BlueFloat}
\]

When an scRGB color is specified with three numeric values, an alpha of 1.0 is implied. When an scRGB color is specified with four numeric values, the first value is the alpha channel. 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 [M8.15].

### 15.1.14 CMYK Color Syntax

XPS Document producers specify CMYK 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 as follows:

\[
\text{ContextColor ProfileURI AlphaFloat, Chan0Float, Chan1Float, Chan2Float, Chan3Float}
\]

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]. 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 [M8.16]. Channel float values MUST also be clamped to the valid range from 0.0 to 1.0 before further processing. If the value is used as input for an ICC profile color transformation, it MUST subsequently be linearly scaled to the range from 0 to 255 or from 0 to 65535, depending on whether the profile uses 8-bit or 16-bit input tables [M8.31].

### 15.1.15 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:

\[
\text{ContextColor ProfileURI AlphaFloat, Chan0Float, ..., ChanN-1Float}
\]

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 3-, 4-, 5-, 6-, 7- or 8-channel profile. The context color MUST specify a matching number of channel float values, setting unused ones to 0.0 [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. If the value is used as input for an ICC profile color transformation, it MUST subsequently be linearly scaled to the range from 0 to 255 or from 0 to 65535, depending on whether the profile uses 8-bit or 16-bit input tables [M8.31]. To represent a 1-channel profile, create a 3-channel profile and use only the first channel. To represent a 2-channel profile, create a 3-channel profile and use only the first and second channels.
15.1.16 Named Color Syntax

A named color is an industry-defined color specification that identifies a particular color in a well-defined color schema, usually for the purpose of printing. There are currently several named color schemas. Producers specify named colors using the context color syntax, which allows specification of a named color using an ICC profile. The named color schema used is determined by the ICC profile.

A named color is expressed as a combination of an ink name stored in the ICC profile and a tint level (percentage ink dilution). The ink name for the tint is contained in the colorantTable clrt tag. This tag is not defined for ICC Version 3.4 profiles, and its presence is benignly ignored by ICM V2 implementations. Therefore, it is used in XPS Documents to specify the names of named colors.

The syntax for referencing a single named color is as follows:

```
ContextColor ProfileURI AlphaFloat,TintFloat,0,0
```

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 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 [M8.19]. To represent a single tone or duotone profile, create a 3-channel profile and specify a colorant name in the profile’s colorantTable with zero-length (name field all set to 0) for the unused channels.

AlphaFloat specifies the alpha to be applied to the named color. TintFloat specifies how diluted with respect to the color schema’s white color point the named color is, with 1.0 being the pure named color and 0.0 being fully diluted. The final two values are always set to 0 to specify a single named color.

For duotone named colors, use the first two values and set the final value to 0. [Example: The syntax for referencing a duotone named color is as follows:]

```
ContextColor ProfileURI AlphaFloat,Tint0Float,Tint1Float,0
```

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 [M8.20]. The tint float value MUST also be clamped to the valid range from 0.0 to 1.0 before further processing. If the value is used as input for an ICC profile color transformation, it MUST subsequently be linearly scaled to the range from 0 to 255 or from 0 to 65535, depending on whether the profile uses 8-bit or 16-bit input tables [M8.31].

Consumers that do not understand the named color MUST compute a color approximation through ICM-compliant color management functions using the specified profile. When a named color is used in a gradient brush or with transparency, the result produced by consumers that are not named-color aware MAY differ significantly from the result produced by consumers that are named-color aware [O8.3].

An XPS consumer that is aware of named colors looks for the clrt tag to find out if a specific ContextColor designates a named color description. 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) [S8.3].
This method can be used for 1 to 8 named colors; however, the ContextColor syntax requires a minimum of 1 Alpha value and 3 Channel values. 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 [S8.4].

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 [S8.6].

A named color profile MAY be used with images [O8.4], which is especially useful for the reproduction of corporate logo images using strictly controlled ink sets. A named profile could contain SWOP CMYK in the first 4 channels, and use the remaining 4 channels for highlight colors. This allows the use of up to 4 accent colors in images.

15.2 Rich Colors in Raster Images

This section describes specific considerations for including raster images with rich colors in XPS Documents.

15.2.1 sRGB Raster Images

XPS Documents support sRGB raster images in the following formats:

- JPEG
- PNG
- TIFF
- Windows Media Photo

The following Windows Media Photo pixel formats are supported:

- WICPixelFormat24bppRGB
- WICPixelFormat24bppBGR
- WICPixelFormat32bppBGR
- WICPixelFormat32bppBGRA
- WICPixelFormat32bppPBGRA
- WICPixelFormat48bppRGB
- WICPixelFormat64bppRGBA
- WICPixelFormat64bppPRGBA

Pixel formats WICPixelFormat32bppPBGRA and WICPixelFormat64bppPRGBA are pre-multiplied alpha formats. See §18.4.1 for details.

The following Windows Media Photo packed pixel formats are supported:

- WICPixelFormat16bppBGR555
- WICPixelFormat16bppBGR565
- WICPixelFormat32bppBGR101010

See §9.1.5 for more details.
15.2.2 scRGB Raster Images

XPS Documents support scRGB raster images only in the Windows Media Photo image format. The following pixel formats are supported:

- WICPixelFormat48bppRGBFixedPoint
- WICPixelFormat48bppRGBHalf
- WICPixelFormat96bppRGBFixedPoint
- WICPixelFormat128bppRGBFloat
- WICPixelFormat64bppRGBAFixedPoint
- WICPixelFormat64bppRGBFixedPoint
- WICPixelFormat64bppRGBAHalf
- WICPixelFormat64bppRGBOHalf
- WICPixelFormat128bppRGBOFixedPoint
- WICPixelFormat128bppRGBOFixedPoint
- WICPixelFormat128bppRGBOFloat
- WICPixelFormat128bppRGBOFloat
- WICPixelFormat32bppRGBO

Pixel format WICPixelFormat128bppRGBOFloat is a pre-multiplied alpha format. See §18.4.1 for details.

15.2.3 Gray Raster Images

XPS Documents support gray raster images in the following formats:

- JPEG
- PNG
- TIFF
- Windows Media Photo

The following Windows Media Photo pixel formats are supported:

- WICPixelFormatBlackWhite
- WICPixelFormat8bppGray
- WICPixelFormat16bppGray
- WICPixelFormat16bppGrayFixedPoint (scRGB range)
- WICPixelFormat16bppRGBOHalf (scRGB range)
- WICPixelFormat32bppRGBOFixedPoint (scRGB range)
- WICPixelFormat32bppRGBOFloat

15.2.4 CMYK Raster Images

CMYK images are stored in TIFF 6.0 or Windows Media Photo format.
15.2.4.1 TIFF CMYK Raster Images

CMYK TIFF image tags are described in §9.1.5.3.

ICC profiles can be associated with CMYK raster images by using an ICC profile embedded in the TIFF file (tag 34675) or associated using the mechanism described in §15.2.8.

15.2.4.2 Windows Media Photo CMYK Raster Images

The Windows Media Photo CMYK format is described in the Windows Media Photo specification. The following formats are supported:

- WICPixelFormat32bppCMYK
- WICPixelFormat40bppCMYKAlpha
- WICPixelFormat64bppCMYK
- WICPixelFormat80bppCMYKAlpha

15.2.4.3 JPEG CMYK Raster Images

Support for JPEG CMYK images varies by implementation and SHOULD NOT be used in XPS Documents [S2.7]. See §9.1.5.1 for more details.

15.2.5 N-channel Raster Images

N-channel images are stored in the Windows Media Photo image file format using an ICC profile. The following formats are supported:

- WICPixelFormat24bpp3Channels, WICPixelFormat48bpp3Channels
- WICPixelFormat32bpp4Channels, WICPixelFormat64bpp4Channels
- WICPixelFormat40bpp5Channels, WICPixelFormat80bpp5Channels
- WICPixelFormat48bpp6Channels, WICPixelFormat96bpp6Channels
- WICPixelFormat56bpp7Channels, WICPixelFormat112bpp7Channels
- WICPixelFormat64bpp8Channels, WICPixelFormat128bpp8Channels
- WICPixelFormat32bpp3ChannelsAlpha, WICPixelFormat48bpp3ChannelsAlpha
- WICPixelFormat40bpp4ChannelsAlpha, WICPixelFormat64bpp4ChannelsAlpha
- WICPixelFormat48bpp5ChannelsAlpha, WICPixelFormat96bpp5ChannelsAlpha
- WICPixelFormat56bpp6ChannelsAlpha, WICPixelFormat112bpp6ChannelsAlpha
- WICPixelFormat64bpp7ChannelsAlpha, WICPixelFormat128bpp7ChannelsAlpha
- WICPixelFormat72bpp8ChannelsAlpha, WICPixelFormat144bpp8ChannelsAlpha

15.2.6 Named Color Raster Images

Named color (N-tone) raster images are stored in the Windows Media Photo image file format using an ICC profile that maps the tint channel combinations to valid PCS values. See §15.2.5 for pixel format definitions.

Consumers unaware of named colors can then compute color approximations using the PCS values computed from the profile.
15.2.7 Device Color Raster Images

Device color (N-channel) raster images are stored in the Windows Media Photo image file format in the same manner as a named color raster image. See §15.1.7 for more details. RGB and CMYK raster images can also be stored in the TIFF image file format. JPEG CMYK images SHOULD NOT be used [S2.7].

15.2.8 Images and Color Profile Association

Images can use a color profile matching the channel configuration of the image using one of two methods:

- Color profile embedded in an image using the image format specific mechanism
- Color profile contained in a separate part associated with the image using the following markup:
  
  `<ImageBrush ImageSource="{ColorConvertedBitmap image.tif profile.icc}" ... />`

  An associated color profile overrides an embedded color profile and is processed instead of any embedded color profile. The profile URI MUST be added as a Required Resource relationship to the FixedPage part [M2.10].

15.2.9 Color Space Pixel Formats for Raster Images

*It is the responsibility of the consumer to determine profile usability. A consumer incapable of supporting a particular optional ICC profile tag should treat this tag as a user-defined custom tag, and therefore ignore it.* If an ICC profile is not embedded or associated with a raster image or if both the embedded and associated profiles, which are present, are not compatible with the pixel format of the image or contain optional tags or invalid tag type signatures such that neither profile is usable by a consumer, then the default pixel formats for each raster color space MUST be treated as follows [M8.30].

**Table 15–3. Color Space Pixel Format Defaults**

<table>
<thead>
<tr>
<th>Color Space</th>
<th>Pixel Formats</th>
</tr>
</thead>
<tbody>
<tr>
<td>sRGB</td>
<td>Integer RGB</td>
</tr>
<tr>
<td></td>
<td>Integer Grayscale</td>
</tr>
<tr>
<td></td>
<td>Integer 3-Channel</td>
</tr>
<tr>
<td>scRGB (wcsRGB gamut)</td>
<td>Floating Point scRGB</td>
</tr>
<tr>
<td></td>
<td>Half-Float scRGB</td>
</tr>
<tr>
<td></td>
<td>Fixed-Point scRGB</td>
</tr>
<tr>
<td></td>
<td>Floating Point Grayscale</td>
</tr>
<tr>
<td></td>
<td>Half-Float Grayscale</td>
</tr>
<tr>
<td></td>
<td>Fixed-Point Grayscale</td>
</tr>
<tr>
<td>CMYK (with SWOP profile)</td>
<td>Integer CMYK</td>
</tr>
<tr>
<td></td>
<td>Integer 4-Channel</td>
</tr>
<tr>
<td></td>
<td>Integer 5-Channel (ignore channel 5)</td>
</tr>
<tr>
<td></td>
<td>Integer 6-Channel (ignore channels 5 and 6)</td>
</tr>
<tr>
<td></td>
<td>Integer 7-Channel (ignore channels 5, 6, and 7)</td>
</tr>
<tr>
<td></td>
<td>Integer 8-Channel (ignore channels 5, 6, 7, and 8)</td>
</tr>
</tbody>
</table>
15.3 Color Separation

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].

15.4 Alpha and Gradient Blending with Rich Colors

For consumers that understand rich colors, it is necessary to understand how they can be blended to create gradient or transparency effects. 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 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 PageBlendColorSpace PrintTicket setting, the color interpolation mode of the gradient brush MUST be used instead [M8.25].

The behavior of documents using rich color features is implementation specific. Consumers MUST support sRGB [M8.1], but they MAY support rich color spaces such as scRGB or CMYK as well [O8.6]. Consumers that encounter any document using non-sRGB colors MAY process those colors using the simpler sRGB color space, resulting in deviations, especially for alpha blending [O8.6].
### 15.5 PrintTicket Color Settings

This section subclause summarizes the color-related PrintTicket settings. For more information, refer to the Print Schema.

#### Table 15–4. PrintTicket color settings

<table>
<thead>
<tr>
<th>Feature or ParameterDef</th>
<th>Option/ScoredProperty or Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PageColorManagement</td>
<td>Device (default)</td>
<td>Perform color management only in device.</td>
</tr>
<tr>
<td></td>
<td>Driver</td>
<td>Allow driver to perform color management. The driver MAY color manage elements or convert them to different color spaces [O8.7].</td>
</tr>
<tr>
<td>PageDeviceColorSpaceProfile</td>
<td>profileUri properties fileURI</td>
<td>Identifies an ICC profile contained in the XPS package. The processing of this option depends on the setting of the PageDeviceColorSpaceUsage feature. No default value is specified.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Contains an absolute part name relative to the package root.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>MatchToDefault (default)</td>
<td>If the device determines that the profile specified by the PageDeviceColorSpaceProfile URI 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].</td>
</tr>
</tbody>
</table>
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].

<table>
<thead>
<tr>
<th>OverrideDeviceDefault</th>
<th>If the profile specified by the PageDeviceColorSpaceProfile eURI 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.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>PageBlendColorSpace</th>
<th>The sRGB color space that SHOULD be used for blending [S8.9].</th>
</tr>
</thead>
<tbody>
<tr>
<td>scRGB</td>
<td>The scRGB color space that SHOULD be used for blending [S8.9].</td>
</tr>
</tbody>
</table>

<p>| ICCProfile | The Uri property of the option specifies an ICC profile defining the color space that SHOULD be used for blending [S11.13]. The Uri is an absolute part name relative to the package root. The profile MUST be an output profile (containing AToB0Tag, BToA0Tag, AToB1Tag, BToA1Tag, AToB2Tag, and BToA2Tag), otherwise it MUST be ignored [M8.28]. The rendering intent specified by PageICMRenderingIntent PrintTicket setting is used, unless the profile specifies a rendering intent of its own. |</p>
<table>
<thead>
<tr>
<th>Element/Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elements using the profile specified by</td>
<td>PageBlendColorSpace MAY be blended naively (channel-by-channel) without converting through PCS [O8.8].</td>
</tr>
<tr>
<td>Working Draft 1.1, February 2008</td>
<td>-------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>PageICMRenderingIntent</strong></td>
<td>AbsoluteColorimetric</td>
</tr>
<tr>
<td>RelativeColorimetric (default)</td>
<td></td>
</tr>
<tr>
<td>Photographs</td>
<td>The rendering intent as defined by the ICC Version 2 specification. This value SHOULD be ignored for</td>
</tr>
<tr>
<td>BusinessGraphics</td>
<td>elements using a profile that specifies the rendering intent in the profile [S8.13].</td>
</tr>
<tr>
<td><strong>DocumentImpositionColor</strong> colorName</td>
<td>Elements using the named color identified by the colorName properties MUST appear on all color</td>
</tr>
<tr>
<td>properties</td>
<td>separations [M8.29]. See §15.3 and 15.1.16 for details.</td>
</tr>
<tr>
<td>Consumers that do not produce separations</td>
<td>Treat these elements like other elements using named color, without any additional required</td>
</tr>
<tr>
<td>treat these elements like other elements</td>
<td>processing steps. No default value is specified.</td>
</tr>
<tr>
<td>using named color, without any additional</td>
<td></td>
</tr>
<tr>
<td>required processing steps. No default value</td>
<td></td>
</tr>
<tr>
<td>is specified.</td>
<td></td>
</tr>
<tr>
<td><strong>PageBlackGenerationProcessing</strong> Automatic</td>
<td>Default.</td>
</tr>
<tr>
<td>Custom/TotalInkCoverageLimit</td>
<td>The maximum allowable sum of the four ink coverages anywhere in an image or element (200% to 400%).</td>
</tr>
<tr>
<td>Custom/BlackInkLimit</td>
<td>The maximum allowed K-channel value (0% to 100%).</td>
</tr>
<tr>
<td>Custom/GrayComponentReplacementLevel</td>
<td>The percentage of gray component replacement to perform (0% to 100%).</td>
</tr>
<tr>
<td>Custom/GrayComponentReplacementStart</td>
<td>The point in the highlight-to-shadow range where GCR should start (0% to 100%; 100% = darkest shadow).</td>
</tr>
<tr>
<td>Custom/</td>
<td>The extent beyond neutrals</td>
</tr>
<tr>
<td>GrayComponentReplacement Extent</td>
<td>(into chromatic colors) that GCR applies. 0% = Undercolor component replacement; 100% = Gray component replacement.</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>--------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Custom/UnderColorAdditionStart</td>
<td>The shadow level below which UCA will be applied (0% to 100%).</td>
</tr>
<tr>
<td>Custom/UnderColorAdditionLevel</td>
<td>The amount of chromatic ink (in gray component ratios) to add to areas where GCR/UCR has generated “BlackInkLimit” (or “UCAStart”, if specified) in the dark neutrals and near-neutral areas (0% to 100%).</td>
</tr>
</tbody>
</table>
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.


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
refers to. When referring to a `<Canvas>` element, producers SHOULD consider all descendant elements to be referenced in markup order [S9.3]. Consumers MAY choose to interpret these scenarios as duplicate document content [O9.3].

Children of `<VisualBrush>` elements SHOULD NOT be referenced by document structure markup [S9.30].

Because each named element in a FixedPage part that is intended as an addressable location is specified in the `<PageContent.LinkTargets>` element in the FixedDocument part, consumers MAY first attempt to locate named elements directly from the FixedDocument part [O9.4].

### 16.1.1.1 `<DocumentStructure>` Element


**Example 16-1. Document structure markup**

```
<DocumentStructure
    xmlns="http://schemas.microsoft.com/xps/2005/06/documentstructure">
  <DocumentStructure.Outline>
    ...
  </DocumentStructure.Outline>
  <Story>
    ...
  </Story>
  <Story>
    ...
  </Story>
  ...
</DocumentStructure>
```

*end example*
16.1.1.2 <DocumentStructure.Outline> Element


16.1.1.3 <DocumentOutline> Element

The `<DocumentOutline>` element lets producers specify an organizational hierarchy in the form of a list of URIs to locations in the fixed page markup or to external addresses, similar to a table of contents or a set of bookmarks. The `<DocumentOutline>` element contains only `<OutlineEntry>` elements.

The `xml:lang` attribute specifies the default language used by the Description attribute of the child `<OutlineEntry>` element.

Consumers can use the document outline to implement such features as a table of contents or a navigation pane.

16.1.1.4 <OutlineEntry> Element

The `<OutlineEntry>` element contains a structured document outline that provides a list of links into the document contents or external sites.
Each `<OutlineEntry>` element represents an index to a specific location in the document or a specific location external to the document. Consumers can use the document outline information to support interactive functionality.

Example 16–2. Document outline markup

A viewing consumer can create a navigation pane that uses the Unicode value of the Description attribute of each `<OutlineEntry>` element. The corresponding location is specified by the OutlineTarget attribute, which are specified in a manner identical to hyperlinks. The OutlineLevel attribute allows consumers to indent entries in the navigation pane.

```
  <DocumentStructure
    xmlns="http://schemas.microsoft.com/xps/2005/06/documentstructure">
    <DocumentStructure.Outline>
      <DocumentOutline>
        <OutlineEntry
          OutlineLevel="1"
          Description="1. Documents"
```

<table>
<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>

Represents an index to a specific location in the document.
A consumer might display this information as follows, with the first entry linked to Documents_1 and the second entry linked to Paragraphs_1_1.

end example]

16.1.1.5  <Story> Element

The <Story> element is the root for a single story and orders all of the story fragments containing content structure information such as sections, paragraphs, and tables. Each story has a unique name that is used to correlate the content structure for each page to that story. The <Story> element contains one or more <StoryFragmentReference> elements.
16.1.1.6 `<StoryFragmentReference>` Element

The `<StoryFragmentReference>` element identifies the page with a relationship to the StoryFragments part in which the single story fragment is defined. By identifying where in the XPS Document each story fragment appears, consumers can easily access only the pages that contain a particular story.

Each page that contains a story fragment is identified by number. This number refers to the \( n \)th page of the XPS Document referenced within the fixed document sequence and fixed document markup, starting at the fixed payload root. This value is identified in the Page attribute. The StoryFragments part containing the corresponding content structure is referenced by retrieving the part associated via relationship from the indicated page. This allows consumers to access only the pages of the document that contain the story of interest. It is also possible for a single story to return to a page containing a different fragment of the same story.

The FragmentName attribute MUST be unique within the scope of the story [M2.72].

The following markup describes a four-page document containing one story that covers the first one and one-half pages and then continues on page 4. It is interrupted by a second story that begins in the middle of page 2 and concludes on page 3.

```xml
<DocumentStructure
  xmlns="http://schemas.microsoft.com/xps/2005/06/documentstructure">
  <Story Name="Story1">
    <StoryFragmentReference Page="1"/>
    <StoryFragmentReference Page="2"/>
    <StoryFragmentReference Page="4"/>
  </Story>
  <Story Name="Story2">
    <StoryFragmentReference Page="2"/>
    <StoryFragmentReference Page="3"/>
  </Story>
</DocumentStructure>
```

end example]

Example 16–4. Story flowing back and forth across a page boundary

The following markup describes a page containing two tables, arranged side-by-side, each of which continues to the following page. In this case, the fragment is split and a fragment name is specified. FragmentA refers to the content leading up to the middle of the first (left) table and FragmentB is the continuation of this table on the following page. The flow then returns to the second (right) table on page 1 (FragmentC) before continuing with the rest of the story in FragmentD.

```xml
<DocumentStructure
  xmlns="http://schemas.microsoft.com/xps/2005/06/documentstructure">
  <Story Name="Story_1">
    <StoryFragmentReference FragmentName="FragmentA" Page="1"/>
    <StoryFragmentReference FragmentName="FragmentB" Page="2"/>
    <StoryFragmentReference FragmentName="FragmentC" Page="1"/>
    <StoryFragmentReference FragmentName="FragmentD" Page="2"/>
  </Story>
</DocumentStructure>
```

end example]

16.1.2 StoryFragments Part

The StoryFragments part contains content structure markup (describing such things as tables and paragraphs) for each story fragment that appears on the page. The content structure is expressed by tags that ultimately wrap <NamedElement> references that point to fixed page markup.

Table 16–1. StoryFragments part elements

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;StoryFragments&gt;</td>
<td>Root element.</td>
</tr>
<tr>
<td>&lt;StoryFragment&gt;</td>
<td>Contains all content structure markup elements for a single story fragment.</td>
</tr>
</tbody>
</table>
Because a single content structural element can be split across pages, the `<StoryBreak>` element is provided to identify that a given element continues to the next story fragment or continues from a previous story fragment. A `<StoryBreak>` element MUST NOT be included in a position other than the first or last child element of a `<StoryFragment>` element [M2.72].

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 [S9.4]. Likewise, if a `<StoryBreak>` element is not present at the end of the content structure markup, consumers SHOULD consider the markup a continuation to the next story fragment that must be merged to determine the cross-fragment content structure [S9.4].

Content structure is merged on an element-by-element basis, merging the last element closed in the leading story fragment with the first element opened in the trailing story fragment. This process continues until the closing tag from the leading story fragment no longer matches the opening tag from the trailing story fragment.

`<TableCellStructure>` elements require special merging, such that all `<TableCellStructure>` elements within a `<TableRowStructure>` element are merged. 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 [M9.1].
Example 16–5. Content structure spanning pages

Given the following two StoryFragments parts, consumers can construct the content structure as shown.

    <-- First StoryFragments part -->

    <StoryFragments
        xmlns="http://schemas.microsoft.com/xps/2005/06/documentstructure">
    <StoryFragment FragmentType="Header">
        <StoryBreak />
        <ParagraphStructure>
            <NamedElement NameReference="Block1" />
        </ParagraphStructure>
        <StoryBreak />
    </StoryFragment>
    <StoryFragment StoryName="Story1" FragmentType="Content">
        <StoryBreak />
        <SectionStructure>
            <TableStructure>
                <TableRowGroupStructure>
                    <TableRowStructure>
                        <TableCellStructure>
                            <ParagraphStructure>
                                <NamedElement NameReference="Block2" />
                                <NamedElement NameReference="Block3" />
                            </ParagraphStructure>
                        </TableCellStructure>
                        <TableCellStructure>
                            <ParagraphStructure>
                                <NamedElement NameReference="Block4" />
                            </ParagraphStructure>
                        </TableCellStructure>
                    </TableRowStructure>
                    <TableRowStructure>
                        <TableCellStructure>
                            <ParagraphStructure>
                                <NamedElement NameReference="Block5" />
                                <NamedElement NameReference="Block6" />
                            </ParagraphStructure>
                        </TableCellStructure>
                        <TableCellStructure>
                            <ParagraphStructure>
                                <NamedElement NameReference="Block7" />
                            </ParagraphStructure>
                        </TableCellStructure>
                    </TableRowStructure>
                </TableRowGroupStructure>
            </TableStructure>
        </SectionStructure>
    </StoryFragment>
    <StoryFragment FragmentType="Footer">
        <StoryBreak />
        <ParagraphStructure>
            <NamedElement NameReference="Block8" />
    </StoryFragments>
16. Document Structure and Interactivity  

XPS Specification and Reference Guide

```xml
<!-- Second StoryFragments part -->
<StoryFragments
  xmlns="http://schemas.microsoft.com/xps/2005/06/documentstructure">
  <StoryFragment FragmentType="Header">
    <StoryBreak />
    <ParagraphStructure>
      <NamedElement NameReference="Block9" />
    </ParagraphStructure>
  </StoryFragment>
  <StoryFragment StoryName="Story1" FragmentType="Content">
    <SectionStructure>
      <TableRowGroupStructure>
        <TableRowStructure>
          <TableCellStructure>
            <ParagraphStructure>
              <NamedElement NameReference="Block10" />
              <NamedElement NameReference="Block11" />
            </ParagraphStructure>
          </TableCellStructure>
        </TableRowStructure>
        <TableRowStructure>
          <TableCellStructure>
            <ParagraphStructure>
              <NamedElement NameReference="Block12" />
            </ParagraphStructure>
          </TableCellStructure>
        </TableRowStructure>
        <TableRowStructure>
          <TableCellStructure>
            <ParagraphStructure>
              <NamedElement NameReference="Block13" />
            </ParagraphStructure>
          </TableCellStructure>
        </TableRowStructure>
      </TableRowGroupStructure>
      <TableStructure>
        <TableCellStructure />
        <TableCellStructure>
          <ParagraphStructure>
            <NamedElement NameReference="Block14" />
          </ParagraphStructure>
        </TableCellStructure>
      </TableStructure>
    </SectionStructure>
    <StoryBreak />
  </StoryFragment>
  <StoryFragment FragmentType="Footer">
    <StoryBreak />
    <ParagraphStructure>
      <NamedElement NameReference="Block14" />
    </ParagraphStructure>
  </StoryFragment>
</StoryFragments>
```
<!-- Resulting merged content structure for Story1 -->

```xml
<SectionStructure>
  <TableStructure>
    <TableRowGroupStructure>
      <TableRowStructure>
        <TableCellStructure>
          <ParagraphStructure>
            <NamedElement NameReference="Block2" />
            <NamedElement NameReference="Block3" />
          </ParagraphStructure>
        </TableCellStructure>
      </TableRowStructure>
      <TableRowStructure>
        <TableCellStructure>
          <ParagraphStructure>
            <NamedElement NameReference="Block4" />
            <NamedElement NameReference="Block5" />
            <NamedElement NameReference="Block6" />
          </ParagraphStructure>
        </TableCellStructure>
      </TableRowStructure>
      <TableRowStructure>
        <TableCellStructure>
          <ParagraphStructure>
            <NamedElement NameReference="Block7" />
            <NamedElement NameReference="Block10" />
            <NamedElement NameReference="Block11" />
          </ParagraphStructure>
        </TableCellStructure>
      </TableRowStructure>
      <TableRowStructure>
        <TableCellStructure>
          <ParagraphStructure>
            <NamedElement NameReference="Block12" />
            <NamedElement NameReference="Block13" />
          </ParagraphStructure>
        </TableCellStructure>
      </TableRowStructure>
    </TableRowGroupStructure>
  </TableStructure>
</SectionStructure>
```

end example]

16.1.2.1 `<StoryFragments>` Element

element StoryFragments
The root of a StoryFragments part. Contains all story fragments that appear on a specific page.

The `<StoryFragments>` element groups all of the `<StoryFragment>` elements on a page.
16.1.2.2 <StoryFragment> Element

element StoryFragment

<table>
<thead>
<tr>
<th>attributes</th>
<th>Name</th>
<th>Type</th>
<th>Use</th>
<th>Default</th>
<th>Fixed</th>
<th>Annotation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>StoryName</td>
<td>xs:string</td>
<td>optional</td>
<td></td>
<td></td>
<td>Identifies the story that this story fragment belongs to. If omitted, the story fragment is not associated with any story.</td>
</tr>
<tr>
<td></td>
<td>FragmentName</td>
<td>xs:string</td>
<td>optional</td>
<td></td>
<td></td>
<td>Used to uniquely identify the story fragment.</td>
</tr>
<tr>
<td></td>
<td>FragmentType</td>
<td>ST_FragmentType</td>
<td>required</td>
<td></td>
<td></td>
<td>Specifies the type of content included in the story fragment. Valid values are Content, Header, and Footer.</td>
</tr>
</tbody>
</table>

Each <StoryFragment> has a StoryName attribute that associates it with a story defined in the DocumentStructure part. It also has a FragmentType attribute, the values for which are Content (the default), Header, or Footer.
Headers and footers are defined in their own story fragment on each page. These stories do not specify a StoryName value, so they are essentially unreferenced stories that exist only on a single page.

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 [S9.5].

Example 16–6. StoryFragments part markup

The following markup describes the StoryFragments part of a one-page document:

```xml
<StoryFragments
    xmlns="http://schemas.microsoft.com/xps/2005/06/documentstructure">
  <StoryFragment FragmentType="Header">
    <StoryBreak />
    <ParagraphStructure>
      <NamedElement NameReference="Block13" />
      <NamedElement NameReference="Block14" />
    </ParagraphStructure>
    <StoryBreak />
  </StoryFragment>
  <StoryFragment StoryName="Story1" FragmentType="Content">
    <StoryBreak />
    <ParagraphStructure>
      <NamedElement NameReference="Block1" />
      <NamedElement NameReference="Block2" />
    </ParagraphStructure>
    <TableStructure>
      <TableRowGroupStructure>
        <TableRowStructure>
          <TableCellStructure>
            <ParagraphStructure>
              <NamedElement NameReference="Block3" />
              <NamedElement NameReference="Block4" />
            </ParagraphStructure>
            <NamedElement NameReference="Block5" />
          </TableCellStructure>
          <TableCellStructure>
            <ParagraphStructure>
              <NamedElement NameReference="Block6" />
            </ParagraphStructure>
            <NamedElement NameReference="Block7" />
            <NamedElement NameReference="Block8" />
          </TableCellStructure>
        </TableRowStructure>
      </TableRowGroupStructure>
    </TableStructure>
    <SectionStructure>
      <ParagraphStructure>
        <NamedElement NameReference="Block6" />
      </ParagraphStructure>
      <ParagraphStructure>
        <NamedElement NameReference="Block7" />
      </ParagraphStructure>
      <ParagraphStructure>
        <NamedElement NameReference="Block8" />
      </ParagraphStructure>
    </SectionStructure>
    <SectionStructure>
    </SectionStructure>
  </StoryFragment>
</StoryFragments>
```
A `<StoryFragment>` element MAY be identified with a `FragmentName` attribute to distinguish it from other fragments for the same story on a single page [M2.72].

**Example 16–7. Story fragments markup using a fragment name**

```xml
<StoryFragments
  xmlns="http://schemas.microsoft.com/xps/2005/06/documentstructure">

  <StoryFragment
    StoryName="Story1"
    FragmentName="Fr1"
    FragmentType="Content">
    <StoryBreak />
    <ParagraphStructure>
      <NamedElement NameReference="Block1" />
      <NamedElement NameReference="Block2" />
    </ParagraphStructure>
    <StoryBreak />
  </StoryFragment>

  <StoryFragment
    StoryName="Story1"
    FragmentName="Fr2"
    FragmentType="Content">
    <StoryBreak />
    <ParagraphStructure>
      <NamedElement NameReference="Block1" />
      <NamedElement NameReference="Block2" />
    </ParagraphStructure>
    <StoryBreak />
  </StoryFragment>

</StoryFragments>
```
FragmentType="Content">
  <StoryBreak />
  <ParagraphStructure>
    <NamedElement NameReference="Block8" />
  </ParagraphStructure>
  <StoryBreak />
</StoryFragment>
</StoryFragments>

end example]
16.1.2.3 <StoryBreak> Element

The <StoryBreak> element signals to the consumer not to perform merging across story fragments to determine the content structure.

16.1.2.4 <SectionStructure> Element

The <SectionStructure> element provides an arbitrary grouping of <Paragraph>, <TableStructure>, <ListStructure>, and <FigureStructure> elements.

16.1.2.5 <ParagraphStructure> Element

A <ParagraphStructure> element describes the list of <NamedElement> elements that constitute a single paragraph.
16.1.2.6  

**<TableStructure> Element**

*element TableStructure*

<table>
<thead>
<tr>
<th>diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Diagram" /></td>
</tr>
</tbody>
</table>

Contains a complete definition of a table in the XPS Document.

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 [O9.5].

16.1.2.7  

**<TableRowGroupStructure> Element**

*element TableRowGroupStructure*

<table>
<thead>
<tr>
<th>diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image2" alt="Diagram" /></td>
</tr>
</tbody>
</table>

Contains the set of table rows that make up a table.

A `<TableRowGroupStructure>` element is REQUIRED in order to specify a set of `<TableRowStructure>` elements [M2.72].

16.1.2.8  

**<TableRowStructure> Element**

*element TableRowStructure*

<table>
<thead>
<tr>
<th>diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image3" alt="Diagram" /></td>
</tr>
</tbody>
</table>

Contains the set of table cells that make up a row of a table.

This element groups `<TableCellStructure>` child elements that define a single row of a table.
16.1.2.9 `<TableCellStructure>` Element

element `TableCellStructure`

This element defines the appearance of a table cell. It MAY contain nested `<TableStructure>` elements [M2.72].

16.1.2.10 `<ListStructure>` Element

element `ListStructure`

The `<ListStructure>` element is the complete definition of a list of related items.
16.1.2.11 <ListItemStructure> Element

A <ListItemStructure> element defines a single item in a list.

16.1.2.12 <FigureStructure> Element

A <FigureStructure> element includes a group of named elements that comprise a single drawing or diagram.
16.1.2.13 `<NamedElement>` Element

A `<NamedElement>` references a specific element in the fixed page by using the NameReference attribute to specify an element in the fixed page markup with a corresponding name.

If the targeted fixed page uses markup compatibility markup that changes the presence of certain named elements, the StoryFragments part should also use it in order to reference each element in either representation.

16.2 Hyperlinks

If consumers enable user interactivity, they SHOULD support hyperlink activation and addressing [S9.6].

16.2.1 Hyperlink Activation

Hyperlinks are specified inline on any `<Canvas>`, `<Path>`, or `<Glyphs>` element by means of the FixedPage.NavigateUri attribute. The value of the attribute is the destination URI. 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 [M9.2].

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 [S9.7].

If a producer specifies a FixedPage.NavigateUri attribute on a `<Canvas>` element, consumers MUST treat all child elements of that canvas as having an associated hyperlink [M9.3]. Child or descendant elements can override this value with their own FixedPage.NavigateUri attribute.

Relative internal hyperlinks between FixedPage parts MUST specify, at a minimum, the named address relative to the FixedDocument part [M9.4].
Producers can mark any `<FixedPage>`, `<Canvas>`, `<Path>`, or `<Glyphs>` element as an addressable location within the XPS Document by specifying a value for the Name attribute. The name SHOULD be unique within the scope of the fixed document [S9.8]. If it is not unique, only the first occurrence of the named address is addressable.

These elements, if specified as a `<VisualBrush.Visual>` property element are not addressable by a hyperlink.

It is RECOMMENDED that Name attribute values be unique within an entire fixed document sequence [S9.9]. If they are not, only the first occurrence of the named address is addressable from an external location. 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 [O9.6].

In order to be addressable by either a hyperlink or the document outline, the named address MUST appear in the `<PageContent.LinkTargets>` element in the fixed document [M9.5]. 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 [M9.6]. If the named address in a URI fragment is not found, consumers MUST ignore the fragment portion of the URI [M9.7].

Example 16–8. A relative, internal, named-address hyperlink

```xml
  FixedPage.NavigateUri="../MyDocument.fdoc#MyAddress"
end example]
```

16.2.2 Hyperlink Addressing

XPS Documents specify two forms of URI fragment identifiers to address locations within an XPS Document. The first is a named address. [Example: "http://xps/MyPackage#MyAddress", where "http://xps/MyPackage" is an XPS Document and "MyAddress" is a named address within the document. end example] The second is an absolute page number within the XPS Document. [Example: "http://xps/MyPackage#15", where "15" references the FixedPage part associated with the fifteenth `<PageContent>` entry among all the fixed documents in the fixed document sequence. end example]

Page number fragment identifiers refer to the absolute page number (1-based) in the fixed document sequence. [Example: If an XPS Document has a 3-page fixed document, followed by a 10-page fixed document, followed by an 8-page fixed document, the fragment identifier 

"#15" refers to the second page of the third fixed document in the fixed document sequence. end example] Internal references MUST specify a page address relative to the fixed document sequence [M9.8].

Example 16–9. A relative internal page address hyperlink

```xml
  FixedPage.NavigateUri="../../.../MyDocSeq.fdseq#12"
end example]
```

16.2.3 Name Attribute

The Name attribute contains a string value that identifies the current element as a named, addressable point for the purpose of hyperlinking. The Name attribute is optional. Names SHOULD be unique within a fixed document [S9.8], and it is RECOMMENDED that they be
unique within a fixed document sequence [S9.9]. The Name attribute MUST NOT be specified on any children of a `<ResourceDictionary>` element [M9.10].

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 [S9.10]. Consumers MAY ignore this attribute [O9.7], but devices that support user interaction with the contents of XPS Documents SHOULD support hyperlinks [S9.6].

The Name value, if specified, MUST meet the following requirements [M2.72]:

1. The initial character MUST be an underscore character or a letter, that is, it falls within the Lu, Ll, Lo, Lt, and Nl categories [M2.72].
2. Trailing characters MUST be an underscore character or a letter or number, that is, they fall within the Lu, Ll, Lo, Lt, Ni, Mn, Mc, and Nd categories [M2.72].

[Note: These requirements match those of XML identifiers with additional restrictions. end note]

The category abbreviations, as defined within the Unicode Character Database, are partially reproduced in Table 16–2.

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lu</td>
<td>Letter, uppercase</td>
</tr>
<tr>
<td>Ll</td>
<td>Letter, lowercase</td>
</tr>
<tr>
<td>Lt</td>
<td>Letter, titlecase</td>
</tr>
<tr>
<td>Lo</td>
<td>Letter, other</td>
</tr>
<tr>
<td>Mn</td>
<td>Mark, non-spacing</td>
</tr>
<tr>
<td>Mc</td>
<td>Mark, spacing combining</td>
</tr>
<tr>
<td>Nd</td>
<td>Number, decimal</td>
</tr>
<tr>
<td>Ni</td>
<td>Number, letter</td>
</tr>
</tbody>
</table>

### 16.2.4 FixedPage.NavigateUri Attribute

The `FixedPage.NavigateUri` attribute associates a hyperlink URI with an element, making it a hyperlink source. Its value can be a relative or absolute URI that addresses a resource that is internal or external to the XPS Document package, respectively. The base URI used to resolve a relative URI is that of the FixedPage part in which the element with the `FixedPage.NavigateUri` attribute appears. Therefore, a hyperlink to a destination within the fixed document of the source MUST specify the destination in the context of the FixedDocument part [M9.4].

[Example: “../FixedDoc1.fdoc#MyDestination”. end example] A destination in the same fixed document SHOULD be expressed as a relative URI [S9.11].

The `FixedPage.NavigateUri` attribute is OPTIONAL [M2.72]. It SHOULD be included only if the element is intended to be a hyperlink. Consumers MAY ignore this attribute [O9.8], but devices that support user interaction with the contents of XPS Documents SHOULD support hyperlinks [S9.6].
16.3 Selection

Viewing consumers that support interactivity MAY support selection and copying [O9.9]. If selection is supported, consumers SHOULD provide a visual cue over or around selected elements [S9.12]. 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].

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].
16. Document Structure and Interactivity

- 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].

- Within a story fragment, order by `<NamedElement>` references in the StoryFragments part markup.

- Append all un-referenced elements that appear in the fixed page markup, ordered by page number, then markup order.

16.4.2 Screen Reader Applications

Screen reader applications read the contents of the document aloud. A screen reader consumer SHOULD read the document according to its reading order [S9.22]. The application SHOULD use the UnicodeString attribute of each `<Glyphs>` element [S9.23]. In addition, screen readers MAY inspect the Indices attribute to resolve potential ambiguities [O9.13].

If the 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 [S9.24].

If the screen reader provides features to describe images, it SHOULD read the text provided in the AutomationProperties.Name and AutomationProperties.HelpText attributes [S9.25].

If the screen reader provides features to describe hyperlink addresses, it SHOULD read the text provided in the FixedPage.NavigateUri attribute [S9.26].

16.4.3 Text Alternatives for Graphics and Images

Images and graphics SHOULD specify text alternatives for images and graphics to make this content accessible to vision-impaired individuals [S9.27]. There are short and long textual descriptions, specified in the AutomationProperties.Name and AutomationProperties.HelpText attributes of `<Path>` and `<Canvas>`, respectively.

The AutomationProperties.Name attribute SHOULD contain a short description of the basic contents of the image or vector graphic [S9.27]. [Example: “A sitting dog.” end example] The AutomationProperties.HelpText attribute can contain a more detailed description of the image or graphic. [Example: "A cocker spaniel with brown eyes, golden fur, and its tongue hanging out. It is sitting on a beanbag directly facing the camera." end example]


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 [S9.29].

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 [S9.27].
17. XPS Document Package Features

The XPS Document format extends package-level interleaving and digital signatures as described in the OPC specification.

17.1 Interleaving Optimizations

Interleaving concerns the physical organization of XPS Documents, rather than their logical structure. It allows consumers to linearly process 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 specification [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].

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 correctly process 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].

Consumers can choose to parse an XPS Document in a head-first or tail-first manner. Tail-first parsing reveals certain package errors earlier, such as inconsistencies between the ZIP central directory and local file headers. Head-first XPS Document consumers SHOULD attempt to detect inconsistent packages as soon as possible and SHOULD generate an error message, even if they have already processed the pages that resulted in the error [S10.18]. Head-first consumers that discard parts would need to retain the name and length of any discarded part to comply with this recommendation.
## 17.1.1 Empty PrintTicket

An empty PrintTicket has the following form:

```xml
<psf:PrintTicket
datawork" version="1"/>
```

It is RECOMMENDED that one empty PrintTicket be shared for all parts that attach an empty PrintTicket [S10.6].

### 17.1.2 Optimizing Interleaving Order

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 [O10.2].

The optimization strategy is suggested by the consumer architecture. Therefore, interleaving optimization is typically implemented by a software component such as a driver or filter that is specific to (or aware of) the consumer architecture.

#### 17.1.2.1 Single-Threaded Parsing Architectures

An optimal interleaving scheme for consumers with a single-threaded parsing model interleaves parts so that each part that is required to consume a single page (FixedPage, images, and fonts) is contained in the package in its entirety, prior to the FixedPage part being referenced from the FixedDocument part’s markup.

Single-threaded parsing architectures typically require more run-time memory resources than multi-threaded parsing architectures because the context in which a resource is used is unknown at the time the resource is received. This requires deferred processing and additional buffering.

[Note: When interleaving entities containing XML markup, such as the DiscardControl part, the Content Types stream, and the FixedDocument part, there is no guarantee that XML element boundaries will align with piece boundaries in the physical package. This adds a complexity to single-threaded parsing architectures: the parser must be pre-emptable. Certain existing XML parser implementations might require a pre-tokenization step. end note]

Example 17–1. Optimized interleaving for a single-threaded parsing architecture

The following markup describes a sequence of two fixed documents, the first having two FixedPage parts and the second having one FixedPage part:

<table>
<thead>
<tr>
<th>Part/Piece</th>
<th>Markup</th>
</tr>
</thead>
<tbody>
<tr>
<td>Font1.ttf</td>
<td>...binary font data...</td>
</tr>
<tr>
<td>Other resources</td>
<td>...resource data...</td>
</tr>
</tbody>
</table>
| Page1            | `<FixedPage xmlns="http://schemas.microsoft.com/xps/2005/06" ...>  
                   <Glyphs FontURI="Font1.ttf"/>  
                   </FixedPage>` |
| Page1.rels       | `<Relationships xmlns=` |
17.1.2.2 Multi-Threaded Parsing Architectures

An optimal interleaving scheme for consumers with a multi-threaded parsing model interleaves parts so that each resource part that is required to consume a single page (images and fonts) is contained in the package after the FixedPage part referencing it.

Multi-threaded parsing architectures typically require less run-time memory resources than single-threaded parsing architectures because the context in which resources appear is fully determined and, therefore, resources can be processed immediately.

[Note: When interleaving entities containing XML markup, such as the DiscardControl part, the content type stream, and the FixedDocument part, there is no guarantee that XML element boundaries will align with piece boundaries in the physical package. A multi-threaded parsing architecture is naturally suited to address this problem. end note]
Example 17-2. Optimized interleaving for a multi-threaded parsing architecture

The following markup describes a sequence of two FixedDocument parts, the first having two FixedPage parts and the second having one FixedPage part:

<table>
<thead>
<tr>
<th>Part/Piece</th>
<th>Markup</th>
</tr>
</thead>
<tbody>
<tr>
<td>_rels/.rels/[0].piece</td>
<td>&lt;Relationships xmlns=&quot;http://schemas.microsoft.com/package/2005/06/relationships&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;Relationship Type=&quot;StartPart&quot; Target=&quot;Sequence1&quot;/&gt;</td>
</tr>
<tr>
<td>Sequence1/[0].piece</td>
<td>&lt;FixedDocumentSequence xmlns=&quot;http://schemas.microsoft.com/xps/2005/06&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;DocumentReference Source=&quot;FixedDocument1&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;PageContent Source=&quot;Page1&quot;/&gt;</td>
</tr>
<tr>
<td>Page1.rels</td>
<td>&lt;Relationships xmlns=&quot;http://schemas.openxmlformats.org/package/2006/relationships&quot;&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;Relationship Type=&quot;http://schemas.microsoft.com/xps/2005/06/required-resource&quot; Target=&quot;Font1.ttf&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/Relationships&gt;</td>
</tr>
<tr>
<td>Page1</td>
<td>&lt;FixedPage xmlns=&quot;<a href="http://schemas.microsoft.com/xps/2005/06">http://schemas.microsoft.com/xps/2005/06</a>&quot; ...&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;Glyphs FontURI=&quot;Font1.ttf&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/FixedPage&gt;</td>
</tr>
<tr>
<td>Font1.ttf</td>
<td>...binary font data...</td>
</tr>
<tr>
<td>Other resources</td>
<td>...resource data...</td>
</tr>
<tr>
<td>FixedDocument1/[1].last.piece</td>
<td>&lt;PageContent Source=&quot;Page2&quot;/&gt;</td>
</tr>
<tr>
<td>Page2</td>
<td>&lt;FixedPage xmlns=&quot;<a href="http://schemas.microsoft.com/xps/2005/06">http://schemas.microsoft.com/xps/2005/06</a>&quot; ...&gt;</td>
</tr>
<tr>
<td>FixedDocument1/[2].last.piece</td>
<td>...&lt;/FixedPage&gt;</td>
</tr>
<tr>
<td>Sequence1/[1].last.piece</td>
<td>&lt;/FixedDocumentSequence&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;PageContent Source=&quot;Page3&quot;/&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;/FixedDocument&gt;</td>
</tr>
</tbody>
</table>

256 Working Draft 1.1, February 2008
17.1.3 Consuming Interleaved Packages

Consumers MUST be able to consume packages regardless of their interleaving structure [M10.2]. Consumers that lack the resources to process a part MUST indicate an error condition [M10.3]. Such a resource constraint exists when a consumer lacks sufficient memory resources to hold enough of the package to resolve all the references required to process a part.

To address resource constraints:

- Consumers MAY discard FixedPage parts once they have been processed [O10.3]
- Consumers MAY discard FixedDocument and FixedDocumentSequence parts after all their child elements and their closing tags have been processed [O10.4].
- In the absence of explicit directives to the contrary (see §17.1.4), consumers MAY discard parts as directed by the DiscardControl part [O10.5]. Consumers MUST NOT discard any other parts [Example: Such as parts containing fonts, images, or other resources end example] unless they have the ability to access the parts again [M10.4].

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 or until the end of the package is reached (indicating an error condition) [M10.5].

17.1.4 Consumers with Resource Constraints

To produce an XPS Document for streaming consumption by consumers with limited memory resources, some producers MAY choose a suitable interleaving order by modeling the resource management behavior of the consumer [O10.6]. These producers, referred to as drivers, must have specific knowledge of the XPS Document consumer. Due to resource constraints, some consumers are unable to consume arbitrary XPS Documents and always require assistance from an external driver.

When some consumers with limited memory resources receive a XPS Document in a streaming fashion, there might be an opportunity to discard parts when necessary and reload them again when needed. Producers, such as drivers, that target such consumers SHOULD follow these steps [S10.7]:

- Conservatively model the memory usage of the device.
- Interleave pieces of parts in the correct order.
- Decide when certain parts can be discarded by the consumer and inform the consumer within the package stream (see §17.1.4.1).
- 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. Those later references are also updated to refer to the new copy of the resource.

17.1.4.1 DiscardControl Part

In addition to optimally ordering interleaved parts, producers can support consumers with resource constraints by means of the DiscardControl part. The DiscardControl part is a well-known part containing a list of resources that are safe for the consumer to discard.
DiscardControl parts are stored in XPS Documents in an interleaved fashion, allowing a resource-constrained consumer to discard a part as soon as it appears in the DiscardControl part. DiscardControl parts are targeted with a DiscardControl package relationship, as specified in H, “Standard Namespaces and Content Types.” There MUST NOT be more than one DiscardControl package relationship [M10.23]. The DiscardControl part MUST NOT reference itself [M10.6]; doing so is considered an error.

DiscardControl parts that are not well-formed SHOULD NOT be processed and an error SHOULD NOT be reported [S10.8]. The consumer MAY decide to ignore the malformed DiscardControl part in its entirety or from the first malformed node onward [O10.7].

In some cases, producers might rewrite the contents of a package so that parts are provided more than once, allowing consumers to discard a part in order to free resources for additional processing. Each instance of a part MUST be stored as a new, uniquely named part in the package [M10.24].

Example 17-3. A DiscardControl part

```xml
<DiscardControl xmlns="http://schemas.microsoft.com/xps/2005/06/discard-control">
  <!-- May discard partname1 as soon as starting to process page11.xaml -->
  <Discard SentinelPage="/page11.xaml" Target="/partname1" />
  <!-- May discard partname2 as soon as starting to process page13.xaml -->
  <Discard SentinelPage="/page13.xaml" Target="/partname2" />
  ...
</DiscardControl>
```

end example]

17.1.4.1.1 <DiscardControl> Element

element DiscardControl

diagram

annotation Contains a list of resources that are safe for a consumer to discard.

17.1.4.1.2 <Discard> Element

element Discard

diagram

Parts that can be discarded are identified in a `<Discard>` element by the Target attribute value, which is expressed as relative to the package root, and by the SentinelPage attribute value, which identifies the first FixedPage part that no longer requires the discarded part. (The processing order for FixedPage parts is implied by the order of `<PageContent>` element references in the FixedDocument part. Therefore, the value of the SentinelPage attribute is unambiguous.)

If either the Target attribute or the SentinelPage attribute contain an invalid reference (refer outside the package), the respective `<Discard>` element MUST be ignored [M10.7]. 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 [S10.9].

### 17.1.5 Interleaving Optimizations and Digital Signatures

In general, it is not feasible to produce well-ordered, interleaved ZIP packages and apply digital signatures in a way that enables reasonable consumption scenarios for the following reasons:

- The digital signature parts must be known to consumers before they process other signed parts because the selected hash-methods and transforms must be known. A streaming consumer might not be able to access part data after it has been processed for printing.
- Producers cannot create the digital signature parts before producing the signed packages.
- There are cyclic dependencies with signed relationship parts containing the relationship to the signature parts themselves.

Therefore, 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 [S10.10].

### 17.2 Digital Signatures

The digital signature specification for XPS Documents is described in the OPC specification. It allows users to sign arbitrary parts, relationship parts, and individual relationships. Although XPS Documents also use these digital signature mechanisms, they have a specific signature policy and a specific signing mechanism for documents containing co-signing requests.
17. XPS Document Package Features

17.2.1 Signature Policy

This specification 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]. Consumers MUST NOT disable functionality based on whether or not an action will invalidate a signature [M10.9].

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 [M10.10]:
   a. The CoreProperties part.
   b. The Digital Signature Origin part.
c. A Digital Signature Certificate part.
d. PrintTicket parts.
e. DiscardControl parts.

3. All relationships with the following RelationshipTypes (see §H) MUST be signed [M10.10]:
   a. StartPart relationship from the package root
   b. DocumentStructure relationship from a FixedDocument part
   c. StoryFragments relationship from a FixedPage part
   d. Digital Signature Definitions relationship from a FixedDocument part
   e. Required Resource relationship from a FixedPage part
   f. Restricted Font relationship from a FixedDocument part
   g. Thumbnail relationship from a FixedPage part, a FixedDocument part, or the package root

4. All relationships with the following RelationshipTypes MUST be signed if their Target part is signed [M10.10]:
   a. Core Properties relationship
   b. Digital Signature Origin relationship
   c. Digital Signature Certificate relationship from a Digital Signature XML Signature part
   d. PrintTicket relationship
   e. DiscardControl relationship

5. Relationships with the following RelationshipTypes MAY be signed as a group (they MUST NOT be signed individually) [M10.10]:
   a. All Digital Signature XML Signature relationships from the Digital Signature Origin part (signing all relationships of this RelationshipType will cause this signature to break when a new signature is added).

6. All of the above-referenced parts and relationships MUST be signed using a single digital signature [M10.10].

An XPS Document MUST NOT be considered signed according to the XPS Document signing policy if [M10.11]:
   1. Any part not covered by the signing rules above is included in the signature.
   2. Any relationship not covered by the signing rules above is included in the signature.

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 [M10.12]. 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 [O10.8].
17.2.1.2 Signing Validity

An XPS Document digital signature MUST be shown as an *incompliant digital signature* if [M10.13]:

- It violates any of the signing rules described above regarding parts or relationships that MUST or MUST NOT be signed.

An XPS Document digital signature MUST be shown as a *broken digital signature* if [M10.14]:

- It is not an incompliant digital signature, but the signature fails the signature validation routines described in the OPC.

An XPS Document digital signature MUST be shown as a *questionable digital signature* if any of the following are true [M10.15]:

- It is not an incompliant or broken digital signature, but the certificate cannot be authenticated against the certificate authority.
- It is not an incompliant or broken digital signature, but the signed content (parts and relationships) contain elements or attributes from an unknown namespace introduced through the Markup Compatibility mechanisms.

An XPS Document digital signature MAY be shown as a questionable digital signature if [O10.9]:

- It is not an incompliant or broken digital signature, but contains some other detectable problem at the discretion of the consumer.

An XPS Document digital signature MUST be shown as a *valid digital signature* if [M10.16]:

- It is not an incompliant, broken, or questionable digital signature.

17.2.1.3 Adding Signatures

XPS Documents MAY be signed more than once [O10.10]. A user who signs an XPS Document might or might not want to allow any additional signing of the document. 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 [M10.17].

17.2.1.4 Certificate Store

XPS Document signatures MUST NOT refer to a remote certificate store (certificate not contained in the XPS Document). All certificates MUST be stored in the XPS Document either as a Certificate part or in the Digital Signature XML Signature part [M10.18].

17.2.1.5 Printing Signed Documents

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>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PrintInvalidSignatures</td>
<td>Print the job regardless of the validity of the digital signatures. Digital signatures can be ignored.</td>
</tr>
</tbody>
</table>

Table 17–1. JobDigitalSignatureProcessing PrintTicket settings
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.
17. XPS Document Package Features

XPS Specification and Reference Guide

Example 17-4. A SignatureDefinitions part

```xml
<SignatureDefinitions xmlns="http://schemas.microsoft.com/xps/2005/06/signature-definitions">
  <SignatureDefinition SignerName="Dorena Paschke" SpotID="AE2351H">
    <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>2005-08-20</SignBy>
      <SignBy>2005-08-20T23:59:59Z</SignBy>
      <SigningLocation>Redmond, WA</SigningLocation>
    </SignatureDefinition>
  </SignatureDefinition>
</SignatureDefinitions>
```

end example]

17.2.2.1 <SignatureDefinitions> Element

element SignatureDefinitions

If the SignatureDefinitions part exists, it MUST contain only one <SignatureDefinitions> element [M2.72]. The XML namespace for the <SignatureDefinitions> element is specified in §H.1.

17.2.2.2 <SignatureDefinition> Element

element SignatureDefinitionsType/SignatureDefinition
If the SignatureDefinitions part exists, there MUST be at least one `<SignatureDefinition>` element [M2.72].

17.2.2.2.1 SpotID Attribute

The SpotID attribute is REQUIRED [M2.72]. This attribute MAY be used to link an existing signature to the `<SignatureDefinition>` element [O10.12]. The value of this attribute MUST be globally unique to ensure that a Signature part can be linked to only one `<SignatureDefinition>` element [M2.72]. 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 [M10.19]. For more information, see “Digital Signatures” in the OPC specification.

17.2.2.3 `<SpotLocation>` Element

element SignatureDefinitionType/SpotLocation
The `<SpotLocation>` element is OPTIONAL [M2.72]. It specifies where an XPS Document viewer should place a visual representation or *signature spot* to indicate that a digital signature has been applied or requested. The viewing consumer SHOULD use the values specified in this element. Due to space and rendering limitations, producers MUST NOT assume that consumers will use these values [M10.20]. If the location specified by this element is not used, it is RECOMMENDED that consumers choose a location that does not contain any page content [S10.13].

The size and shape of the signature spot are determined by the consumer. Consumers MAY choose a size and shape based on the desired display information and page content [O10.13]. However, it is RECOMMENDED that they 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 [S10.14]. It is also RECOMMENDED that the signature spot be a clickable area used to launch the digital signing process [S10.15].

*Figure 17–1. A sample signature spot*

17.2.2.4 `<Intent>` Element

**element** `SignatureDefinitionType/Intent`
 Consumers MUST display the full value of the `<Intent>` element to the signing party, either in the signature spot or through some other mechanism [M10.21].

[Note: Consumers that wish to display signature spots must consider the implications of supporting any Unicode character that can be specified in the `<Intent>` element. They must also make decisions about the appropriate font face and size to use as well as determine the proper layout and interactivity of the signature spot. These decisions are implementation-specific. end note]
17.2.2.5  <SignBy> Element

element SignatureDefinitionType/SignBy

<table>
<thead>
<tr>
<th>diagram</th>
<th>SignBy</th>
</tr>
</thead>
<tbody>
<tr>
<td>annotation</td>
<td>The date and time by which the requested party is to sign the XPS Document.</td>
</tr>
</tbody>
</table>

If specified, the consumer SHOULD NOT allow the signing party to sign the document using this particular signature spot after the date and time specified [S10.16]. The date and time MUST be specified as a complete date plus hours, minutes, and seconds in UTC time, using the format "Complete date plus hours, minutes and seconds" as described in the W3C Note "Date and Time Formats" [M10.22], [Example: "2006-12-31T23:59:59Z" for 11:59 PM (UTC) on December 31, 2006. end example]

17.2.2.6  <SigningLocation> Element

element SignatureDefinitionType/SigningLocation

<table>
<thead>
<tr>
<th>diagram</th>
<th>SigningLocation</th>
</tr>
</thead>
<tbody>
<tr>
<td>annotation</td>
<td>The legal location where the document is signed.</td>
</tr>
</tbody>
</table>

The <SigningLocation> element MAY be set by the original producer of the XPS Document or by the signing party at the time of signing [O10.14].

17.3 Core Properties

XPS Documents use the Core Properties part described in the OPC. The core properties specified in that part SHOULD refer to the entire fixed payload, including the root FixedDocumentSequence part and the compilation of all FixedDocument parts it references [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 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.
18. Rendering Rules

18.1.3 Transforms

XPS Document markup supports affine transforms as expressed through the RenderTransform and Transform properties. An affine transform is represented as a list of six real numbers: m11, m12, m21, m22, dx, dy. (For markup details, see §14.4.)

The full matrix is as follows:

\[
\begin{bmatrix}
    m_{11} & m_{12} & 0 \\
    m_{21} & m_{22} & 0 \\
    dx & dy & 1
\end{bmatrix}
\]

A given $x,y$ coordinate is transformed with a render transform to yield the resulting coordinate $x',y'$ by applying the following computations:

\[
x' = x \times m_{11} + y \times m_{21} + dx
\]

\[
y' = x \times m_{12} + y \times m_{22} + dy
\]

When rendering a child or descendant element, the effective transform used for rendering is the concatenation of all the transforms specified by the RenderTransform or Transform property on parent or ancestor elements, starting from the outermost ancestor.

Non-invertible effective transforms can be specified in markup or occur as a result of limited numerical precision during concatenation. If a non-invertible transform is encountered during rendering, consumers MUST omit rendering the affected element and all of its child and descendant elements [M11.2].

If a non-invertible transform is encountered on a brush (as specified directly on the brush, as a result of the Viewbox or Viewport attributes, or through concatenation), the brush is treated according to §18.7.1.

The Width and Height values specified in the Viewbox and Viewport attributes of an <ImageBrush> or <VisualBrush> element MUST NOT be negative [M2.72].

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 [M11.3].

A final, device-dependent step using the horizontal resolution and vertical resolution of the device converts the resulting coordinates $x',y'$ to device coordinates $x'',y''$, as follows:

\[
x'' = x' \times R_x/96
\]

\[
y'' = y' \times R_y/96
\]

Where $R_x$ is the horizontal resolution and $R_y$ is the vertical resolution of the device, specified in device pixels per inch.

18.1.4 Pixel Center Location, Pixel Placement, and Pixel Inclusion

A pixel covers the range from $x$ to $x+1$.

An ideal 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 [S11.2]. Other implementations MAY use different rendering logic as long as it closely approximates this logic [O11.2].
When rendering a shape, a practical 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 [S11.3]. Devices MAY use sub-pixel masking instead [O11.3].

By definition, a shape with an area width of 0 (that is, no included area) does not touch or cover any pixel centers. A stroke with a width of 0 is treated in the same manner.

As a result of these rules, the behavior for very thin lines is implementation-specific:

- An implementation capable of anti-aliasing MAY draw a thin line in a way that blends with the background to varying degrees [O11.4].
- A bi-tonal implementation on a printer MAY draw thin lines, with or without drop-outs, or by applying half-toning, depending on the desired output quality [O11.5]. If such 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 above, or as solid rules of 1 pixel thickness [O11.26].

Note: Also see section §18.6.12 for discussion of thin strokes. end note

18.1.5 Maximum Placement Error

When rendering geometries, consumers SHOULD render curves so they appear smooth from a normal viewing distance [S11.4]. Producers MUST NOT assume a specific placement error for curve decomposition or rely on side-effects of a specific consumer implementation [M11.4].

18.1.6 Pixel Placement for Glyphs

Regardless of other rules expressed here, consumers MAY apply pixel placement rules optimized for character rendering to individual glyphs in a <Glyphs> element [O11.6]. Such rules can result from font hinting applied by the typeface scaler used by a consumer implementation.

18.1.7 Abutment of Shapes

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 [S11.5]. 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 [S11.5].

18.1.8 Clipping Behavior

Clipping occurs as if a mask were created from the clip geometry according to the pixel placement rules defined in §18.1.4. 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 [S11.6].

A practical implementation (such as a bi-tonal printing device) SHOULD create a pixel mask according to Point2 of §18.1.4, 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 [S11.7]. Devices MAY use sub-pixel masking instead [O11.3].
18. Rendering Rules

18.2 Implementation Limits

XPS Document markup does not assume fixed implementation limits. However, consumers can have specific implementation limits imposed by their operating environment. XPS Document markup has been designed so that even complex pages can be represented accurately and with high fidelity.

A typical consumer implementation SHOULD be able to process markup with the characteristics indicated in Table 18–1 [S11.8]. Encountering markup with characteristics outside of the consumer-specific implementation limits MUST cause an error condition [M11.5].

Table 18–1 provides the RECOMMENDED minimum requirements for individual elements [S11.8]. Consumers also have limits on the total number of elements, as imposed by available memory. Producers SHOULD produce only XPS Documents that stay within these implementation limits [S11.8].

In order to process pages that contain a large number of elements, consumers MAY implement support for the DiscardControl part in order to discard elements that have already been processed [O10.5].

Table 18–1. Recommended minimum processing requirements

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Type</th>
<th>Limit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coordinates/transformation matrix elements</td>
<td>Real number</td>
<td>$\pm 10^{12}$</td>
<td>Largest and smallest coordinate values. Calculations involving numbers close to this limit within a few orders of magnitude are likely to be inaccurate.</td>
</tr>
<tr>
<td>Smallest representable non-zero value</td>
<td></td>
<td>$\pm 10^{-12}$</td>
<td>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.</td>
</tr>
<tr>
<td>Required precision for coordinates</td>
<td>Single floating point</td>
<td></td>
<td>Coordinates are real numbers and SHOULD be computed with at least single floating point precision [S11.1, S11.9].</td>
</tr>
<tr>
<td>Nested $&lt;$Canvas$&gt;$ elements</td>
<td>At-least-16</td>
<td></td>
<td>Depth of nested $&lt;$Canvas$&gt;$ elements.</td>
</tr>
<tr>
<td>Nested $&lt;$VisualBrush$&gt;$ elements</td>
<td>At-least-16</td>
<td></td>
<td>Depth of nested $&lt;$VisualBrush$&gt;$ 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].</td>
</tr>
<tr>
<td>Total number of points in a path figure</td>
<td></td>
<td>100,000</td>
<td></td>
</tr>
<tr>
<td>Total number of points in a segment</td>
<td></td>
<td>100,000</td>
<td></td>
</tr>
<tr>
<td>Total number of points in a geometry</td>
<td></td>
<td>100,000</td>
<td></td>
</tr>
<tr>
<td>Total number of elements per page</td>
<td></td>
<td>1,000,000</td>
<td></td>
</tr>
</tbody>
</table>
### 18. Rendering Rules

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Type</th>
<th>Limit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of glyphs per <code>&lt;Glyphs&gt;</code> element</td>
<td></td>
<td>5,000</td>
<td></td>
</tr>
<tr>
<td>Number of elements in a single resource dictionary</td>
<td></td>
<td>10,000</td>
<td></td>
</tr>
<tr>
<td>Total number of elements in all resource dictionaries of an individual page</td>
<td></td>
<td>10,000</td>
<td></td>
</tr>
<tr>
<td>Total number of resource dictionaries in nested canvas scope</td>
<td>Number of nested <code>&lt;Canvas&gt;</code> elements + 1</td>
<td>Number of nested <code>&lt;Canvas&gt;</code> elements + 1</td>
<td>The <code>&lt;FixedPage&gt;</code> element and each nested <code>&lt;Canvas&gt;</code> element can have at most one associated <code>&lt;ResourceDictionary&gt;</code> element</td>
</tr>
<tr>
<td>Number of gradient stops in a gradient brush</td>
<td></td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>Number of fixed documents in a fixed document sequence</td>
<td></td>
<td>1,000</td>
<td></td>
</tr>
<tr>
<td>Number of fixed pages in a fixed document</td>
<td></td>
<td>1,000,000</td>
<td></td>
</tr>
<tr>
<td>Number of dash-gap segments in StrokeDashArray property</td>
<td>No preset limit</td>
<td>No preset limit</td>
<td>Practical number of dash-gap segments depends on the StrokeThickness and the total length of the stroked path.</td>
</tr>
<tr>
<td>Total size of XPS Document markup per page</td>
<td>Bytes</td>
<td>64,000,000</td>
<td>Total size of markup after removing all unnecessary whitespace (according to the schema in B), assuming markup elements are specified in the default namespace without namespace prefixes, and assuming the most compact representation of all attributes using abbreviated syntax where possible.</td>
</tr>
</tbody>
</table>

#### 18.3 Gradient Computations

To ensure the greatest possible consistency among consumers, gradients SHOULD be rendered according to the guidelines described in this [subclause](#) section [S11.11].

#### 18.3.1 All Gradients

Linear gradients and radial gradients share a common set of recommended operations for pre-processing gradient stops and blending colors. These are described below.

##### 18.3.1.1 Gradient Stop Pre-Processing

Consumers SHOULD pre-process gradient stops for all gradients using the following steps [S11.12]:

1. Sort all gradient stops by their respective offset values in ascending order. When two or more gradient stops have the same offset value, preserve their relative order from the markup while sorting. When more than two gradient stops have the same offset value, remove all but the first and last gradient stops having the same offset value.

2. If no gradient stop with an offset of 0.0 exists,
18. Rendering Rules

a. And no gradient stop with an offset less than 0.0 exists, create an artificial gradient stop having an offset of 0.0 and a color of the gradient stop with the smallest offset value.

b. And a gradient stop with an offset less than 0.0 exists and a gradient stop with an offset greater than 0.0 exists, create an artificial gradient stop having an offset of 0.0 and a color interpolated between the two gradient stops surrounding 0.0. Discard all gradient stops with an offset less than 0.0.

c. And a gradient stop with an offset less than 0.0 exists and no gradient stop with an offset greater than 0.0 exists, create an artificial gradient stop having an offset of 0.0 and a color of the gradient stop with the largest offset value. Discard all gradient stop elements with an offset less than 0.0.

3. If no gradient stop with an offset of 1.0 exists,

a. And no gradient stop with an offset of greater than 1.0 exists, create an artificial gradient stop having an offset of 1.0 and a color equal to the color of the gradient stop with the largest offset value.

b. And a gradient stop with an offset greater than 1.0 exists and a gradient stop with an offset less than 1.0 exists, create an artificial gradient stop having an offset of 1.0 and a color interpolated between the two surrounding gradient stops. Discard all gradient stops with an offset greater than 1.0.

c. And a gradient stop with an offset greater than 1.0 exists and no gradient stop with an offset less than 1.0 exists, create an artificial gradient stop having an offset of 1.0 and a color equal to that of the gradient stop with the smallest offset value. Discard all gradient stops with an offset greater than 1.0.

18.3.1.2 Blending Colors

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 [S11.13]. 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 [S11.13].

The function used for blending is:

\[
\text{BLEND}(\text{offset}, \ c_{lo}, \ c_{hi})
\]

Where the offset is between 0 and 1. \( c_{lo} \) and \( c_{hi} \) designate the color values for an offset of 0 and 1, respectively.

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 [S11.14].

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 [S11.15].

If blending is performed in the color space identified by the PageBlendColorSpace PrintTicket setting, it SHOULD be a linear, channel-by-channel blend operation [S11.13].
In the presence of transformations or when individual gradient stops are very close (separated by a few pixels or less in the device space), 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 BLEND() function interpolate the gradient over the extent of each device pixel [S11.16]. However, the behavior MAY differ from this recommendation in an implementation-specific manner [O11.7] and, therefore, producers SHOULD NOT rely on a specific effect for such dense gradient specifications [S11.16].

As a consequence of this interpolation, radial gradients that define the gradient origin on or outside ellipse create an “outside” area that can be rendered inconsistently. The radial gradients that are affected are those that define multiple gradient stops that are of different colors and are very close in Offset value to 0.0 or 1.0 (the gradient end points), for radial gradients with a SpreadMethod value of Repeat or Reflect, respectively. For these affected gradients, consumers MAY use an interpolated color value for the outside area [O11.8]. Depending on the resolution, this can result in different colors than those defined by the gradient end points. The closer a gradient stop is to the affected gradient end point, the more the rendering results on different consumers and at different display resolutions can differ. Producers SHOULD therefore either avoid such 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 [S11.17].

18.3.2 Linear Gradients
Consumers SHOULD render an element filled with a linear gradient brush such that the appearance is the same as if the following steps had been taken [S11.11]:

1. Transform the StartPoint and EndPoint attribute values using the current effective render transform (including the render transform for the element being filled by the linear gradient brush and the brush’s transform itself).

2. If the SpreadMethod value is Pad, the colors of points on the line defined by the StartPoint and EndPoint attributes are defined by interpolating the coordinates linearly, and each color component (such as R, G, B for sRGB and scRGB) as well as the alpha component is interpolated between the component values of the closest enclosing gradient stops:

For each offset (real number) \( t < 0 \):

\[
\begin{align*}
    x(t) &= (\text{EndPoint}_x - \text{StartPoint}_x) \cdot t + \text{StartPoint}_x \\
    y(t) &= (\text{EndPoint}_y - \text{StartPoint}_y) \cdot t + \text{StartPoint}_y \\
    c(t) &= c_{\text{first}} \\
    a(t) &= a_{\text{first}}
\end{align*}
\]

Where \( c \) is the color component and \( a \) is the alpha component. \( c_{\text{first}} \) are the color component values of the first gradient stop (after sorting) and \( a_{\text{first}} \) is the alpha component value at the first gradient stop (after sorting).

For each offset (real number) \( 0 \leq t \leq 1 \):

\[
\begin{align*}
    x(t) &= (\text{EndPoint}_x - \text{StartPoint}_x) \cdot t + \text{StartPoint}_x \\
    y(t) &= (\text{EndPoint}_y - \text{StartPoint}_y) \cdot t + \text{StartPoint}_y \\
    c(t) &= \text{BLEND}((t-t_{i0})/(t_{hi}-t_{i0}), c_{i0}, c_{hi}) \\
    a(t) &= \left[ (t-t_{i0})/(t_{hi}-t_{i0}) \right] \cdot (a_{hi}-a_{i0}) + a_{i0}
\end{align*}
\]
Where $t_{lo}$ and $t_{hi}$ are the offsets, $c_{lo}$ and $c_{hi}$ are the color component values at the closest enclosing gradient stops (that is, $t_{lo} \leq t \leq t_{hi}$) and $a_{lo}$ and $a_{hi}$ are the alpha component values at the closest enclosing gradient stops ($t_{lo} \leq t \leq t_{hi}$).

For each offset (real number) $t > 1$:

\[
\begin{align*}
    x(t) &= (\text{EndPoint}_x - \text{StartPoint}_x) \times t + \text{StartPoint}_x \\
    y(t) &= (\text{EndPoint}_y - \text{StartPoint}_y) \times t + \text{StartPoint}_y \\
    c(t) &= c_{\text{last}} \\
    a(t) &= a_{\text{last}}
\end{align*}
\]

Where $c_{\text{last}}$ are the color component values of the last gradient stop (after sorting) and $a_{\text{last}}$ is the alpha component value at the last gradient stop (after sorting).

3. If the SpreadMethod value is Repeat, the colors of points on the line defined by the StartPoint and EndPoint attributes are defined by interpolating the coordinates linearly, and each color component (such as R, G, B for sRGB and scRGB) as well as the alpha component is interpolated between the component values of the closest enclosing gradient stops:

For each repetition (all integers) $N$:

\[
\begin{align*}
    &\text{For each offset (real number) } 0 \leq t < 1: \\
    &\quad \begin{align*}
        &x(t) = (\text{EndPoint}_x - \text{StartPoint}_x) \times (N+t) + \text{StartPoint}_x \\
        &y(t) = (\text{EndPoint}_y - \text{StartPoint}_y) \times (N+t) + \text{StartPoint}_y \\
        &c(t) = \text{BLEND}((t-t_{lo})/(t_{hi}-t_{lo}), c_{lo}, c_{hi}) \\
        &a(t) = [(t-t_{lo})/(t_{hi}-t_{lo})] \times (a_{hi}-a_{lo})+a_{lo}
    \end{align*}
\end{align*}
\]

Where $c$ is the color component and $a$ is the alpha component. $t_{lo}$ and $t_{hi}$ are the offsets, $c_{lo}$ and $c_{hi}$ are the color component values at the closest enclosing gradient stops (that is, $t_{lo} \leq t \leq t_{hi}$) and $a_{lo}$ and $a_{hi}$ are the alpha component values at the closest enclosing gradient stops ($t_{lo} \leq t \leq t_{hi}$).

4. If the SpreadMethod value is Reflect, the colors of points on the line defined by the StartPoint and EndPoint attributes are defined by interpolating the coordinates linearly, and each color component (such as R, G, B for sRGB and scRGB) as well as the alpha component is interpolated between the component values of the closest enclosing gradient stops:

For each repetition (all integers) $N$:

\[
\begin{align*}
    &\text{For each offset (real number) } 0 \leq t \leq 1: \\
    &\quad \begin{align*}
        &\text{If (N is EVEN)} \\
        &\quad \begin{align*}
            &x(t) = (\text{EndPoint}_x - \text{StartPoint}_x) \times (N+t) + \text{StartPoint}_x \\
            &y(t) = (\text{EndPoint}_y - \text{StartPoint}_y) \times (N+t) + \text{StartPoint}_y
        \end{align*}
        &\text{Else} \\
        &\quad \begin{align*}
            &x(t) = (\text{EndPoint}_x - \text{StartPoint}_x) \times (N+1-t) + \text{StartPoint}_x \\
            &y(t) = (\text{EndPoint}_y - \text{StartPoint}_y) \times (N+1-t) + \text{StartPoint}_y
        \end{align*}
    \end{align*}
\end{align*}
\]

276 Working Draft 1.1, February 2008
where \( c \) is the color component and \( a \) is the alpha component. \( t_{10} \) and \( t_{hi} \) are the offsets, \( c_{10} \) and \( c_{hi} \) are the color component values at the closest enclosing gradient stops (that is, \( t_{10} <= t <= t_{hi} \)), and \( a_{10} \) and \( a_{hi} \) are the alpha component values at the closest enclosing gradient stops (\( t_{10} <= t <= t_{hi} \)).

5. The colors of points not on the extended line defined by the StartPoint and EndPoint attributes are the same as the color of the closest point on the line defined by the StartPoint and EndPoint attributes, measured in the coordinate space as transformed by the current effective render transform (including the render transform for the element being filled by the linear gradient brush and the brush’s transform itself).

6. Clip the resulting set of points to the intersection of the current clip geometry and the path or glyphs to be filled. Both the clip and path (or glyphs) must be transformed according to the current effective render transform, including the render transform for the element being filled, but not including the transform of the linear gradient brush.

For purposes of the above steps, the closest enclosing gradient stops mean the gradient stops that, if the relative sequencing of the gradient stop offsets in the markup order is respected, are numerically closest to the interpolation point if that interpolation point were converted to an offset value and inserted in a sorted fashion into the list of gradient stops. [Example: If a gradient contains three gradient stops at offset values 0.0, 0.0, and 1.0, the closest enclosing gradient stops for any value 0 <= value <= 1 are the second gradient stop (offset 0.0) and the third gradient stop (offset 1.0). end example]

18.3.3 Radial Gradients

Consumers SHOULD render an element filled with a radial gradient brush such that the appearance is the same as if these steps had been followed [S11.11]:

1. The boundary of the area filled by a radial gradient brush is defined by interpolating ellipses from the GradientOrigin value to the circumference of the ellipse centered at the point specified by the Center attribute with radii equal to the RadiusX and RadiusY attribute values (the interpolated ellipses and point being transformed by the current effective render transform, including the render transform for the element being filled by the radial gradient brush and the brush’s transform itself). If the gradient origin is outside the circumference of the ellipse specified, the effect will be as if a cone were drawn, tapering to the gradient origin.

2. If the SpreadMethod value is Pad, the centers and radii of the interpolated ellipses are defined by linearly interpolating the center of the ellipse from the GradientOrigin attribute value to the Center attribute value, and simultaneously linearly interpolating the radii of the ellipse from 0 to the RadiusX and RadiusY attribute values:

For each offset (real number) 0 <= t <= 1:

```plaintext
{ 
    \( c_x(t) = (Center_x\text{-}GradientOrigin_x)\cdot t + GradientOrigin_x \)
    \( c_y(t) = (Center_y\text{-}GradientOrigin_y)\cdot t + GradientOrigin_y \)
    \( r_x(t) = \text{RadiusX}\cdot t \)
    \( r_y(t) = \text{RadiusY}\cdot t \)
}
```
The ellipses defined by the interpolation are transformed by the current effective render transform, including the render transform for the element being filled by the radial gradient brush and the brush's transform itself.

3. The colors of the points within the boundary of this shape are defined as the color of the smallest interpolated ellipse containing the point. The color of an interpolated ellipse is defined by interpolating each color component (such as R, G, B for sRGB and scRGB) as well as the alpha component between the component values of the closest enclosing gradient stops:

For each offset (real number) 0 \leq t \leq 1:
\{ 
  c(t) = \text{BLEND}((t-t_{l0})/(t_{hi}-t_{l0}),c_{l0},c_{hi}) 
  a(t) = [(t-t_{l0})/(t_{hi}-t_{l0})]*(a_{hi}-a_{l0})+a_{l0} 
\}

Where \( t_{l0} \) and \( t_{hi} \) are the offsets, \( c_{l0} \) and \( c_{hi} \) are the color component values at the closest enclosing gradient stops (that is, \( t_{l0} \leq t \leq t_{hi} \)) and \( a_{l0} \) and \( a_{hi} \) are the alpha component values at the closest enclosing gradient stops (\( t_{l0} \leq t \leq t_{hi} \)).

4. If the SpreadMethod value is Repeat, the centers and radii of the interpolated ellipses are defined by linearly interpolating the center of the ellipse from the GradientOrigin attribute value to the Center attribute value, and simultaneously linearly interpolating the radii of the ellipse from 0 to RadiusX and RadiusY attribute values:

For each repetition (all non-negative integers) \( N \):
\{ 
  For each offset (real number) 0 \leq t < 1:
  \{ 
    c_x(t) = (Center_x-\text{GradientOrigin}_x)*(N+t)+\text{GradientOrigin}_x 
    c_y(t) = (Center_y-\text{GradientOrigin}_y)*(N+t)+\text{GradientOrigin}_y 
    r_x(t) = \text{RadiusX}*(N+t) 
    r_y(t) = \text{RadiusY}*(N+t) 
  \}
\}

The ellipses defined by the interpolation are transformed by the current effective render transform, including the render transform for the element being filled by the radial gradient brush and the brush's transform itself.

5. The colors of the points within the boundary of this shape are defined as the color of the smallest interpolated ellipse containing the point. The color of an interpolated ellipse is defined by interpolating each color component (such as R, G, B for sRGB and scRGB) as well as the alpha component between the component values of the closest enclosing gradient stops:

For each repetition (all non-negative integers) \( N \):
\{ 
  For each offset (real number) 0 \leq t < 1:
  \{ 
    c(t) = \text{BLEND}((t-t_{l0})/(t_{hi}-t_{l0}),c_{l0},c_{hi}) 
    a(t) = [(t-t_{l0})/(t_{hi}-t_{l0})]*(a_{hi}-a_{l0})+a_{l0} 
  \}
\}

Where \( t_{l0} \) and \( t_{hi} \) are the offsets, \( c_{l0} \) and \( c_{hi} \) are the color component values at the closest enclosing gradient stops (that is, \( t_{l0} \leq t \leq t_{hi} \)) and \( a_{l0} \) and \( a_{hi} \) are the alpha component values at the closest enclosing gradient stops (\( t_{l0} \leq t \leq t_{hi} \)).
6. If the SpreadMethod value is Reflect, the centers and radii of the interpolated ellipses are defined by linearly interpolating the center of the ellipse from the GradientOrigin attribute value to the Center attribute value, and simultaneously linearly interpolating the radii of the ellipse from 0 to the RadiusX and RadiusY attribute values:

For each non-negative integer N:

{ For each offset (real number) 0 <= t <= 1:

\[
\begin{align*}
  c_x(t) &= (Center_x - GradientOrigin_x)(N+t) + GradientOrigin_x \\
  c_y(t) &= (Center_y - GradientOrigin_y)(N+t) + GradientOrigin_y \\
  r_x(t) &= RadiusX(N+t) \\
  r_y(t) &= RadiusY(N+t)
\end{align*}
\]

} } The ellipses defined by the interpolation are transformed by the current effective render transform, including the render transform for the element being filled by the radial gradient brush and the brush's transform itself.

7. The colors of the points within the boundary of this shape are defined as the color of the smallest interpolated ellipse containing the point. The color of an interpolated ellipse is defined by interpolating each color component (such as R, G, B for sRGB and scRGB) as well as the alpha component between the component values of the closest enclosing gradient stops:

For each non-negative integer N:

{ For each offset (real number) 0 <= t <= 1:

\[
\begin{align*}
  c(t) &= \text{BLEND}((t' - t_{lo})/(t_{hi} - t_{lo}), c_{lo}, c_{hi}) \\
  a(t) &= [(t' - t_{lo})/(t_{hi} - t_{lo})](a_{hi} - a_{lo}) + a_{lo}
\end{align*}
\]

} } Where \( t_{lo} \) and \( t_{hi} \) are the offsets, \( c_{lo} \) and \( c_{hi} \) are the color component values at the closest enclosing gradient stops (that is, \( t_{lo} \leq t \leq t_{hi} \)) and \( a_{lo} \) and \( a_{hi} \) are the alpha component values at the closest enclosing gradient stops (\( t_{lo} \leq t \leq t_{hi} \)).

8. The colors of points outside the boundary of this shape (points which cannot be drawn by any combination of non-negative N and t) are defined as having the color and alpha defined in the gradient stop with the offset of 0.0 for radial gradients with a SpreadMethod value of Reflect and the color and alpha defined in the gradient stop with the offset of 1.0 for radial gradients with a SpreadMethod value of Repeat or Pad. The colors outside of the boundary of this shape can also vary in an implementation-specific manner (see §18.3.1.2 for more details).

9. Clip the resulting set of points by the intersection of the current clip geometry and the path or glyphs to be filled. Both the clip and path (or glyphs) must be transformed according to the current effective render transform, including the render transform for the element being filled, but not including the transform of the radial gradient brush.
For purposes of the above steps, the closest enclosing gradient stops mean the gradient stops that, if the relative sequencing of the gradient stop offsets in the markup order is respected, are numerically closest to the interpolation point if that interpolation point were converted to an offset value and inserted in a sorted fashion into the list of gradient stops. [Example: If a gradient contains three gradient stops at offset values 0.0, 0.0, and 1.0, the closest enclosing gradient stops for any value 0 ≤ value ≤ 1 are the second gradient stop (offset 0.0) and the third gradient stop (offset 1.0). end example]

18.4 Opacity Computations

Opacity is used to transparently blend two elements when rendering, also known as alpha blending. The value of the Opacity property ranges from 0.0 (fully transparent) to 1.0 (fully opaque), inclusive. Values outside of this range are invalid.

The opacity is applied through the following computations, assuming source and destination values are not pre-multiplied. All opacity calculations SHOULD be performed with at least 8-bit precision to provide sufficient quality for nested content [S11.18].

Individual pixels are blended as defined below.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>O_e</td>
<td>Opacity attribute of element</td>
</tr>
<tr>
<td>O_m</td>
<td>Alpha value at corresponding pixel position in the OpacityMask attribute value</td>
</tr>
<tr>
<td>a_s</td>
<td>Alpha value present in source color</td>
</tr>
<tr>
<td>r_s</td>
<td>Red value present in source color</td>
</tr>
<tr>
<td>g_s</td>
<td>Green value present in source color</td>
</tr>
<tr>
<td>b_s</td>
<td>Blue value present in source color</td>
</tr>
<tr>
<td>a_d</td>
<td>Alpha value already present in destination surface</td>
</tr>
<tr>
<td>r_d</td>
<td>Red value already present in destination surface</td>
</tr>
<tr>
<td>g_d</td>
<td>Green value already present in destination surface</td>
</tr>
<tr>
<td>b_d</td>
<td>Blue value already present in destination surface</td>
</tr>
<tr>
<td>a*</td>
<td>Resulting Alpha value for destination surface</td>
</tr>
<tr>
<td>r*</td>
<td>Resulting Red value for destination surface</td>
</tr>
<tr>
<td>g*</td>
<td>Resulting Green value for destination surface</td>
</tr>
<tr>
<td>b*</td>
<td>Resulting Blue value for destination surface</td>
</tr>
</tbody>
</table>

All values designated with a T subscript (as in r_T) are temporary values.

The opacity is calculated as follows:

1. Multiply source alpha value with opacity value and alpha value of opacity mask.

\[ A_{S1} = A_s * O_e * O_m \]

2. Pre-multiply source alpha.

Omit this step if the source data specifies pre-multiplied alpha (see §18.4.1 for details).
A\textsubscript{T1} = A\textsubscript{s1}
R\textsubscript{T1} = R\textsubscript{s}A\textsubscript{s1}
G\textsubscript{T1} = G\textsubscript{s}A\textsubscript{s1}
B\textsubscript{T1} = B\textsubscript{s}A\textsubscript{s1}

3. Pre-multiply destination alpha.

Omit this step in consumers supporting superluminous colors (see §18.4.1 for details).

A\textsubscript{T2} = A\textsubscript{0}
R\textsubscript{T2} = R\textsubscript{0}A\textsubscript{0}
G\textsubscript{T2} = G\textsubscript{0}A\textsubscript{0}
B\textsubscript{T2} = B\textsubscript{0}A\textsubscript{0}

4. Blend.

See §18.4.1 for special case handling.

A\textsubscript{T3} = (1-A\textsubscript{T1})A\textsubscript{T2}+A\textsubscript{T1}
R\textsubscript{T3} = (1-A\textsubscript{T1})R\textsubscript{T2}/A\textsubscript{T3}+R\textsubscript{T1}
G\textsubscript{T3} = (1-A\textsubscript{T1})G\textsubscript{T2}/A\textsubscript{T3}+G\textsubscript{T1}
B\textsubscript{T3} = (1-A\textsubscript{T1})B\textsubscript{T2}/A\textsubscript{T3}+B\textsubscript{T1}

5. Reverse pre-multiplication.

Omit this step in consumers supporting superluminous colors. See also §18.4.1.

If \text{A\textsubscript{T3}} = 0

\{ set all A* R* G* B* to 0. \}

Else

\{ \text{A* = A\textsubscript{T3}}
\text{R* = R\textsubscript{T3}/A\textsubscript{T3}}
\text{G* = G\textsubscript{T3}/A\textsubscript{T3}}
\text{B* = B\textsubscript{T3}/A\textsubscript{T3}} \}

When blending colors in a color space other than sRGB, color channels are independently interpolated in a manner analogous to the RGB channel blending method described above.
18.4.1 Pre-Multiplied Alpha and Superluminous Colors

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 [M11.6]. In certain scenarios (such as when rendering 3D scenes to a bitmap), producers MAY choose to create pre-multiplied bitmap data specifying “superluminous” colors [O11.9].

Superluminous colors are defined as pre-multiplied color values with an alpha value smaller than the individual color channel values but greater than or equal to 0.

The effect of composing superluminous colors on a background is similar to adding additional light of the source color to the destination, as opposed to regular alpha composition which works more like a colored filter. One can easily verify this statement by substituting 0 for AT1 in step 4 of the above opacity computations, which is simplified as follows (note that R T1 , G T1 , B T1 are not 0, because the pre-multiplication in step 2 has been skipped):

\[
\begin{align*}
A_{T3} &= A_{T2} \\
R_{T3} &= R_{T2} + R_{T1} \\
G_{T3} &= G_{T2} + G_{T1} \\
B_{T3} &= B_{T2} + B_{T1}
\end{align*}
\]

Consumers supporting superluminous colors retain all temporary information in pre-multiplied formats. Note, that throughout the XPS specification non-pre-multiplied alpha processing is assumed. It is up to the implementer of such a consumer to identify equivalent composition and rendering rules for processing in pre-multiplied space.

Also note, 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 [S11.19].

Consumers MAY handle superluminous colors 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 [O11.10], as described in the following steps:

For each superluminous pixel with \( A < R \) or \( A < G \) or \( A < B \)

\{
  
  If \( A = 0 \)
  
  \{
    R^* = 1 \\
    G^* = 1 \\
    B^* = 1
  \}

  Else
  
  \{
    A^* = A \\
    R^* = \min(R/A, 1) \\
    G^* = \min(G/A, 1) \\
    B^* = \min(B/A, 1)
  \}
\}
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]:

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].

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.

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.)

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.

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.

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.
7. If a non-grouping element (a `<Path>` or `<Glyphs>` element) has an Opacity property, an OpacityMask property, or a fill or stroke using transparency, the following additional composition steps must be taken:

   a. If the element has a RenderTransform property, apply it to the element and its Clip, Fill, Stroke, and OpacityMask properties, if present.

   b. Create a mask from the set of all painted pixels representing the child element (after the Clip property of the element has been applied).

8. Combine the Fill or Stroke property with the OpacityMask and the Opacity property and apply to the surface through the computed mask. For more information, see §14.1.

The behavior that results from this process is:

- Opacity is not applied cumulatively to self-overlapping areas created when rendering an individual `<Glyphs>` element.
- Opacity is not applied cumulatively to self-overlapping areas created by `<PathFigure>` elements within the same path (see Example 18–1).
- Opacity is not applied cumulatively if the border of a path has self-intersections. When the border of a path is stroked, the area of the path is filled by first applying the brush specified by the Fill property. After filling the area, the border is drawn using the stroke-related properties including the brush specified by the Stroke property, with half the stroke width extending outside the filled area and half extending inside (see Example 18–2). If the path has self-intersections, the opacity is not accumulated.
- The color of the stroke and the color of the filled area are combined on the inside half of a stroked border (overlapping the filled area of the path) if the brush specified by the Stroke property is transparent.
- If a path that has a stroked border has an opacity of less than 1.0 or an opacity mask, the path (filled area and stroked border) is first rendered onto a temporary surface using an opacity of 1.0 and no opacity mask (while preserving any transparency of the fill or the stroked border themselves), and the resulting figure is drawn onto the background using the specified opacity and opacity mask (see Example 18–3).

18.5.1 Optimization Guidelines

The composition rules above describe the behavior of an ideal implementation. Practical implementations can optimize the processing of the composition rules according to the following guidelines:

1. 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 [O11.12].

2. If an element is fully transparent (an opacity of 0.0), it MAY be skipped [O11.13].

3. If a canvas has an opacity of 0.0, it and all of its child and descendant elements MAY be skipped [O11.14].

4. If a canvas has a Clip property with no contained area, the canvas and all of its child and descendant elements MAY be skipped [O11.15].
5. When creating a temporary surface, a consumer MAY further restrict the size of the temporary surface by the effective extent of the geometry specified by the Clip property of the canvas [O11.16].

6. A consumer MAY use methods to achieve transparency other than creating a temporary surface [O11.17]. Such methods MAY include planar mapping (that is, computation of intersections of transparent elements and resulting colors) [O11.17].

### 18.5.2 Composition Examples

The following examples illustrate the composition rules described above.

**Example 18–1. Path opacity behavior for overlapping path figures**

In the following markup, opacity is not applied cumulatively to self-overlapping areas created by path figures within the same path.

```xml
<Path Opacity="0.5">
  <Path.Fill>
    <SolidColorBrush Color="#0000FF" />
  </Path.Fill>
  <Path.Data>
    <PathGeometry FillRule="NonZero">
      <PathFigure StartPoint="10,10" IsClosed="true">
        <PolyLineSegment Points="110,10 110,110 10,110 10,10" />
      </PathFigure>
      <PathFigure StartPoint="50,50" IsClosed="true">
        <PolyLineSegment Points="150,50 150,150 50,150 50,50" />
      </PathFigure>
    </PathGeometry>
  </Path.Data>
</Path>
```

This markup is rendered as follows:

![Example 18–1](image)

**Example 18–2. Opacity behavior of path stroke intersections**

In the following markup, opacity is not applied cumulatively if the border of a path has self-intersections.

```xml
<Path Stroke="#80FF0000" StrokeThickness="10">
  <Path.Fill>
    <SolidColorBrush Color="#0000FF" />
  </Path.Fill>
</Path>
```
Example 18–3. Opacity behavior of paths with stroked edges

The following markup describes a path with a stroked border and an opacity of less than 1.0:

```xml
<Path>
  <Path.Fill>
    <SolidColorBrush Color="#7F7F7F" />
  </Path.Fill>
  <Path.Data>
    <PathGeometry>
      <PathFigure StartPoint="0,110" IsClosed="true">
        <PolyLineSegment Points="0,0 200,200 200,0 0,110"/>
      </PathFigure>
    </PathGeometry>
  </Path.Data>
</Path>
```
This markup is rendered as follows:
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.
18.6.4 Rules for Dash Cap Rendering

The appearance of dash caps is controlled by the StrokeDashCap attribute. Valid values are Flat, Square, Round, and Triangle.

18.6.4.1 Flat Dash Caps

The effective render transform of the path being stroked is used to transform the control points of the contour of the dash.

The length of the dash is the approximate distance on the curve between the two intersections of the flat lines ending the dash and the contour of the shape. The distance from the end of one dash to the start of the next dash is the specified dash gap length. Dashes with a length greater than 0 are drawn, and degenerate dashes with a length of 0 are not drawn.

Figure 18–2. Flat dash caps

18.6.4.2 Square Dash Caps

The effective render transform of the path being stroked is used to transform the control points of the contour of the dash.

The length of the dash is the approximate distance on the curve between the two contour intersection points, that is, the intersection of the flat line ending the dash (without the square caps attached) and the contour of the shape.

The caps are drawn as half-squares attached to the ends of the dash. The boundaries of the square caps are not curved to follow the contour, but are transformed using the effective render transform.

The distance between the contour intersection points of consecutive dashes is the specified dash gap length. Degenerate dashes with a length of 0 are drawn as squares.

Figure 18–3. Square dash caps
18.6.4.3 Round Dash Caps

The effective render transform of the path being stroked is used to transform the control points of the contour of the dash.

The length of the dash is the approximate distance on the curve between the two contour intersection points, that is, the intersection of the flat line ending the dash (without the round caps attached) and the contour of the shape.

The caps are drawn as half-circles attached to the ends of the dash. The boundaries of the round caps are not distorted to follow the contour, but are transformed using the effective render transform.

The distance between the contour intersection points of consecutive dashes is the specified dash gap length. Degenerate dashes with a length of 0 are drawn as circles.

Figure 18–4. Round dash caps

18.6.4.4 Triangular Dash Caps

The effective render transform of the path being stroked is used to transform the control points of the contour of the dash.

The length of the dash is the approximate distance on the curve between the two contour intersection points, that is, the intersection of the flat line ending the dash (without the triangular caps attached) and the contour of the shape.

The caps are drawn as triangles attached with their base to the ends of the dash. The boundaries of the triangular caps are not distorted to follow the contour, but are transformed using the effective render transform. The height of the triangles is half of the stroke width.

The distance between the contour intersection points of consecutive dashes is the specified dash gap length. Degenerate dashes with a length of 0 are drawn as diamonds.
18. Rendering Rules

18.6.4.5 Overlapping Dashes

It is possible to specify dash sequences with overlapping dash caps. In this circumstance, the union of the dash segments (inclusive of dash caps), is used as a mask through which the brush is applied as illustrated in Figure 18–6 with a stroke dash cap value of Round.
18.6.5 Rules for Line Cap Rendering

The appearance of line caps is controlled by the StrokeStartLineCap and StrokeEndLineCap attribute. Valid values are Flat, Square, Triangle, and Round. Every start line cap can be used in combination with any end line cap. Line caps are not drawn for closed paths.

The rules for line caps on curved lines are analogous to the rules for dash cap rendering. For more information, see §18.6 and §18.6.4.

Figure 18–7. Flat start line cap, flat end line cap

Figure 18–8. Square start line cap, square end line cap

Figure 18–9. Triangular start line cap, triangular end line cap

Figure 18–10. Round start line cap, round end line cap

18.6.6 Line Caps for Dashed Strokes

If the start point of a stroke is within a dash or touches the start or end of a dash, a start line cap is appended to the stroke. Similarly, if the end point of a stroke is within a dash or touches the start or end of a dash, an end line cap is appended to the stroke.

Figure 18–11. Stroke start or end point within a dash

[Note: Because the right-most line cap begins at the point exactly coincident with the start of the next dash in the sequence, it is rendered. end note]

However, if the start point of a stroke is within a gap (as can result from a StrokeDashOffset attribute), no start line cap is appended to the stroke. If the end point of a stroke is within a gap, no end line cap is appended to the stroke.
18. Rendering Rules

18.6.7 Rules for Line Join Rendering

The appearance of line joins is controlled by the StrokeLineJoin attribute. Valid values are Round, Bevel, and Miter.

18.6.7.1 Round Line Joins

A StrokeLineJoin attribute value of Round indicates that the outer corner of the joined lines should be filled by enclosing the rounded region with its center point at the point of intersection between the two lines and a radius of one-half the stroke thickness value.

Figure 18–13. Round line join with right angle

Figure 18–14. Round line join with acute angle
18.6.7.2 Beveled Line Joins

A StrokeLineJoin attribute value of Bevel indicates that the outer corner of the joined lines should be filled by enclosing the triangular region of the corner with a straight line between the outer corners of each stroke.

Figure 18–15. Round line join with obtuse angle

Figure 18–16. Beveled line join with right angle

Figure 18–17. Beveled line join with acute angle
18. Rendering Rules

Figure 18–18. Beveled line join with obtuse angle

Figure 18–19. Mitered line join with right angle and miter limit of 1.0

18.6.7.3 Mitered Line Joins

If the StrokeLineJoin attribute value is Miter, the value of the StrokeMiterLimit attribute value is also used for rendering these joins. A StrokeLineJoin value of Miter indicates that the region to be filled includes the intersection of the strokes projected to infinity, and then clipped at a specific distance. The intersection of the strokes is clipped at a line perpendicular to the bisector of the angle between the strokes, at the distance equal to the stroke miter limit value multiplied by half the stroke thickness value.

When drawing mitered line joins, the presence of one or more degenerate line segments between the non-degenerate line segments to be joined results in a mitered line join of only the two non-degenerate line segments with an implied StrokeMiterLimit attribute value of 1.0.
Figure 18–20. Mitered line join with acute angle and miter limit of 1.0

Figure 18–21. Mitered line join with obtuse angle and miter limit of 1.0

Figure 18–22. Mitered line join with right angle and miter limit of 2.0
18.6.8 Rules for Degenerate Line and Curve Segments

Degenerate line segments (that is, where the start point and end point coincide) are not drawn.

Degenerate curve segments (where the start point, end point, and all control points coincide) are not drawn, regardless of the stroke start line cap or stroke end line cap specified.

If an open degenerate path (formed from degenerate line or curve segments) with non-flat start cap and/or non-flat end line cap is stroked, only the start line cap and/or end line cap is drawn, in the x direction relative to the current effective render transform (that is, as if a segment were drawn from \(x,y\) to \(x+d,y\), with \(d \rightarrow 0\)).

If a closed degenerate path (formed from degenerate line or curve segments) is stroked, a circular dot with a diameter of the stroke thickness is drawn instead.

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 [S11.25].
18.6.9 Stroking and Fill Rule
Stroking a path is independent of the fill rule. The fill rule affects the filled area only.

18.6.10 Mixing Stroked and Non-Stroked Segments
When a path figure contains multiple segments and one or more of the segments has an IsStroked value of false, the phase for dashes starts anew with the next stroked segment, including application of the dash offset.

When a segment is stroked and the subsequent segment has an IsStroked value of false, thus causing a dash to be truncated, the dash cap is drawn for both ends of the truncated dash, exactly as it would for a non-truncated dash.

18.6.11 Stroke Behavior with Multiple Path Figures
When a geometry containing multiple path figures is stroked, the phase for dashes (including application of the dash offsets) starts anew with each new path figure.

In general, for any path geometry, each path figure is drawn independently of every other path figure, so the dash array is reset for each. Dashes are also reset after every unstroked segment.

18.6.12 Consistent Nominal Stroke Width
For certain scenarios, it is desirable for producers to generate documents targeted at specific aliasing consumers with particular lines in the document indicated as hairlines or consistent-width strokes. The following recommendation allows these producers and consumers to handle these strokes consistently.

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 a StrokeDashOffset attribute value less than -1.0 times the sum of all the numbers in the StrokeDashArray attribute value [O11.25].

For a solid line, the producer would set the StrokeDashArray to the value "1 0" and the StrokeDashOffset to a value such as "-2". The "-2" value fulfills the restriction on the StrokeDashOffset value in a numerically stable manner, and the phase of the dash pattern is identical to a StrokeDashOffset value of "0". Values less than "-2" can be used to specify a shifted phase of the dash pattern.

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 [S11.31]. See section §11.1.4, for further considerations for rendering thin lines.

18.7 Brushes and Images
Images require the following special considerations for scaling and tile placement.
18. Rendering Rules

18.7.1 Small Tiles
Tiles for visual brushes and image brushes can be specified with a viewport width or height of a few device pixels, or even less than a single device pixel in size.

If both width and height are nearly zero, implementations SHOULD average the color values of the brush contents, resulting in a constant-color brush [S11.26]. [Example:

- A visual brush or image brush that contains a blue and white checkerboard pattern results in a solid light-blue fill as either the width or the height value approaches 0.0.
- A visual brush or image brush whose viewbox is constant-colored produces a constant-colored brush regardless of the width and height values of the viewport.

dend example]

If only one of the width and height values is nearly zero, the brush should be constant-colored along lines parallel to the narrow side of the viewport. For cases such as these, implementations MAY differ [O11.21]. Producers SHOULD avoid producing such extreme cases and SHOULD NOT rely on any specific behavior when they do [S11.27].

18.7.2 Image Scaling
Source sampling SHOULD be done from the center of the pixel and should be mapped to the center of the pixel in the device-space [S11.28]. With one extent of the viewbox zero, sampling SHOULD be done along a line parallel to the non-zero side [S11.28]. With both extents of the viewbox zero, a point sample SHOULD be taken [S11.28].

When up-sampling an image presented at a lower resolution than the device resolution, bilinear filtering SHOULD be used [S11.29]. The precise source coordinates as specified by the viewbox MUST be used to place the up-sampled image tile, which is equivalent to using fractional pixels of the original source image [M11.8].

When down-sampling, at least a bilinear filter SHOULD be used [S11.30]. Consumers MAY choose to implement a more sophisticated algorithm, such as a Fant scaler, to prevent aliasing artifacts [O11.22].

18.7.3 Tile Placement
Consumers MUST precisely position the tiles specified by the image brush and visual brush. If the specified values result in fractional device pixels, the consumer MUST calculate a running placement-error delta and adjust the placement of the next tile where the delta reaches a full device pixel in order to keep the tiles from being increasingly out of phase as the expanse of the path is filled [M11.9]. Consumers MAY choose any technique desired to achieve this requirement, such as linear filtering for seams, stretching of the tile (up-sampling or down-sampling), or pre-computing multiple tiles and adjusting behavior according to how the tiles fit on a grid [O11.23].

18.7.4 Tiling Transparent Visual Brushes and Image Brushes
The contents of a visual brush’s Visual property are first rendered to a temporary work canvas with an opacity of 0.0. The viewbox of the visual brush defines the tile or portion of the temporary canvas that is copied onto the specified geometry, stroke, or text. Likewise, an image specified by an image brush is also copied to a temporary work canvas. The viewbox also defines the tile for an image brush. In either case, the work canvas is scaled to properly match the edges of the tile to the size specified by the viewport.
Each pixel of the resultant tile is separately blended with the background of its destination, using the alpha of each pixel. This process is repeated for each tile replication, while respecting the TileMode attribute value, although the temporary work canvas MAY be re-used [O11.24].
19. Elements

19.1 ArcSegment

element ArcSegment

<table>
<thead>
<tr>
<th>attributes</th>
<th>Name</th>
<th>Type</th>
<th>Use</th>
<th>Default</th>
<th>Fixed</th>
<th>Annotation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<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></td>
<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></td>
<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></td>
<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></td>
<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></td>
<td>IsStroked</td>
<td>ST_Boolean</td>
<td>true</td>
<td></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>

annotation

Represents an elliptical arc between two points.
19.2 Canvas

The `Canvas` element is used to create a canvas element. A canvas is a declarative surface that can contain various child elements including other canvases, paths, and text. The `Canvas` element has several attributes that control its appearance and behavior:

<table>
<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>RenderTransform</td>
<td>ST_RscRefMatrix</td>
<td></td>
<td></td>
<td></td>
<td>Establishes a new coordinate frame for the child and descendant elements of the canvas, such as another canvas. Also affects clip and opacity mask.</td>
</tr>
<tr>
<td>Clip</td>
<td>ST_RscRefAbbrGeomF</td>
<td></td>
<td></td>
<td></td>
<td>Limits the rendered region of the element.</td>
</tr>
<tr>
<td>Opacity</td>
<td>ST_ZeroOne</td>
<td></td>
<td>1.0</td>
<td></td>
<td>Defines the uniform transparency of the canvas. Values range from 0 (fully</td>
</tr>
</tbody>
</table>

The diagram shows the structure of the `Canvas` element, including its attributes and relationships with other elements such as paths and text.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpacityMask</td>
<td>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.</td>
</tr>
<tr>
<td>Name</td>
<td>Contains a string value that identifies the current element as a named, addressable point in the document for the purpose of hyperlinking.</td>
</tr>
<tr>
<td>RenderOptions.EdgeMode</td>
<td>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.</td>
</tr>
<tr>
<td>FixedPage.NavigateUri</td>
<td>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.</td>
</tr>
<tr>
<td>xml:lang</td>
<td>Specifies the default language used for the current element and for any child or descendant elements. The language is specified according to RFC 3066.</td>
</tr>
<tr>
<td>x:Key</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 [M3.20].</td>
</tr>
<tr>
<td>AutomationProperties.Name</td>
<td>A brief description of the &lt;Canvas&gt; contents for</td>
</tr>
</tbody>
</table>
19. Elements

For more information, see §10.4.

19.3 Canvas.Clip

element Canvas.Clip

diagram

annotation

Limits the rendered region of the element.

For more information, see §14.3.

19.4 Canvas.OpacityMask

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.
For more information, see §14.5.1.

19.5 Canvas.RenderTransform
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.

For more information, see §14.4.

19.6 Canvas.Resources
element Canvas.Resources

diagram

annotation
Contains the resource dictionary for the <Canvas> element.

For more information, see §14.2.

19.7 Discard
element Discard

attributes

<table>
<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>
19. Elements

| annotation | Identifies a resource that can be safely discarded by a resource-constrained consumer. |

For more information, see §17.1.4.1.2.

19.8 DiscardControl

element DiscardControl

<table>
<thead>
<tr>
<th>diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT_DiscardControl</td>
</tr>
<tr>
<td>DiscardControl</td>
</tr>
<tr>
<td>Discard</td>
</tr>
</tbody>
</table>

| annotation | Contains a list of resources that are safe for a consumer to discard. |

For more information, see §17.1.4.1.1.

19.9 DocumentOutline

element DocumentOutline

<table>
<thead>
<tr>
<th>diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT_DocumentOutline</td>
</tr>
<tr>
<td>DocumentOutline</td>
</tr>
<tr>
<td>xml:lang</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
</tr>
<tr>
<td>xml:lang</td>
</tr>
</tbody>
</table>

| 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. |

For more information, see §16.1.1.3.

19.10 DocumentReference

element DocumentReference
### 19.11 DocumentStructure

**element** `DocumentStructure`

- **Diagram:**
  - `DocumentStructure` diagram showing the structure.

- **Annotation:**

For more information, see §16.1.1.1.

### 19.12 DocumentStructure.Outline

**element** `DocumentStructure.Outline`

- **Diagram:**
  - `DocumentStructure.Outline` diagram showing the structure.

- **Annotation:**
  - Contains a structured document outline that provides a list of links into the document contents or external sites.

For more information see §16.1.1.2.
19.13 **FigureStructure**

element `FigureStructure`

Groups the named elements that constitute a single drawing or diagram.

For more information, see §16.1.2.12.

19.14 **FixedDocument**

element `FixedDocument`

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

For more information, see §10.2.

19.15 **FixedDocumentSequence**

element `FixedDocumentSequence`

Specifies a sequence of fixed documents.

For more information, see §10.1.

19.16 **FixedPage**

element `FixedPage`
### 19. Elements

#### CT_FixedPage

- **attributes**
  - **Width**
    - Type: ST_GEOOne
    - Use: required
    - Default: null
    - Description: Width of the page, expressed as a real number in units of the effective coordinate space.
  - **Height**
    - Type: ST_GEOOne
    - Use: required
    - Default: null
    - Description: Height of the page, expressed as a real number in units of the effective coordinate space.
  - **ContentBox**
    - Type: ST_ContentBox
    - Description: 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**
    - Type: ST_BleedBox
    - Description: 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**
    - Type: required
    - Description: 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**
    - Type: ST_Name
    - Description: Contains a string value that identifies the current

---

Working Draft 1.1, February 2008
19. Elements

19.17 FixedPage.Resources

For more information, see §10.3.

19.18 Glyphs

For more information, see §14.2.
Table of attributes:

<table>
<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>BidiLevel</td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
<td>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...</td>
</tr>
<tr>
<td>Element</td>
<td>Type</td>
<td>Description</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------------</td>
<td>-------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CaretStops</td>
<td>ST_CaretStops</td>
<td>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].</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DeviceFontName</td>
<td>ST_UnicodeString</td>
<td>Uniquely identifies a specific device font. The identifier is typically defined by a hardware vendor or font vendor.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fill</td>
<td>ST_RscRefColor</td>
<td>Describes the brush used to fill the shape of the rendered glyphs.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FontRenderingEmSize</td>
<td>ST_GEZero</td>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FontUri</td>
<td>xs:anyURI</td>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OriginX</td>
<td>ST_Double</td>
<td>required</td>
<td>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.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------</td>
<td>----------------</td>
<td>--------------</td>
<td>--------------------------------------------------------------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OriginY</td>
<td>ST_Double</td>
<td>required</td>
<td>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.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IsSideways</td>
<td>ST_Boolean</td>
<td>false</td>
<td>Indicates that a glyph is turned on its side, with the origin being defined as the top center of the unturned glyph.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Indices</td>
<td>ST_Indices</td>
<td></td>
<td>Specifies a series of glyph indices and their attributes used for rendering the glyph run. If the UnicodeString attribute specifies an empty string (&quot;&quot; or &quot;{}&quot;) and the Indices attribute is not specified or is also empty, a consumer MUST generate an error [M5.2].</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UnicodeString</td>
<td>ST_UnicodeString</td>
<td></td>
<td>Contains the string of text rendered by the &lt;Glyphs&gt; element. The text is specified as Unicode code points.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>StyleSimulations</td>
<td>ST_StyleSimulations</td>
<td>None</td>
<td>Specifies a style simulation. Valid values are None, ItalicSimulation, BoldSimulation, and BoldItalicSimulation.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RenderTransform</td>
<td>ST_RscRefMatrix</td>
<td></td>
<td>Establishes a new coordinate frame for the glyph run specified by the &lt;Glyphs&gt; element. The render transform affects clip, opacity mask, fill, x origin, y origin, the actual shape of individual glyphs, and the advance widths. The render transform also affects the font size and values specified in the</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Indices attribute.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clip</td>
<td><code>ST_RscRefAbbrGeomF</code></td>
<td>Limits the rendered region of the element. Only portions of the <code>&lt;Glyphs&gt;</code> element that fall within the clip region (even partially clipped characters) produce marks on the page.</td>
</tr>
<tr>
<td>Opacity</td>
<td><code>ST_ZeroOne</code></td>
<td>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.</td>
</tr>
<tr>
<td>OpacityMask</td>
<td><code>ST_RscRef</code></td>
<td>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.</td>
</tr>
<tr>
<td>Name</td>
<td><code>ST_Name</code></td>
<td>Contains a string value that identifies the current element as a named, addressable point in the document for the purpose of hyperlinking.</td>
</tr>
<tr>
<td>FixedPage.NavigateUri</td>
<td><code>xs:anyURI</code></td>
<td>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.</td>
</tr>
<tr>
<td>xml:lang</td>
<td></td>
<td>Specifies the default language used for the current element. The language is specified according to RFC 3066.</td>
</tr>
<tr>
<td>x:Key</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 [M5.3].</td>
</tr>
</tbody>
</table>

Annotation: Represents a run of text from a single font.

For more information, see Section §12.1, Section §9.1.7, and Section §12.1.3.
19.19 Glyphs.Clip

element Glyphs.Clip

Diagram: CT_CP_Geometry

Annotation: Limits the rendered region of the element. Only portions of the <Glyphs> element that fall within the clip region (even partially clipped characters) produce marks on the page.

For more information, see §14.3.

19.20 Glyphs.Fill

element Glyphs.Fill

Diagram: CT_CP_Brush

Annotation: Describes the brush used to fill the shape of the rendered glyphs.

For more information, see §12.2.

19.21 Glyphs.OpacityMask

element Glyphs.OpacityMask

Diagram: CT_CP_Brush

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.

For more information, see §14.5.3.

## 19.22 Glyphs.RenderTransform

### element Glyphs.RenderTransform

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.

For more information, see §14.4.

## 19.23 GradientStop

### element GradientStop

Indicates a location and range of color progression for rendering a gradient.

For more information, see §13.7.
19.24 ImageBrush

Element ImageBrush

<table>
<thead>
<tr>
<th>attributes</th>
<th>Name</th>
<th>Type</th>
<th>Use</th>
<th>Default</th>
<th>Fixed</th>
<th>Annotation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Opacity</td>
<td>ST_ZeroOne</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></td>
<td>x:Key</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Specifies a name for a resource in a resource dictionary. x:Key MUST be present when the current element is defined in a resource dictionary. x:Key MUST NOT be specified outside of a resource dictionary [M6.2].</td>
</tr>
<tr>
<td></td>
<td>Transform</td>
<td>ST_RscRefMatrix</td>
<td></td>
<td></td>
<td></td>
<td>Describes the matrix transformation applied to the coordinate space of the brush. The Transform property is concatenated with the current effective render transform to yield an effective render transform local to the brush. The viewport for the brush is transformed using the local effective render transform.</td>
</tr>
<tr>
<td></td>
<td>Viewbox</td>
<td>ST_ViewBox</td>
<td>required</td>
<td></td>
<td></td>
<td>Specifies the position and dimensions of the brush's source content. Specifies four comma-separated real numbers ((x, y, \text{Width}, \text{Height})), where width and height are non-negative. The dimensions specified are relative to the image's physical dimensions expressed in</td>
</tr>
<tr>
<td>Element</td>
<td>Type</td>
<td>Required</td>
<td>Description</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>------</td>
<td>----------</td>
<td>-------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Viewport</td>
<td>ST_ViewBox</td>
<td>required</td>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TileMode</td>
<td>ST_TileMode</td>
<td>None</td>
<td>Specifies how contents will be tiled in the filled region. Valid values are None, Tile, FlipX, FlipY, and FlipXY.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ViewboxUnits</td>
<td>ST_ViewUnits</td>
<td>required</td>
<td>Absolute</td>
<td>Specifies the relationship of the viewbox coordinates to the containing coordinate space.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ViewportUnits</td>
<td>ST_ViewUnits</td>
<td>required</td>
<td>Absolute</td>
<td>Specifies the relationship of the viewport coordinates to the containing coordinate space.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ImageSource</td>
<td>ST_UriCtxBmp</td>
<td>required</td>
<td></td>
<td>Specifies the URI of an image resource or a combination of the URI of an image resource and a color profile resource. See the Color clause for important details. The URI MUST refer to parts in the package [M2.1].</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fills a region with an image.

For more information, see §13.2.

19.25 **ImageBrush.Transform**

Element **ImageBrush.Transform**

*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.*
For more information, see §14.4.

19.26 Intent

element SignatureDefinitionType/Intent

A string that represents the intent to which the signing party agrees when signing the document.

For more information, see §17.2.2.4.

19.27 LinearGradientBrush

element LinearGradientBrush

- **Attributes**
  - **Name**: Opacity
    - **Type**: ST_ZeroOne
    - **Use**: 1.0
    - **Default**: 1.0
    - **Fixed**: FALSE
    - **Annotation**: 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.
  - **Name**: x:Key
    - **Annotation**: Specifies a name for a resource in a resource dictionary. x:Key MUST be present when the
<table>
<thead>
<tr>
<th>Element</th>
<th>Type</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ColorInterpolationMode</td>
<td>ST_ClrIntMode</td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>SpreadMethod</td>
<td>ST_SpreadMethod</td>
<td></td>
<td>Describes how the brush should fill the content area outside of the primary, initial gradient area. Valid values are Pad, Reflect and Repeat.</td>
</tr>
<tr>
<td>MappingMode</td>
<td>ST_MappingMode</td>
<td>required</td>
<td>Specifies that the start point and end point are defined in the effective coordinate space (includes the Transform attribute of the brush).</td>
</tr>
<tr>
<td>Transform</td>
<td>ST_RscRefMatrix</td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>StartPoint</td>
<td>ST_Point</td>
<td>required</td>
<td>Specifies the starting point of the linear gradient.</td>
</tr>
<tr>
<td>EndPoint</td>
<td>ST_Point</td>
<td>required</td>
<td>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.</td>
</tr>
</tbody>
</table>
annotation

Fills a region with a linear gradient.

For more information, see Section §13.5 and §15.

---

19.28 LinearGradientBrush.GradientStops

For more information, see §13.5.2.

---

19.29 LinearGradientBrush.Transform

For more information, see §14.4.8.

---

19.30 LinkTarget

Contains a string value that identifies the current element as a named, addressable point in the document for the purpose of


diagram

annotation

description

name
For more information, see §10.2.3.

### 19.31 ListItemStructure

**element ListItemStructure**

<table>
<thead>
<tr>
<th>attributes</th>
<th>Name</th>
<th>Type</th>
<th>Use</th>
<th>Default</th>
<th>Fixed</th>
<th>Annotation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Marker</td>
<td>ST_NameUnique</td>
<td>optional</td>
<td></td>
<td></td>
<td></td>
<td>The named element that represents the marker for this list items, such as a bullet, number, or image.</td>
</tr>
</tbody>
</table>

For more information, see §16.1.2.11.

### 19.32 ListStructure

**element ListStructure**

| annotation | Contains a collection of items that are group together in a list. |
For more information, see §16.1.2.10.

### 19.33 MatrixTransform

**element** MatrixTransform

<table>
<thead>
<tr>
<th>attribute</th>
<th>Name</th>
<th>Type</th>
<th>Use</th>
<th>Default</th>
<th>Fixed</th>
<th>Annotation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Matrix</td>
<td>ST_Matrix</td>
<td>required</td>
<td></td>
<td></td>
<td></td>
<td>Specifies the matrix structure that defines the transformation.</td>
</tr>
<tr>
<td>x:Key</td>
<td></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 [M7.11].</td>
</tr>
</tbody>
</table>

**annotation**

Creates an arbitrary affine matrix transformation that manipulates objects or coordinate systems in a two-dimensional plane.

For more information, see §14.4.1.

### 19.34 NamedElement

**element** NamedElement

<table>
<thead>
<tr>
<th>attribute</th>
<th>Name</th>
<th>Type</th>
<th>Use</th>
<th>Default</th>
<th>Fixed</th>
<th>Annotation</th>
</tr>
</thead>
<tbody>
<tr>
<td>NameReference</td>
<td>ST_Name</td>
<td>required</td>
<td></td>
<td></td>
<td></td>
<td>Identifies the named element in the FixedPage part markup that is referenced by the document structure markup.</td>
</tr>
</tbody>
</table>

**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.
For more information, see §16.1.2.13.

19.35 OutlineEntry

<table>
<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>

For more information, see §16.1.1.4.

19.36 PageContent

For more information, see §16.1.2.13.
## 19.37 PageContent.LinkTargets

element **PageContent.LinkTargets**

For more information, see §10.2.2.

## 19.38 ParagraphStructure

element **ParagraphStructure**

For more information, see §10.2.2.
For more information, see §16.1.2.5.

19.39 Path

element Path
<table>
<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>Data</td>
<td>ST_RscRefAbbrGeomF</td>
<td></td>
<td></td>
<td></td>
<td>Describes the geometry of the path.</td>
</tr>
<tr>
<td>Element</td>
<td>Type</td>
<td>Description</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------------</td>
<td>---------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fill</td>
<td>ST_RscRefColor</td>
<td>Describes the brush used to paint the geometry specified by the Data property of the path.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RenderTransform</td>
<td>ST_RscRefMatrix</td>
<td>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 &lt;Path.Data&gt; property element.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Clip</td>
<td>ST_RscRefAbbrGeomF</td>
<td>Limits the rendered region of the element.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Opacity</td>
<td>ST_ZeroOne 1.0</td>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OpacityMask</td>
<td>ST_RscRef</td>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stroke</td>
<td>ST_RscRefColor</td>
<td>Specifies the brush used to draw the stroke.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>StrokeDashArray</td>
<td>ST_EvenArrayPos</td>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>StrokeDashCap</td>
<td>ST_DashCap Flat</td>
<td>Specifies how the ends of each dash are drawn. Valid values are Flat, Round, Square, and</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Value</td>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------------------</td>
<td>------------</td>
<td>-------</td>
<td>-------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>StrokeDashOffset</td>
<td>ST_Double</td>
<td>0.0</td>
<td>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.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>StrokeEndLineCap</td>
<td>ST_LineCap</td>
<td>Flat</td>
<td>Defines the shape of the end of the last dash in a stroke. Valid values are Flat, Square, Round, and Triangle.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>StrokeStartLineCap</td>
<td>ST_LineCap</td>
<td>Flat</td>
<td>Defines the shape of the beginning of the first dash in a stroke. Valid values are Flat, Square, Round, and Triangle.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>StrokeLineJoin</td>
<td>ST_LineJoin</td>
<td>Miter</td>
<td>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.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>StrokeMiterLimit</td>
<td>ST_GEOOne</td>
<td>10.0</td>
<td>The ratio between the maximum miter length and half of the stroke thickness. This value is significant only if the StrokeLineJoin attribute specifies Miter.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>StrokeThickness</td>
<td>ST_GEZero</td>
<td>1.0</td>
<td>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 &lt;Path&gt; 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.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>ST_Name</td>
<td></td>
<td>Contains a string value that identifies the current element</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### 19. Elements

<table>
<thead>
<tr>
<th>Element</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FixedPage.NavigateUri</td>
<td>xs:anyURI</td>
<td>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.</td>
</tr>
<tr>
<td>xml:lang</td>
<td></td>
<td>Specifies the default language used for the current element and for any child or descendant elements. The language is specified according to RFC 3066.</td>
</tr>
<tr>
<td>x:Key</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 [M4.1].</td>
</tr>
<tr>
<td>AutomationProperties.Name</td>
<td>xs:string</td>
<td>A brief description of the &lt;Path&gt; for accessibility purposes, particularly if filled with an &lt;ImageBrush&gt;.</td>
</tr>
<tr>
<td>AutomationProperties.HelpText</td>
<td>xs:string</td>
<td>A detailed description of the &lt;Path&gt; for accessibility purposes, particularly if filled with an &lt;ImageBrush&gt;.</td>
</tr>
<tr>
<td>SnapsToDevicePixels</td>
<td>ST_Boolean</td>
<td>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].</td>
</tr>
<tr>
<td><strong>annotation</strong></td>
<td></td>
<td>Defines a single graphical effect to be rendered to the page. It paints a geometry with a brush and draws a stroke around it.</td>
</tr>
</tbody>
</table>

For more information, see Section §11.1 and Section §11.2.3.
19.40 Path.Clip

element Path.Clip

annotation Limits the rendered region of the element.

For more information, see §14.3.

19.41 Path.Data

element Path.Data

annotation Describes the geometry of the path.

For more information, see §11.1.1.

19.42 Path.Fill

element Path.Fill

annotation Describes the brush used to paint the geometry specified by the Data property of the path.

For more information, see §11.1.2.

19.43 Path.OpacityMask

element Path.OpacityMask
19. Elements

For more information, see §14.5.2.

19.44 Path.RenderTransform

element Path.RenderTransform

For more information, see §14.4.

19.45 Path.Stroke

element Path.Stroke

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.

For more information, see §14.5.2.

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.

Specifies the brush used to draw the stroke.
For more information, see §11.1.3.

### 19.46 PathFigure

**element PathFigure**

<table>
<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>IsClosed</td>
<td>ST_Boolean</td>
<td>false</td>
<td></td>
<td></td>
<td>Specifies whether the path is closed. If set to true, the stroke is drawn &quot;closed,&quot; 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 &quot;open,&quot; and the last point is not connected to the start point. Only applicable if the path figure is used in a &lt;Path&gt; element that specifies a stroke.</td>
</tr>
<tr>
<td>StartPoint</td>
<td>ST_Point</td>
<td>required</td>
<td></td>
<td></td>
<td>Specifies the starting point for the first segment of the path figure.</td>
</tr>
<tr>
<td>IsFilled</td>
<td>ST_Boolean</td>
<td>true</td>
<td></td>
<td></td>
<td>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.</td>
</tr>
</tbody>
</table>

*annotation* Specifies a set of one or more segment elements defining a closed region.

For more information, see §11.2.2.1.

### 19.47 PathGeometry

**element PathGeometry**
19. Elements

<table>
<thead>
<tr>
<th>attributes</th>
<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>ST_AbbrGeom</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Describes the geometry of the path.</td>
</tr>
<tr>
<td>FillRule</td>
<td>ST_FillRule</td>
<td>EvenOdd</td>
<td></td>
<td></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>ST_RscRefMatrix</td>
<td></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></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>

For more information, see §11.2.1.1.

19.48 PathGeometry.Transform

element PathGeometry.Transform

For more information, see §11.2.1.1.
For more information, see §14.4.5.

### 19.49 PolyBezierSegment

The **PolyBezierSegment** element is used to specify a series of Bézier segments. Each segment is defined by a set of control points, and the stroke for the segment can be specified.

#### Attributes

<table>
<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>true</td>
<td></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>

#### Annotation

A series of Bézier segments.

### 19.50 PolyLineSegment

The **PolyLineSegment** element is used to specify a series of line segments. Each segment is defined by a set of coordinates, and the stroke for the segment can be specified.

#### Attributes

<table>
<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>

#### Annotation

A series of line segments.
### 19. Elements

#### 19.51 PolyQuadraticBezierSegment

element **PolyQuadraticBezierSegment**

<table>
<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 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.</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>

annotation: A series of quadratic Bézier segments.

---

#### 19.52 RadialGradientBrush

element **RadialGradientBrush**
### Attributes

<table>
<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>ST_ZeroOne</td>
<td>Use</td>
<td>1.0</td>
<td></td>
<td>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.</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 [M6.6].</td>
</tr>
<tr>
<td>ColorInterpolationMode</td>
<td>ST_ClrIntMode</td>
<td>Use</td>
<td>SRgbLinear Interpolation</td>
<td></td>
<td>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 ScRrgbLinearInterpolation.</td>
</tr>
<tr>
<td>SpreadMethod</td>
<td>ST_SpreadMethod</td>
<td>Use</td>
<td>Pad</td>
<td></td>
<td>Describes how the brush should fill the content area outside of</td>
</tr>
</tbody>
</table>
### 19. Elements

<table>
<thead>
<tr>
<th></th>
<th>ST_ Mapping Mode</th>
<th>required</th>
<th>Absolute</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MappingMode</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>ST_RscRef Matrix</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Transform</strong></td>
<td></td>
<td>ST_Point</td>
<td>required</td>
</tr>
<tr>
<td><strong>Center</strong></td>
<td>ST_Point</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>GradientOrigin</strong></td>
<td>ST_Point</td>
<td>required</td>
<td></td>
</tr>
<tr>
<td><strong>RadiusX</strong></td>
<td>ST_GEZero</td>
<td>required</td>
<td></td>
</tr>
<tr>
<td><strong>RadiusY</strong></td>
<td>ST_GEZero</td>
<td>required</td>
<td></td>
</tr>
</tbody>
</table>

**MappingMode**

Valid values are Pad, Reflect and Repeat.

**ST_Mapping Mode**

Specifies that center, x radius, and y radius are defined in the effective coordinate space (includes the Transform attribute of the brush).

**ST_RscRef Matrix**

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.

**ST_Point**

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.

**ST_GEZero**

Specifies the radius in the x dimension of the ellipse which defines the radial gradient.

**ST_GEZero**

Specifies the radius in the y dimension of the ellipse which defines the radial gradient.

---

For more information, see Section §13.6 and §15.
19.53  **RadialGradientBrush.GradientStops**

<table>
<thead>
<tr>
<th>element</th>
<th>RadialGradientBrush.GradientStops</th>
</tr>
</thead>
</table>

**Diagram**

**Annotation**

Holds a sequence of `<GradientStop>` elements.

For more information, see §13.6.2, “<RadialGradientBrush.GradientStops> Element,” on page 165.

19.54  **RadialGradientBrush.Transform**

<table>
<thead>
<tr>
<th>element</th>
<th>RadialGradientBrush.Transform</th>
</tr>
</thead>
</table>

**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.

For more information, see §14.4.9.

19.55  **ResourceDictionary**

<table>
<thead>
<tr>
<th>element</th>
<th>ResourceDictionary</th>
</tr>
</thead>
</table>
For more information, see §14.2.

**19.56 SectionStructure**

**element SectionStructure**

For more information, see §16.1.2.4.
19.57 **SignBy**

**element** `SignatureDefinitionType/SignBy`

**diagram**

<table>
<thead>
<tr>
<th><strong>attributes</strong></th>
<th><strong>Name</strong></th>
<th><strong>Type</strong></th>
<th><strong>Use</strong></th>
<th><strong>Default</strong></th>
<th><strong>Fixed</strong></th>
<th><strong>Annotation</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>SpotID</td>
<td>xs:ID</td>
<td>required</td>
<td></td>
<td></td>
<td></td>
<td>A globally unique identifier for this signature spot.</td>
</tr>
<tr>
<td>SignerName</td>
<td>xs:string</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>xml:lang</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Specifies the language used for the current element and its descendants. The language is specified according to RFC 3066.</td>
</tr>
</tbody>
</table>

**annotation** A single signature definition.

For more information, see §17.2.2.2.

19.58 **SignatureDefinition**

**element** `SignatureDefinitionsType/SignatureDefinition`

**diagram**

**attributes**

<table>
<thead>
<tr>
<th><strong>Name</strong></th>
<th><strong>Type</strong></th>
<th><strong>Use</strong></th>
<th><strong>Default</strong></th>
<th><strong>Fixed</strong></th>
<th><strong>Annotation</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>SpotID</td>
<td>xs:ID</td>
<td></td>
<td></td>
<td></td>
<td>A globally unique identifier for this signature spot.</td>
</tr>
<tr>
<td>SignerName</td>
<td>xs:string</td>
<td></td>
<td></td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>xml:lang</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Specifies the language used for the current element and its descendants. The language is specified according to RFC 3066.</td>
</tr>
</tbody>
</table>

**annotation** A single signature definition.

For more information, see §17.2.2.5.
19.59  SignatureDefinitions

Element SignatureDefinitions

![Diagram of SignatureDefinitions]

The root element for the SignatureDefinitions part.

For more information, see §17.2.2.1.

19.60  SigningLocation

Element SignatureDefinitionType/SigningLocation

![Diagram of SigningLocation]

The legal location where the document is signed.

For more information, see §17.2.2.6.

19.61  SolidColorBrush

Element SolidColorBrush

![Diagram of SolidColorBrush]

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Name</th>
<th>Type</th>
<th>Use</th>
<th>Default</th>
<th>Fixed</th>
<th>Annotation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Opacity</td>
<td>ST_ZeroOne</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></td>
<td>x:Key</td>
<td>ST_ZeroOne</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>
Color  ST_Color  required

Specifies the color for filled elements. An sRGB color value specified as a 6-digit hexadecimal number (#RRGGBB) or an extended color.

**Annotation**
Fills defined geometric regions with a solid color.

For more information, see §13.1.

### 19.62 SpotLocation

**Element** `SignatureDefinitionType/SpotLocation`

diagram

<table>
<thead>
<tr>
<th>attributes</th>
<th>Name</th>
<th>Type</th>
<th>Use</th>
<th>Default</th>
<th>Fixed</th>
<th>Annotation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PageURI</td>
<td>xs:anyURI</td>
<td>required</td>
<td></td>
<td></td>
<td>Specifies the page on which the signature spot should be displayed.</td>
</tr>
<tr>
<td></td>
<td>StartX</td>
<td>xs:double</td>
<td>required</td>
<td></td>
<td></td>
<td>Specifies the x coordinate of the origin point (upper-left corner) on the page where the signature spot should be displayed.</td>
</tr>
<tr>
<td></td>
<td>StartY</td>
<td>xs:double</td>
<td>required</td>
<td></td>
<td></td>
<td>Specifies the y coordinate of the origin point (upper-left corner) on the page where the signature spot should be displayed.</td>
</tr>
</tbody>
</table>

For more information, see §17.2.2.3.

### 19.63 Story

**Element** `Story`
19. Elements

For more information, see §16.1.1.5.

19.64 StoryBreak

element StoryBreak

For more information, see §16.1.2.3.

19.65 StoryFragment

element StoryFragment
### StoryFragments

**Element** StoryFragments

<table>
<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>StoryName</td>
<td>xs:string</td>
<td>optional</td>
<td></td>
<td></td>
<td>Identifies the story that this story fragment belongs to. If omitted, the story fragment is not associated with any story.</td>
</tr>
<tr>
<td>FragmentName</td>
<td>xs:string</td>
<td>optional</td>
<td></td>
<td></td>
<td>Used to uniquely identify the story fragment.</td>
</tr>
<tr>
<td>FragmentType</td>
<td>ST_FragmentType</td>
<td>required</td>
<td></td>
<td></td>
<td>Specifies the type of content included in the story fragment. Valid values are Content, Header, and Footer.</td>
</tr>
</tbody>
</table>

**Diagram:**
- CT_StoryFragment
- StoryFragment
- StoryBreak
- SectionStructure
- ParagraphStructure
- ListStructure
- TableStructure
- FigureStructure
- 1..∞

**Annotation:**
- Specifies the document structural markup that appears on the current page for a single story block.

---

1. For more information, see §16.1.2.2.

---

2. **19.66** StoryFragments

3. element StoryFragments
The root of a StoryFragments part. Contains all story fragments that appear on a specific page.

For more information, see §16.1.2.1.

---

### 19.67 StoryFragmentReference

**Element:** StoryFragmentReference

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Use</th>
<th>Default</th>
<th>Fixed</th>
<th>Annotation</th>
</tr>
</thead>
<tbody>
<tr>
<td>FragmentName</td>
<td>xs:string</td>
<td>optional</td>
<td></td>
<td></td>
<td>Used to distinguish between multiple story fragments from the same story on a single page.</td>
</tr>
<tr>
<td>Page</td>
<td>ST_IntGEOne</td>
<td>required</td>
<td></td>
<td></td>
<td>Identifies the page number of the document that the story fragment is related to. Page numbers start at 1 and correspond to the order of &lt;PageContent&gt; elements in the FixedDocument part.</td>
</tr>
</tbody>
</table>

Identifies the StoryFragments part where this individual story fragment is defined.

For more information, see §16.1.1.6.

---

### 19.68 TableCellStructure

**Element:** TableCellStructure

---
### TableRowGroupStructure

**element** TableRowGroupStructure

---

**annotation**
Contains the set of table rows that make up a table.

---

For more information, see §16.1.2.7.

---

### TableRowStructure

**element** TableRowStructure

---

**annotation**
Contains the elements that occupy a single cell of a table.

---

For more information, see §16.1.2.9.
19. Elements

For more information, see §16.1.2.8.

## 19.71 TableStructure

**element** TableStructure

For more information, see §16.1.2.6.

## 19.72 VisualBrush

**element** VisualBrush
<table>
<thead>
<tr>
<th>attributes</th>
<th>Name</th>
<th>Type</th>
<th>Use</th>
<th>Default</th>
<th>Fixed</th>
<th>Annotation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Opacity</td>
<td>ST_ZeroOne</td>
<td>1.0</td>
<td></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></td>
<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 [M6.4].</td>
</tr>
<tr>
<td></td>
<td>Transform</td>
<td>ST_RscRefMatrix</td>
<td></td>
<td></td>
<td></td>
<td>Describes the matrix transformation applied to the coordinate space of the brush. The Transform property is concatenated with the current effective render transform to yield an effective render transform local to the brush. The viewport for the brush is transformed using that local effective render transform.</td>
</tr>
<tr>
<td></td>
<td>Viewbox</td>
<td>ST_Viewport</td>
<td>required</td>
<td></td>
<td></td>
<td>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 &lt;VisualBrush.Visual&gt; property element. The corners of the viewbox are mapped to the...</td>
</tr>
</tbody>
</table>
corners of the viewport, thereby providing the default clipping and transform for the brush’s source content.

**Viewport**

<table>
<thead>
<tr>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_ViewBox</td>
</tr>
</tbody>
</table>

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 \((x, y, \text{Width}, \text{Height})\), where width and height are non-negative. The alignment of the brush pattern is controlled by adjusting the \(x\) and \(y\) values.

**TileMode**

<table>
<thead>
<tr>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_TileMode</td>
</tr>
</tbody>
</table>

None

Specifies how contents will be tiled in the filled region. Valid values are None, Tile, FlipX, FlipY, and FlipXY.

**ViewboxUnits**

<table>
<thead>
<tr>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_ViewUnits</td>
</tr>
</tbody>
</table>

Absolute

Specifies the relationship of the viewbox coordinates to the containing coordinate space.

**ViewportUnits**

<table>
<thead>
<tr>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_ViewUnits</td>
</tr>
</tbody>
</table>

Absolute

Specifies the relationship of the viewport coordinates to the containing coordinate space.

**Visual**

<table>
<thead>
<tr>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST_RscRef</td>
</tr>
</tbody>
</table>

Specifies resource reference to a \(<\text{Path}>, \langle\text{Glyphs}\rangle, \langle\text{Canvas}\rangle\) 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 \(<\text{VisualBrush}\>) element, or as a resource reference. Drawing content is expressed using \(<\text{Canvas}>, \langle\text{Path}\rangle\), and \(<\text{Glyphs}\rangle\) elements.

For more information, see §13.3.

**19.73 VisualBrush.Transform**

**Element** VisualBrush.Transform

**Diagram**

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.
For more information, see §14.4.

19.74 VisualBrush.Visual

element VisualBrush.Visual

- Specifies a <Path> element, <Glyphs> element, or <Canvas> element used to draw the brush’s source contents.
A. Signature Definitions W3C Schema

The schema shown below is also provided in electronic form as a file named SignatureDefinitions.xsd, which is contained in an accompanying zip archive named "XPS WC3 Schemas.zip". If discrepancies exist between the representation as published below and the corresponding electronic version, the published version below is the definitive version.

```xml
<?xml version="1.0" encoding="utf-8"?>
  <xs:complexType name="SignatureDefinitionsType">
    <xs:sequence>
      <xs:element name="SignatureDefinition" type="SignatureDefinitionType" minOccurs="1" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="SpotLocationType">
    <xs:attribute name="PageURI" type="xs:anyURI" use="required"/>
    <xs:attribute name="StartX" type="xs:double" use="required"/>
    <xs:attribute name="StartY" type="xs:double" use="required"/>
  </xs:complexType>

  <xs:complexType name="SignatureDefinitionType">
    <xs:sequence>
      <xs:element name="SpotLocation" type="SpotLocationType" minOccurs="0"/>
      <xs:element name="Intent" type="xs:string" minOccurs="0"/>
      <xs:element name="SignBy" type="xs:dateTime" minOccurs="0"/>
      <xs:element name="SigningLocation" type="xs:string" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="SpotID" type="xs:ID" use="required"/>
    <xs:attribute name="SignerName" type="xs:string"/>
  </xs:complexType>
</xs:schema>
```
A. Signature Definitions W3C Schema

XPS Specification and Reference Guide

1     <xs:element name="SignatureDefinitions" type="SignatureDefinitionsType">
2         </xs:element>
3     </xs:schema>

4
B. XPS Document **W3C** Schema

The schema shown below is also provided in electronic form as a file named S0schema.xsd, which is contained in an accompanying zip archive named "XPS WC3 Schemas.zip". If discrepancies exist between the representation as published below and the corresponding electronic version, the published version below is the definitive version.

```xml
<?xml version="1.0" encoding="utf-8"?>
<xs:schema targetNamespace="http://schemas.microsoft.com/xps/2005/06"
xmlns="http://schemas.microsoft.com/xps/2005/06"
xmlns:mstns="http://tempuri.org/XMLSchema.xsd"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:x="http://schemas.microsoft.com/xps/2005/06/resourcedictionary-key"
elementFormDefault="qualified" blockDefault="#all">
  <xs:import namespace="http://schemas.microsoft.com/xps/2005/06/resourcedictionary-key" />
  <!-- Names used for types and groups:
   ST_*       simpleType
   CT_*       complexType
   G_*        group
   AG_*       attributeGroup
   -->
  <!-- Individual real number patterns
   All patterns using numbers now use <whitespace value="collapse">.
   As a result, any whitespace in the pattern can be expressed as:
     mandatory whitespace, one or more:  " 
     optional whitespace, zero or more:  " ?"
   For better readability, each pattern using numbers is also described in a comment using
   one of the following pattern designators.
   The actual patterns are generated by replacement by the schema publication process.
   -->
  <!--DEFINE [pint]       "([1-9][0-9]*)" -->
  <!--DEFINE [uint]       "([0-9]+)" -->
  <!--DEFINE [scs]        "( ?, ?)" -->
```
B. XPS Document W3C Schema

1. Complex Types -->

2. <xs:complexType name="CT_MatrixTransform">
3.   <xs:attributeGroup ref="AG_MatrixTransform" />
4. </xs:complexType>

5. <xs:complexType name="CT_SolidColorBrush">
6.   <xs:attributeGroup ref="AG_Brush" />
7.   <xs:attributeGroup ref="AG_SolidColorBrush" />
8. </xs:complexType>

9. <xs:complexType name="CT_ImageBrush">
10.  <xs:sequence>
11.   <xs:element ref="ImageBrush.Transform" minOccurs="0" />
12.  </xs:sequence>
13.  <xs:attributeGroup ref="AG_Brush" />
14.  <xs:attributeGroup ref="AG_TileBrush" />
15.  <xs:attributeGroup ref="AG_ImageBrush" />
16. </xs:complexType>

17. <xs:complexType name="CT_VisualBrush">
18.  <xs:sequence>
19.   <xs:element ref="VisualBrush.Transform" minOccurs="0" />
20.   <xs:element ref="VisualBrush.Visual" minOccurs="0" />
21.  </xs:sequence>
22.  <xs:attributeGroup ref="AG_Brush" />
23.  <xs:attributeGroup ref="AG_TileBrush" />
24.  <xs:attributeGroup ref="AG_VisualBrush" />
25. </xs:complexType>

26. <xs:complexType name="CT_LinearGradientBrush">
27.  <xs:sequence>
28.   <xs:element ref="LinearGradientBrush.Transform" minOccurs="0" />
29.   <xs:element ref="LinearGradientBrush.GradientStops" />
30.  </xs:sequence>
31.  <xs:attributeGroup ref="AG_Brush" />
32.  <xs:attributeGroup ref="AG_GradientBrush" />
33.  <xs:attributeGroup ref="AG_LinearGradientBrush" />
34. </xs:complexType>

35. <xs:complexType name="CT_RadialGradientBrush">
36.  <xs:sequence>
37.   <xs:element ref="RadialGradientBrush.Transform" minOccurs="0" />
38.   <xs:element ref="RadialGradientBrush.GradientStops" />
39.  </xs:sequence>
40.  <xs:attributeGroup ref="AG_Brush" />
41.  <xs:attributeGroup ref="AG_GradientBrush" />
42.  <xs:attributeGroup ref="AG_RadialGradientBrush" />
43. </xs:complexType>

44. <xs:complexType name="CT_GradientStop">
45.  <xs:attributeGroup ref="AG_GradientStop" />
46. </xs:complexType>

47. <xs:complexType name="CT_PathGeometry">
48.  <xs:sequence>
<xs:element ref="PathGeometry.Transform" minOccurs="0" />
<xs:element ref="PathFigure" minOccurs="0" maxOccurs="unbounded" />
</xs:sequence>
<xs:attributeGroup ref="AG_PathGeometry" />
</xs:complexType>

<xs:complexType name="CT_Glyphs">
<xs:sequence>
<xs:element ref="Glyphs.RenderTransform" minOccurs="0" />
<xs:element ref="Glyphs.Clip" minOccurs="0" />
<xs:element ref="Glyphs.OpacityMask" minOccurs="0" />
<xs:element ref="Glyphs.Fill" minOccurs="0" />
</xs:sequence>
<xs:attributeGroup ref="AG_Glyphs" />
</xs:complexType>

<xs:complexType name="CT_Path">
<xs:sequence>
<xs:element ref="Path.RenderTransform" minOccurs="0" />
<xs:element ref="Path.Clip" minOccurs="0" />
<xs:element ref="Path.OpacityMask" minOccurs="0" />
<xs:element ref="Path.Fill" minOccurs="0" />
<xs:element ref="Path.Stroke" minOccurs="0" />
<xs:element ref="Path.Data" minOccurs="0" />
</xs:sequence>
<xs:attributeGroup ref="AG_Path" />
<xs:attributeGroup ref="AG_AutomationProvider" />
<xs:attributeGroup ref="AG_SnapsToDevicePixels" />
</xs:complexType>

<xs:complexType name="CT_PathFigure">
<xs:sequence>
<xs:choice maxOccurs="unbounded">
<xs:element ref="PolyLineSegment" />
<xs:element ref="PolyBezierSegment" />
<xs:element ref="ArcSegment" />
<xs:element ref="PolyQuadraticBezierSegment" />
</xs:choice>
</xs:sequence>
<xs:attributeGroup ref="AG_PathFigure" />
</xs:complexType>

<xs:complexType name="CT_ArcSegment">
<xs:attributeGroup ref="AG_ArcSegment" />
</xs:complexType>

<xs:complexType name="CT_PolyQuadraticBezierSegment">
<xs:attributeGroup ref="AG_PolyQuadraticBezierSegment" />
</xs:complexType>

<xs:complexType name="CT_PolyLineSegment">
<xs:attributeGroup ref="AG_PolyLineSegment" />
</xs:complexType>

<xs:complexType name="CT_PolyBezierSegment">
</xs:complexType>
<xs:attributeGroup ref="AG_PolyBezierSegment" />  
</xs:complexType>  

<xsl:complexType name="CT_Canvas">  
  <xs:sequence>  
    <xs:element ref="Canvas.Resources" minOccurs="0" />  
    <xs:element ref="Canvas.RenderTransform" minOccurs="0" />  
    <xs:element ref="Canvas.Clip" minOccurs="0" />  
    <xs:element ref="Canvas.OpacityMask" minOccurs="0" />  
    <xs:choice minOccurs="0" maxOccurs="unbounded">  
      <xs:element ref="Path" />  
      <xs:element ref="Glyphs" />  
      <xs:element ref="Canvas" />  
    </xs:choice>  
  </xs:sequence>  
  <xs:attributeGroup ref="AG_Canvas" />  
  <xs:attributeGroup ref="AG_AutomationProvider" />  
</xs:complexType>  

<xsl:complexType name="CT_ResourceDictionary">  
  <xs:choice minOccurs="0" maxOccurs="unbounded">  
    <xs:element ref="ImageBrush" />  
    <xs:element ref="LinearGradientBrush" />  
    <xs:element ref="RadialGradientBrush" />  
    <xs:element ref="VisualBrush" />  
    <xs:element ref="SolidColorBrush" />  
    <xs:element ref="MatrixTransform" />  
    <xs:element ref="PathGeometry" />  
    <xs:element ref="Path" />  
    <xs:element ref="Glyphs" />  
    <xs:element ref="Canvas" />  
  </xs:choice>  
  <xs:attributeGroup ref="AG_ResourceDictionary" />  
</xs:complexType>  

<xsl:complexType name="CT_FixedPage">  
  <xs:sequence>  
    <xs:element ref="FixedPage.Resources" minOccurs="0" />  
    <xs:choice minOccurs="0" maxOccurs="unbounded">  
      <xs:element ref="Path" />  
      <xs:element ref="Glyphs" />  
      <xs:element ref="Canvas" />  
    </xs:choice>  
  </xs:sequence>  
  <xs:attributeGroup ref="AG_FixedPage" />  
</xs:complexType>  

<xsl:complexType name="CT_FixedDocument">  
  <xs:sequence>  
    <xs:element ref="PageContent" maxOccurs="unbounded" />  
  </xs:sequence>  
</xs:complexType>  

<xsl:complexType name="CT_PageContent">  
  <xs:sequence>  
  </xs:sequence>  
</xs:complexType>
<xs:element ref="PageContent.LinkTargets" minOccurs="0" />
  <xs:attributeGroup ref="AG_PageContent" />
</xs:complexType>

<xs:complexType name="CT_FixedDocumentSequence">
  <xs:sequence>
    <xs:element ref="DocumentReference" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="CT_DocumentReference">
  <xs:attributeGroup ref="AG_DocumentReference" />
</xs:complexType>

<xs:complexType name="CT_LinkTarget">
  <xs:attributeGroup ref="AG_LinkTarget" />
</xs:complexType>

<xs:complexType name="CT_CP_LinkTargets">
  <xs:sequence>
    <xs:element ref="LinkTarget" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="CT_CP_Transform">
  <xs:sequence>
    <xs:element ref="MatrixTransform" />  
  </xs:sequence>
</xs:complexType>

<xs:complexType name="CT_CP_Visual">
  <xs:choice>
    <xs:element ref="Path" />
    <xs:element ref="Glyphs" />
    <xs:element ref="Canvas" />
  </xs:choice>
</xs:complexType>

<xs:complexType name="CT_CP_GradientStops">
  <xs:sequence>
    <xs:element ref="GradientStop" minOccurs="2" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="CT_CP_Geometry">
  <xs:sequence>
    <xs:element ref="PathGeometry" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="CT_CP_Brush">
  <xs:choice>
    <xs:element ref="ImageBrush" />
    <xs:element ref="LinearGradientBrush" />
  </xs:choice>
</xs:complexType>
<xs:element ref="RadialGradientBrush" />
<xs:element ref="SolidColorBrush" />
<xs:element ref="VisualBrush" />
</xs:choice>
</xs:complexType>

<xs:complexType name="CT_CP_Resources">
<xs:sequence minOccurs="0">
<xs:element ref="ResourceDictionary" />
</xs:sequence>
</xs:complexType>

<!-- Root elements -->
<xs:element name="MatrixTransform" type="CT_MatrixTransform">
</xs:element>
<xs:element name="SolidColorBrush" type="CT_SolidColorBrush">
</xs:element>
<xs:element name="ImageBrush" type="CT_ImageBrush">
</xs:element>
<xs:element name="VisualBrush" type="CT_VisualBrush">
</xs:element>
<xs:element name="LinearGradientBrush" type="CT_LinearGradientBrush">
</xs:element>
<xs:element name="RadialGradientBrush" type="CT_RadialGradientBrush">
</xs:element>
<xs:element name="Glyphs" type="CT_Glyphs">
</xs:element>
<xs:element name="Path" type="CT_Path">
</xs:element>
<xs:element name="Canvas" type="CT_Canvas">
</xs:element>
<xs:element name="GradientStop" type="CT_GradientStop">
</xs:element>
<xs:element name="ResourceDictionary" type="CT_ResourceDictionary">
</xs:element>
<xs:element name="PathGeometry" type="CT_PathGeometry">
</xs:element>
<xs:element name="PathFigure" type="CT_PathFigure">
</xs:element>
<xs:element name="PolyLineSegment" type="CT_PolyLineSegment">
</xs:element>
<xs:element name="ArcSegment" type="CT_ArcSegment">
  </xs:element>

<xs:element name="PolyBezierSegment" type="CT_PolyBezierSegment">
  </xs:element>

<xs:element name="PolyQuadraticBezierSegment" type="CT_PolyQuadraticBezierSegment">
  </xs:element>

<xs:element name="FixedPage" type="CT_FixedPage">
  </xs:element>

<xs:element name="FixedDocument" type="CT_FixedDocument">
  </xs:element>

<xs:element name="PageContent" type="CT_PageContent">
  </xs:element>

<xs:element name="FixedDocumentSequence" type="CT_FixedDocumentSequence">
  </xs:element>

<xs:element name="DocumentReference" type="CT_DocumentReference">
  </xs:element>

<xs:element name="LinkTarget" type="CT_LinkTarget">
  </xs:element>

<xs:element name="PageContent.LinkTargets" type="CT_CP_LinkTargets">
  </xs:element>

<xs:element name="ImageBrush.Transform" type="CT_CP_Transform">
  </xs:element>

<xs:element name="VisualBrush.Transform" type="CT_CP_Transform">
  </xs:element>

<xs:element name="LinearGradientBrush.Transform" type="CT_CP_Transform">
  </xs:element>

<xs:element name="RadialGradientBrush.Transform" type="CT_CP_Transform">
  </xs:element>

<xs:element name="PathGeometry.Transform" type="CT_CP_Transform">
  </xs:element>

<xs:element name="Glyphs.RenderTransform" type="CT_CP_Transform">
  </xs:element>

<xs:element name="Path.RenderTransform" type="CT_CP_Transform">
  </xs:element>

<xs:element name="Canvas.RenderTransform" type="CT_CP_Transform">
  </xs:element>

<xs:element name="VisualBrush.Visual" type="CT_CP_Visual">
  </xs:element>
</xs:element>

<xs:element name="LinearGradientBrush.GradientStops" type="CT_CP_GradientStops">
</xs:element>

<xs:element name="RadialGradientBrush.GradientStops" type="CT_CP_GradientStops">
</xs:element>

<xs:element name="Glyphs.Clip" type="CT_CP_Geometry">
</xs:element>

<xs:element name="Path.Clip" type="CT_CP_Geometry">
</xs:element>

<xs:element name="Canvas.Clip" type="CT_CP_Geometry">
</xs:element>

<xs:element name="Glyphs.OpacityMask" type="CT_CP_Brush">
</xs:element>

<xs:element name="Path.OpacityMask" type="CT_CP_Brush">
</xs:element>

<xs:element name="Canvas.OpacityMask" type="CT_CP_Brush">
</xs:element>

<xs:element name="Glyphs.Fill" type="CT_CP_Brush">
</xs:element>

<xs:element name="Path.Fill" type="CT_CP_Brush">
</xs:element>

<xs:element name="Path.Data" type="CT_CP_Geometry">
</xs:element>

<xs:element name="Path.Stroke" type="CT_CP_Brush">
</xs:element>

<xs:element name="Canvas.Resources" type="CT_CP_Resources">
</xs:element>

<xs:element name="FixedPage.Resources" type="CT_CP_Resources">
</xs:element>

<xs:attributeGroup name="AG_GradientStop">
  <xs:attribute name="Color" type="ST_Color" use="required">
  </xs:attribute>
  <xs:attribute name="Offset" type="ST_Double" use="required">
  </xs:attribute>
</xs:attributeGroup>

<xs:attributeGroup name="AG_Brush">
  <xs:attribute name="Opacity" type="ST_ZeroOne" default="1.0">
  </xs:attribute>
  <xs:attribute ref="x:Key" />
</xs:attributeGroup>
<xs:attributeGroup name="AG_TileBrush">
  <xs:attribute name="Transform" type="ST_RscRefMatrix">
  </xs:attribute>
  <xs:attribute name="Viewbox" type="ST_ ViewBox" use="required">
  </xs:attribute>
  <xs:attribute name="Viewport" type="ST_ ViewBox" use="required">
  </xs:attribute>
  <xs:attribute name="TileMode" type="ST_TileMode" default="None">
  </xs:attribute>
  <xs:attribute name="ViewboxUnits" type="ST_ViewUnits" use="required" fixed="Absolute">
  </xs:attribute>
  <xs:attribute name="ViewportUnits" type="ST_ViewUnits" use="required" fixed="Absolute">
  </xs:attribute>
</xs:attributeGroup>

<xs:attributeGroup name="AG_VisualBrush">
  <xs:attribute name="Visual" type="ST_RscRef">
  </xs:attribute>
</xs:attributeGroup>

<xs:attributeGroup name="AG_GradientBrush">
  <xs:attribute name="ColorInterpolationMode" type="ST_ClrIntMode" default="SRgbLinearInterpolation">
  </xs:attribute>
  <xs:attribute name="SpreadMethod" type="ST_SpreadMethod" default="Pad">
  </xs:attribute>
  <xs:attribute name="MappingMode" type="ST_MappingMode" use="required" fixed="Absolute">
  </xs:attribute>
</xs:attributeGroup>

<xs:attributeGroup name="AG_SolidColorBrush">
  <xs:attribute name="Color" type="ST_Color" use="required">
  </xs:attribute>
</xs:attributeGroup>

<xs:attributeGroup name="AG_ImageBrush">
  <xs:attribute name="ImageSource" type="ST_UriCtxBmp" use="required">
  </xs:attribute>
</xs:attributeGroup>

<xs:attributeGroup name="AG_LinearGradientBrush">
  <xs:attribute name="Transform" type="ST_RscRefMatrix">
  </xs:attribute>
  <xs:attribute name="StartPoint" type="ST_Point" use="required">
  </xs:attribute>
  <xs:attribute name="EndPoint" type="ST_Point" use="required">
  </xs:attribute>
</xs:attributeGroup>

<xs:attributeGroup name="AG_RadialGradientBrush">
  <xs:attribute name="Transform" type="ST_RscRefMatrix">
  </xs:attribute>
  <xs:attribute name="StartPoint" type="ST_Point" use="required">
  </xs:attribute>
  <xs:attribute name="EndPoint" type="ST_Point" use="required">
  </xs:attribute>
</xs:attributeGroup>
<xs:attribute name="Transform" type="ST_RscRefMatrix"/>
</xs:attribute>
<xs:attribute name="Center" type="ST_Point" use="required">
</xs:attribute>
<xs:attribute name="GradientOrigin" type="ST_Point" use="required">
</xs:attribute>
<xs:attribute name="RadiusX" type="ST_GEZero" use="required">
</xs:attribute>
<xs:attribute name="RadiusY" type="ST_GEZero" use="required">
</xs:attribute>

<xs:attribute name="Figures" type="ST_AbbrGeom">
</xs:attribute>
<xs:attribute name="FillRule" type="ST_FillRule" default="EvenOdd">
</xs:attribute>
<xs:attribute name="Transform" type="ST_RscRefMatrix">
</xs:attribute>
<xs:attribute ref="x:Key" />
</xs:attributeGroup>

<xs:attribute name="Point" type="ST_Point" use="required">
</xs:attribute>
<xs:attribute name="Size" type="ST_PointGE0" use="required">
</xs:attribute>
<xs:attribute name="RotationAngle" type="ST_Double" use="required">
</xs:attribute>
<xs:attribute name="IsLargeArc" type="ST_Boolean" use="required">
</xs:attribute>
<xs:attribute name="SweepDirection" type="ST_SweepDirection" use="required">
</xs:attribute>
<xs:attribute name="IsStroked" type="ST_Boolean" default="true">
</xs:attribute>

<xs:attributeGroup name="AG_ArcSegment">
<xs:attribute name="Point" type="ST_Point" use="required">
</xs:attribute>
<xs:attribute name="Size" type="ST_PointGE0" use="required">
</xs:attribute>
<xs:attribute name="RotationAngle" type="ST_Double" use="required">
</xs:attribute>
<xs:attribute name="IsLargeArc" type="ST_Boolean" use="required">
</xs:attribute>
<xs:attribute name="SweepDirection" type="ST_SweepDirection" use="required">
</xs:attribute>
<xs:attribute name="IsStroked" type="ST_Boolean" default="true">
</xs:attribute>
</xs:attributeGroup>

<xs:attributeGroup name="AG_PolyBezierSegment">
<xs:attribute name="Points" type="ST_Points" use="required">
</xs:attribute>
<xs:attribute name="IsStroked" type="ST_Boolean" default="true">
</xs:attribute>
</xs:attributeGroup>

<xs:attribute name="Source" type="xs:anyURI">
</xs:attribute>
</xs:attributeGroup>

<xs:attributeGroup name="AG_PolyLineSegment">
<xs:attribute name="Points" type="ST_Points" use="required">
</xs:attribute>
<xs:attribute name="IsStroked" type="ST_Boolean" default="true">
</xs:attribute>
</xs:attributeGroup>

<xs:attribute name="AG_ResourceDictionary">
<xs:attribute name="Source" type="xs:anyURI">
</xs:attribute>
</xs:attributeGroup>

<xs:attributeGroup name="AG_PolyLineSegment">
<xs:attribute name="Points" type="ST_Points" use="required">
</xs:attribute>
<xs:attribute name="IsStroked" type="ST_Boolean" default="true">
</xs:attribute>
</xs:attributeGroup>

<xs:attributeGroup name="AG_ArcSegment">
<xs:attribute name="Point" type="ST_Point" use="required">
</xs:attribute>
<xs:attribute name="Size" type="ST_PointGE0" use="required">
</xs:attribute>
<xs:attribute name="RotationAngle" type="ST_Double" use="required">
</xs:attribute>
<xs:attribute name="IsLargeArc" type="ST_Boolean" use="required">
</xs:attribute>
<xs:attribute name="SweepDirection" type="ST_SweepDirection" use="required">
</xs:attribute>
<xs:attribute name="IsStroked" type="ST_Boolean" default="true">
</xs:attribute>
</xs:attributeGroup>

<xs:attributeGroup name="AG_PolyBezierSegment">
<xs:attribute name="Points" type="ST_Points" use="required">
</xs:attribute>
<xs:attribute name="IsStroked" type="ST_Boolean" default="true">
</xs:attribute>
</xs:attributeGroup>
<xs:attributeGroup name="AG_PolyQuadraticBezierSegment">
  <xs:attribute name="Points" type="ST_Points" use="required">
    <xs:attribute>
  </xs:attribute>
  <xs:attribute name="IsStroked" type="ST_Boolean" default="true">
    <xs:attribute>
  </xs:attribute>
</xs:attributeGroup>

<xs:attributeGroup name="AG_Glyphs">
  <xs:attribute name="BidiLevel" default="0">
    <xs:simpleType>
      <xs:restriction base="xs:integer">
        <xs:minInclusive value="0" />
        <xs:maxInclusive value="61" />
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="CaretStops" type="ST_CaretStops">
    <xs:attribute>
  </xs:attribute>
  <xs:attribute name="DeviceFontName" type="ST_UnicodeString">
    <xs:attribute>
  </xs:attribute>
  <xs:attribute name="Fill" type="ST_RscRefColor">
    <xs:attribute>
  </xs:attribute>
  <xs:attribute name="FontRenderingEmSize" type="ST_GEZero" use="required">
    <xs:attribute>
  </xs:attribute>
  <xs:attribute name="FontUri" type="xs:anyURI" use="required">
    <xs:attribute>
  </xs:attribute>
  <xs:attribute name="OriginX" type="ST_Double" use="required">
    <xs:attribute>
  </xs:attribute>
  <xs:attribute name="OriginY" type="ST_Double" use="required">
    <xs:attribute>
  </xs:attribute>
  <xs:attribute name="IsSideways" type="ST_Boolean" default="false">
    <xs:attribute>
  </xs:attribute>
  <xs:attribute name="Indices" type="ST_Indices">
    <xs:attribute>
  </xs:attribute>
  <xs:attribute name="UnicodeString" type="ST_UnicodeString">
    <xs:attribute>
  </xs:attribute>
  <xs:attribute name="StyleSimulations" type="ST_StyleSimulations" default="None">
    <xs:attribute>
  </xs:attribute>
  <xs:attribute name="RenderTransform" type="ST_RscRefMatrix">
    <xs:attribute>
  </xs:attribute>
  <xs:attribute name="Clip" type="ST_RscRefAbbrGeomF">
    <xs:attribute>
  </xs:attribute>
  <xs:attribute name="Opacity" type="ST_ZeroOne" default="1.0">
    <xs:attribute>
  </xs:attribute>
  <xs:attribute name="OpacityMask" type="ST_RscRef">
    <xs:attribute>
  </xs:attribute>
  <xs:attribute name="Name" type="ST_Name">
    <xs:attribute>
  </xs:attribute>
  <xs:attribute name="FixedPage.NavigateUri" type="xs:anyURI">
    <xs:attribute>
  </xs:attribute>
  <xs:attribute ref="xml:lang" />
  <xs:attribute ref="x:Key" />
</xs:attributeGroup>
### B. XPS Document W3C Schema

```xml
<xsd:attributeGroup name="AG_Path">
  <xsd:attribute name="Data" type="ST_RscRefAbbrGeomF"/>
  <xsd:attribute name="Fill" type="ST_RscRefColor"/>
  <xsd:attribute name="RenderTransform" type="ST_RscRefMatrix"/>
  <xsd:attribute name="Clip" type="ST_RscRefAbbrGeomF"/>
  <xsd:attribute name="Opacity" type="ST_ZeroOne" default="1.0"/>
  <xsd:attribute name="OpacityMask" type="ST_RscRef"/>
  <xsd:attribute name="Stroke" type="ST_RscRefColor"/>
  <xsd:attribute name="StrokeDashArray" type="ST_EvenArrayPos"/>
  <xsd:attribute name="StrokeDashCap" type="ST_DashCap" default="Flat"/>
  <xsd:attribute name="StrokeDashOffset" type="ST_Double" default="0.0"/>
  <xsd:attribute name="StrokeEndLineCap" type="ST_LineCap" default="Flat"/>
  <xsd:attribute name="StrokeStartLineCap" type="ST_LineCap" default="Flat"/>
  <xsd:attribute name="StrokeLineJoin" type="ST_LineJoin" default="Miter"/>
  <xsd:attribute name="StrokeMiterLimit" type="ST_GEOne" default="10.0"/>
  <xsd:attribute name="StrokeThickness" type="ST_GEZero" default="1.0"/>
  <xsd:attribute name="Name" type="ST_Name"/>
  <xsd:attribute name="FixedPage.NavigateUri" type="xs:anyURI"/>
  <xsd:attribute ref="xml:lang"/>
  <xsd:attribute ref="x:Key"/>
</xsd:attributeGroup>

<xsd:attributeGroup name="AG_PathFigure">
  <xsd:attribute name="IsClosed" type="ST_Boolean" default="false"/>
  <xsd:attribute name="StartPoint" type="ST_Point" use="required"/>
  <xsd:attribute name="IsFilled" type="ST_Boolean" default="true"/>
</xsd:attributeGroup>

<xsd:attributeGroup name="AG_Canvas">
  <xsd:attribute name="RenderTransform" type="ST_RscRefMatrix"/>
  <xsd:attribute name="Clip" type="ST_RscRefAbbrGeomF"/>
</xsd:attributeGroup>
```
<xs:attribute name="Opacity" type="ST_ZeroOne" default="1.0">
</xs:attribute>
<xs:attribute name="OpacityMask" type="ST_RscRef">
</xs:attribute>
<xs:attribute name="Name" type="ST_Name">
</xs:attribute>
<xs:attribute name="RenderOptions.EdgeMode" type="ST_EdgeMode">
</xs:attribute>
<xs:attribute name="FixedPage.NavigateUri" type="xs:anyURI">
</xs:attribute>
<xs:attribute ref="xml:lang">
</xs:attribute>
<xs:attribute ref="x:Key" />
</xs:attributeGroup>

<xs:attributeGroup name="AG_PageContent">
<xs:attribute name="Source" type="xs:anyURI" use="required">
</xs:attribute>
<xs:attribute name="Width" type="ST_GEOne">
</xs:attribute>
<xs:attribute name="Height" type="ST_GEOne">
</xs:attribute>
</xs:attributeGroup>

<xs:attributeGroup name="AG_LinkTarget">
<xs:attribute name="Name" type="ST_NUName" use="required">
</xs:attribute>
</xs:attributeGroup>

<xs:attributeGroup name="AG_DocumentReference">
<xs:attribute name="Source" type="xs:anyURI" use="required">
</xs:attribute>
</xs:attributeGroup>

<xs:attributeGroup name="AG_MatrixTransform">
<xs:attribute name="Matrix" type="ST_Matrix" use="required">
</xs:attribute>
<xs:attribute ref="x:Key" />
</xs:attributeGroup>

<xs:attributeGroup name="AG_FixedPage">
<xs:attribute name="Width" type="ST_GEOne" use="required">
</xs:attribute>
<xs:attribute name="Height" type="ST_GEOne" use="required">
</xs:attribute>
<xs:attribute name="ContentBox" type="ST_ContentBox">
</xs:attribute>
<xs:attribute name="BleedBox" type="ST_BleedBox">
</xs:attribute>
<xs:attribute ref="xml:lang" use="required">
</xs:attribute>
<xs:attribute name="Name" type="ST_Name">
</xs:attribute>
</xs:attributeGroup>
<xs:attributeGroup name="AG_AutomationProvider">
  <xs:attribute name="AutomationProperties.Name" type="xs:string">
  </xs:attribute>
  <xs:attribute name="AutomationProperties.HelpText" type="xs:string">
  </xs:attribute>
</xs:attributeGroup>

<xs:attributeGroup name="AG_SnapsToDevicePixels">
  <xs:attribute name="SnapsToDevicePixels" type="ST_Boolean">
  </xs:attribute>
</xs:attributeGroup>

<!-- Simple data types -->
<!-- A unique Name (ID with pattern restriction according to XPS spec) -->
<xs:restriction base="xs:ID">
  <xs:pattern value="(\p{Lu}|\p{Ll}|\p{Lt}|\p{Lo}|\p{Nl}|\p{Mn}\p{Mc}|\p{Nd})*" />
</xs:restriction>
</xs:simpleType>

<!-- A non-unique Name (ID with pattern restriction according to XPS spec) -->
<xs:restriction base="xs:string">
  <xs:pattern value="(\p{Lu}|\p{Ll}|\p{Lt}|\p{Lo}|\p{Nl}|\p{Mn}\p{Mc}|\p{Nd})*" />
</xs:restriction>
</xs:simpleType>

<!-- Boolean with true and false only (no 0 or 1) -->
<xs:restriction base="xs:boolean">
  <xs:pattern value="true|false" />
</xs:restriction>
</xs:simpleType>

<!-- real number from 0.0 to 1.0 inclusive -->
<xs:restriction base="ST_Double">
  <xs:minInclusive value="0.0" />
  <xs:maxInclusive value="1.0" />
</xs:restriction>
</xs:simpleType>

<!-- positive real number -->
<xs:restriction base="ST_Double">
  <xs:minInclusive value="0.0" />
</xs:restriction>
</xs:simpleType>

<!-- positive real number, equal or greater than one -->
<xs:simpleType name="ST_GEOOne">
  <xs:restriction base="ST_Double">
    <xs:minInclusive value="1.0" />
  </xs:restriction>
</xs:simpleType>

<!-- Double -->
<xs:simpleType name="ST_Double">
  <xs:restriction base="xs:double">
    <xs:whiteSpace value="collapse" />
  </xs:restriction>
</xs:simpleType>

<!-- Point: 2 numbers, separated by , and arbitrary whitespace -->
<xs:simpleType name="ST_Point">
  <xs:restriction base="xs:string">
    <xs:whiteSpace value="collapse" />
    <xs:pattern value="([+-]?\d+(\.\d+)?|\.\d+)((e|E)([+-]|)?\d+)?( ?,)??([+-]?\d+(\.\d+)?|\.\d+)((e|E)([+-]|)?\d+)?" />
  </xs:restriction>
</xs:simpleType>

<!-- PointGE0: 2 non-negative numbers, separated by , and arbitrary whitespace -->
<xs:simpleType name="ST_PointGE0">
  <xs:restriction base="xs:string">
    <xs:whiteSpace value="collapse" />
    <xs:pattern value="([+-]?\d+(\.\d+)?|\.\d+)((e|E)([+-]|)?\d+)?( ?,)??([+-]?\d+(\.\d+)?|\.\d+)((e|E)([+-]|)?\d+)?" />
  </xs:restriction>
</xs:simpleType>

<!-- Points: List of ST_Point, separated by arbitrary whitespace -->
<xs:simpleType name="ST_Points">
  <xs:restriction base="xs:string">
    <xs:whiteSpace value="collapse" />
    <xs:pattern value="([+-]?\d+(\.\d+)?|\.\d+)((e|E)([+-]|)?\d+)?( ?,)??([+-]?\d+(\.\d+)?|\.\d+)((e|E)([+-]|)?\d+)?" />
  </xs:restriction>
</xs:simpleType>
<!-- EvenArray: List with even number of entries of non-negative numbers. -->
<xs:simpleType name="ST_EvenArrayPos">
    <xs:restriction base="xs:string">
        <xs:whiteSpace value="collapse" />
    </xs:restriction>
</xs:simpleType>

<!-- Array: List of numbers. -->
<xs:simpleType name="ST_Array">
    <xs:restriction base="xs:string">
        <xs:whiteSpace value="collapse" />
    </xs:restriction>
</xs:simpleType>

<!-- ViewBox: 4 numbers, separated by , and arbitrary whitespace. Second number pair must be non-negative -->
<xs:simpleType name="ST_ViewBox">
    <xs:restriction base="xs:string">
        <xs:whiteSpace value="collapse" />
    </xs:restriction>
</xs:simpleType>

<!-- ContentBox: 4 non-negative numbers, separated by commas and arbitrary whitespace -->
<xs:simpleType name="ST_ContentBox">
    <xs:restriction base="xs:string">
        <xs:whiteSpace value="collapse" />
    </xs:restriction>
</xs:simpleType>
<!-- BleedBox: 4 numbers, separated by , and arbitrary whitespace. Second number pair must be non-negative -->
<xs:simpleType name="ST_BleedBox">
  <xs:restriction base="xs:string">
    <xs:WhiteSpace value="collapse" />
  </xs:restriction>
</xs:simpleType>

<!-- Bare Matrix form: 6 numbers separated by , and arbitrary whitespace -->
<xs:simpleType name="ST_Matrix">
  <xs:restriction base="xs:string">
    <xs:WhiteSpace value="collapse" />
  </xs:restriction>
</xs:simpleType>

<!-- Color: 6 or 8 hex digits -->
<xs:simpleType name="ST_Color">
  <xs:restriction base="xs:string">
    <xs:WhiteSpace value="collapse" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="ST_CaretStops">
    <xs:restriction base="xs:string">
        <xs:WhiteSpace value="collapse" />
        <xs:pattern value="[0-9A-Fa-f]*/" />
    </xs:restriction>
</xs:simpleType>

<!-- Indices grammar for Glyphs.Indices -->
<xs:simpleType name="ST_Indices">
    <xs:restriction base="xs:string">
        <xs:WhiteSpace value="collapse" />
        <xs:pattern value="((\([pint](?:[pint]?[\{]?[uint]?)\)?[\{]?[uint]?)?([prn]?([rn]?([rn]?[\{]?[uint]?)?\))\)?([prn]?([rn]?([rn]?[\{]?[uint]?)?\))?)*" />
    </xs:restriction>
</xs:simpleType>

<!-- UnicodeString grammar -->
<xs:simpleType name="ST_UnicodeString">
    <xs:restriction base="xs:string">
        <xs:pattern value="([\^\{]\{[\}].*?\})?" />
    </xs:restriction>
</xs:simpleType>

<!-- Abbreviated Geometry grammar for Path.Data , clip and Geometries -->
<xs:simpleType name="ST_AbbrGeomF">
    <xs:restriction base="xs:string">
        <xs:WhiteSpace value="collapse" />
        <xs:pattern value="F ?(0|1)*/\\
                    ( ?(M|m)( ?[rn][scs][rn])*)\\
                    ( ?(L|l)( ?[rn][scs][rn])( ?[rn][scs][rn])*)\\
                    ( ?(H|h|V|v)( ?[rn]( ?[rn][scs][rn])*)\\
                        ( ?(C|c)( ?[rn][scs][rn]( ?[rn][scs][rn])*)\\
                            ( ?(A|a)( ?[rn][scs][rn] ?[rn] ?[rn][scs][rn])*)\\
                        )\\
                    )" />
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="ST_RscRef">
  <xs:restriction base="xs:string">
    <xs:pattern value="\{StaticResource\[\s]+[\s]+\[\s]+\}[^\s]*" />
  </xs:restriction>
</xs:simpleType>

<!-- Resource reference OR Color -->
<xs:simpleType name="ST_RscRefColor">
  <xs:union memberTypes="ST_Color ST_RscRef" />
</xs:simpleType>

<!-- Resource reference OR Compact Matrix-->
<xs:simpleType name="ST_RscRefMatrix">
  <xs:union memberTypes="ST_Matrix ST_RscRef" />
</xs:simpleType>

<!-- Resource reference OR AbbrGeomF-->
<xs:simpleType name="ST_RscRefAbbrGeomF">
  <xs:union memberTypes="ST_AbbrGeomF ST_RscRef" />
</xs:simpleType>

<!-- Sweep Direction enumeration -->
<xs:simpleType name="ST_SweepDirection">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Clockwise" />
    <xs:enumeration value="Counterclockwise" />
  </xs:restriction>
</xs:simpleType>

<!-- Dash Cap enumeration -->
<xs:simpleType name="ST_DashCap">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Flat" />
    <xs:enumeration value="Round" />
    <xs:enumeration value="Square" />
    <xs:enumeration value="Triangle" />
  </xs:restriction>
</xs:simpleType>

<!-- Line Cap enumeration -->
<xs:simpleType name="ST_LineCap">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Flat" />
    <xs:enumeration value="Round" />
    <xs:enumeration value="Square" />
    <xs:enumeration value="Triangle" />
  </xs:restriction>
</xs:simpleType>

<!-- Line Join enumeration -->
<xs:simpleType name="ST_LineJoin">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Miter" />
    <xs:enumeration value="Bevel" />
    <xs:enumeration value="Round" />
  </xs:restriction>
</xs:simpleType>
B. XPS Document W3C Schema

1. Tile Mode enumeration --
2. <xs:simpleType name="ST_TileMode">
3.     <xs:restriction base="xs:string">
4.         <xs:enumeration value="None" />
5.         <xs:enumeration value="Tile" />
6.         <xs:enumeration value="FlipX" />
7.         <xs:enumeration value="FlipY" />
8.         <xs:enumeration value="FlipXY" />
9.     </xs:restriction>
10. </xs:simpleType>

2. Color Interpolation Mode enumeration --
3. <xs:simpleType name="ST_ClrIntMode">
4.     <xs:restriction base="xs:string">
5.         <xs:enumeration value="ScRgbLinearInterpolation" />
6.         <xs:enumeration value="SRgbLinearInterpolation" />
7.     </xs:restriction>
8. </xs:simpleType>

2. SpreadMethod Mode enumeration --
3. <xs:simpleType name="ST_SpreadMethod">
4.     <xs:restriction base="xs:string">
5.         <xs:enumeration value="Pad" />
6.         <xs:enumeration value="Reflect" />
7.         <xs:enumeration value="Repeat" />
8.     </xs:restriction>
9. </xs:simpleType>

2. FillRule Mode enumeration --
3. <xs:simpleType name="ST_FillRule">
4.     <xs:restriction base="xs:string">
5.         <xs:enumeration value="EvenOdd" />
6.         <xs:enumeration value="NonZero" />
7.     </xs:restriction>
8. </xs:simpleType>

2. Edge Mode enumeration --
3. <xs:simpleType name="ST_EdgeMode">
4.     <xs:restriction base="xs:string">
5.         <xs:enumeration value="Aliased" />
6.     </xs:restriction>
7. </xs:simpleType>

2. Style Simulation Enumeration --
3. <xs:simpleType name="ST_StyleSimulations">
4.     <xs:restriction base="xs:string">
5.         <xs:enumeration value="None" />
6.         <xs:enumeration value="ItalicSimulation" />
7.         <xs:enumeration value="BoldSimulation" />
8.         <xs:enumeration value="BoldItalicSimulation" />
9.     </xs:restriction>
10. </xs:simpleType>
1
2     <!-- ViewUnits Enumeration -->
3     <xs:simpleType name="ST_ViewUnits">
4         <xs:restriction base="xs:string">
5             <xs:enumeration value="Absolute" />
6         </xs:restriction>
7     </xs:simpleType>
8
9     <!-- MappingMode Enumeration -->
10    <xs:simpleType name="ST_MappingMode">
11       <xs:restriction base="xs:string">
12           <xs:enumeration value="Absolute" />
13       </xs:restriction>
14    </xs:simpleType>
15 </xs:schema>
C. Resource Dictionary Key W3C Schema

The schema shown below is also provided in electronic form as a file named RDKey.xsd, which is contained in an accompanying zip archive named "XPS WC3 Schemas.zip". If discrepancies exist between the representation as published below and the corresponding electronic version, the published version below is the definitive version.

```xml
<?xml version="1.0" encoding="utf-8"?>
<xs:schema
targetNamespace="http://schemas.microsoft.com/xps/2005/06/resourcedictionary-key"
xmlns="http://schemas.microsoft.com/xps/2005/06/resourcedictionary-key"
xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
blockDefault="#all">
  <xs:attribute name="Key">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <!-- A Key (pattern restriction according to XPS spec) -->
        <xs:pattern
          value="(\p{Lu}|\p{Ll}|\p{Lt}|\p{Lo}|\p{Nl}|_)(\p{Lu}|\p{Ll}|\p{Lt}|\p{Lo}|\p{Nl}|\p{Mn}
          |\p{Mc}|\p{Nd}|_)*" />
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:schema>
```
The schema shown below is also provided in electronic form as a file named DocStructure.xsd, which is contained in an accompanying zip archive named "XPS WC3 Schemas.zip". If discrepancies exist between the representation as published below and the corresponding electronic version, the published version below is the definitive version.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://schemas.microsoft.com/xps/2005/06/documentstructure"
    xmlns="http://schemas.microsoft.com/xps/2005/06/documentstructure"
    xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
    blockDefault="#all">
  <!-- ==============Document Structure Part================ -->
  <!-- Complex Types -->
  <xs:complexType name="CT_DocumentStructure">
    <xs:sequence>
      <xs:element ref="DocumentStructure.Outline" minOccurs="0" />
      <xs:element ref="Story" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="CT_CP_Outline">
    <xs:sequence>
      <xs:element ref="DocumentOutline" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="CT_DocumentOutline">
    <xs:sequence>
      <xs:element ref="OutlineEntry" maxOccurs="unbounded" />
    </xs:sequence>
    <xs:attributeGroup ref="AG_DocumentOutline" />
  </xs:complexType>
  <xs:complexType name="CT_OutlineEntry">
    <xs:attributeGroup ref="AG_OutlineEntry" />
  </xs:complexType>
  <xs:complexType name="CT_Story">
    <xs:sequence>
      <xs:element ref="StoryFragmentReference" maxOccurs="unbounded" />
    </xs:sequence>
    <xs:attributeGroup ref="AG_Story" />
  </xs:complexType>
  <xs:complexType name="CT_StoryFragmentReference">
    <xs:attributeGroup ref="AG_StoryFragmentReference" />
  </xs:complexType>
  <!-- Simple Types -->
  <!-- A Name (ID with pattern restriction according to XPS spec) -->
  <xs:simpleType name="ST_Name">
    <xs:restriction base="xs:string">
      <!-- Pattern restriction according to XPS spec -->
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```
<xs:pattern
value="(\p{Lu}|\p{Ll}|\p{Nl})\p{Lu}\p{Ll}\p{Nl}|\p{Lu}\p{Ll}\p{Nl}\p{Lu}\p{Ll}\p{Nl}\p{Lu}\p{Ll}\p{Nl}\p{Lu}\p{Ll}\p{Nl}\p{Lu}\p{Ll}\p{Nl}\p{Lu}\p{Ll}\p{Nl}\p{Lu}\p{Ll}\p{Nl}\p{Lu}\p{Ll}\p{Nl}\p{Lu}\p{Ll}\p{Nl}\p{Lu}\p{Ll}\p{Nl}\p{Lu}\p{Ll}\p{Nl}\p{Lu}\p{Ll}\p{Nl})*" />
</xs:restriction>
</xs:simpleType>

<!-- A Unique Name (ID with pattern restriction according to XPS spec) -->
<xs:simpleType name="ST_NameUnique">
  <xs:restriction base="xs:ID">
    <xs:pattern
value="(\p{Lu}|\p{Ll}|\p{Nl})\p{Lu}\p{Ll}\p{Nl}|\p{Lu}\p{Ll}\p{Nl}\p{Lu}\p{Ll}\p{Nl}\p{Lu}\p{Ll}\p{Nl}\p{Lu}\p{Ll}\p{Nl}\p{Lu}\p{Ll}\p{Nl}\p{Lu}\p{Ll}\p{Nl}\p{Lu}\p{Ll}\p{Nl}\p{Lu}\p{Ll}\p{Nl}\p{Lu}\p{Ll}\p{Nl}\p{Lu}\p{Ll}\p{Nl}\p{Lu}\p{Ll}\p{Nl}\p{Lu}\p{Ll}\p{Nl})*" />
  </xs:restriction>
</xs:simpleType>

<!-- integer greater than or equal to 1 inclusive -->
<xs:simpleType name="ST_IntGEOne">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="1" />
  </xs:restriction>
</xs:simpleType>

<!-- Elements -->
<xs:element name="DocumentStructure" type="CT_DocumentStructure">
  </xs:element>
<xs:element name="DocumentOutline" type="CT_DocumentOutline">
  </xs:element>
<xs:element name="OutlineEntry" type="CT_OutlineEntry">
  </xs:element>
<xs:element name="Story" type="CT_Story">
  </xs:element>
<xs:element name="StoryFragmentReference" type="CT_StoryFragmentReference">
  </xs:element>
</xs:attributeGroup>
<xs:attributeGroup name="AG_OutlineEntry">
  <xs:attribute name="OutlineLevel" type="ST_IntGEOne" use="optional" default="1">
    </xs:attribute>
  <xs:attribute name="OutlineTarget" type="xs:anyURI" use="required">
    </xs:attribute>
  <xs:attribute name="Description" type="xs:string" use="required">
    </xs:attribute>
  </xs:attributeGroup>
<xs:attributeGroup name="AG_Story">
  <xs:attribute name="StoryName" type="xs:string" use="required">
    </xs:attribute>
</xs:attributeGroup>
<xs:attributeGroup name="AG_StoryFragmentReference">
  <xs:attribute name="FragmentName" type="xs:string" use="optional">
    </xs:attribute>
</xs:attributeGroup>
<xs:attribute name="Page" type="ST_IntGEOne" use="required">
</xs:attribute>
</xs:attributeGroup>
<!-- ===============StoryFragments Part============== -->
<!-- Complex Types -->
<xs:complexType name="CT_StoryFragments">
<xs:sequence>
<xs:element ref="StoryFragment" maxOccurs="unbounded" />
</xs:sequence>
</xs:complexType>
<xs:complexType name="CT_StoryFragment">
<xs:sequence>
<xs:element ref="StoryBreak" minOccurs="0" />
<xs:choice maxOccurs="unbounded">
<xs:element ref="SectionStructure" />
<xs:element ref="ParagraphStructure" />
<xs:element ref="ListStructure" />
<xs:element ref="TableStructure" />
<xs:element ref="FigureStructure" />
</xs:choice>
<xs:element ref="StoryBreak" minOccurs="0" />
</xs:sequence>
<xs:attributeGroup ref="AG_StoryFragment" />
</xs:complexType>
<xs:complexType name="CT_Break">
</xs:complexType>
<xs:complexType name="CT_Section">
<xs:choice maxOccurs="unbounded">
<xs:element ref="ParagraphStructure" />
<xs:element ref="ListStructure" />
<xs:element ref="TableStructure" />
<xs:element ref="FigureStructure" />
</xs:choice>
</xs:complexType>
<xs:complexType name="CT_Paragraph">
<xs:choice minOccurs="0" maxOccurs="unbounded">
<xs:element ref="NamedElement" />
</xs:choice>
</xs:complexType>
<xs:complexType name="CT_Table">
<xs:choice maxOccurs="unbounded">
<xs:element ref="TableRowGroupStructure" />
</xs:choice>
</xs:complexType>
<xs:complexType name="CT_TableRowGroup">
<xs:choice maxOccurs="unbounded">
<xs:element ref="TableRowStructure" />
</xs:choice>
</xs:complexType>
<xs:complexType name="CT_TableRow">
<xs:choice maxOccurs="unbounded">
<xs:element ref="TableCellStructure" />
</xs:choice>
</xs:complexType>
<xs:complexType name="CT_TableCell">
</xs:complexType>
<xs:choice minOccurs="0" maxOccurs="unbounded">
  <xs:element ref="ParagraphStructure" />
  <xs:element ref="ListStructure" />
  <xs:element ref="TableStructure" />
  <xs:element ref="FigureStructure" />
</xs:choice>
<xs:attributeGroup ref="AG_TableCell" />
</xs:complexType>
<xs:complexType name="CT_List">
  <xs:choice maxOccurs="unbounded">
    <xs:element ref="ListItemStructure" />
  </xs:choice>
</xs:complexType>
<xs:complexType name="CT_ListItem">
  <xs:choice minOccurs="0" maxOccurs="unbounded">
    <xs:element ref="ParagraphStructure" />
    <xs:element ref="ListStructure" />
    <xs:element ref="TableStructure" />
    <xs:element ref="FigureStructure" />
  </xs:choice>
  <xs:attributeGroup ref="AG_ListItem" />
</xs:complexType>
<xs:complexType name="CT_Figure">
  <xs:choice minOccurs="0" maxOccurs="unbounded">
    <xs:element ref="NamedElement" />
  </xs:choice>
</xs:complexType>
<xs:complexType name="CT_NamedElement">
  <xs:attributeGroup ref="AG_NamedElement" />
</xs:complexType>

<!-- Simple Types -->
<!-- FragmentType enumeration -->
<xs:simpleType name="ST_FragmentType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Content" />
    <xs:enumeration value="Header" />
    <xs:enumeration value="Footer" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="ST_Location">
  <xs:restriction base="xs:string">
    <xs:pattern value="([0-9][0-9]*)\([0-9][0-9]*\)" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="ST_TableSpan">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="1" />
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="ST_ElementIndex">
  <xs:restriction base="xs:int">
    <xs:minInclusive value="0" />
  </xs:restriction>
</xs:simpleType>

<!-- Elements -->
<xs:element name="StoryFragments" type="CT_StoryFragments">
</xs:element>
<xs:element name="StoryFragment" type="CT_StoryFragment">
</xs:element>
<xs:element name="StoryBreak" type="CT_Break">
</xs:element>
<xs:element name="SectionStructure" type="CT_Section">
</xs:element>
<xs:element name="ParagraphStructure" type="CT_Paragraph">
</xs:element>
<xs:element name="TableStructure" type="CT_Table">
</xs:element>
<xs:element name="TableRowGroupStructure" type="CT_TableRowGroup">
</xs:element>
<xs:element name="TableRowStructure" type="CT_TableRow">
</xs:element>
<xs:element name="TableCellStructure" type="CT_TableCell">
</xs:element>
<xs:element name="ListStructure" type="CT_List">
</xs:element>
<xs:element name="ListItemStructure" type="CT_ListItem">
</xs:element>
<xs:element name="FigureStructure" type="CT_Figure">
</xs:element>
<xs:element name="NamedElement" type="CT_NamedElement">
</xs:element>
</xs:element>
</xs:attributeGroup>
<xs:attributeGroup name="AG_TableCell">
  <xs:attribute name="RowSpan" type="ST_TableSpan" use="optional" default="1">
  </xs:attribute>
  <xs:attribute name="ColumnSpan" type="ST_TableSpan" use="optional" default="1">
  </xs:attribute>
</xs:attributeGroup>
<xs:attributeGroup name="AG_ListItem">
  <xs:attribute name="Marker" type="ST_NameUnique" use="optional">
  </xs:attribute>
</xs:attributeGroup>
<xs:attributeGroup name="AG_NamedElement">
  <xs:attribute name="NameReference" type="ST_Name" use="required">
  </xs:attribute>
</xs:attributeGroup>
</xs:element>
</xs:schema>
E. Discard Control W3C Schema

The schema shown below is also provided in electronic form as a file named DiscardControl.xsd, which is contained in an accompanying zip archive named "XPS WC3 Schemas.zip. If discrepancies exist between the representation as published below and the corresponding electronic version, the published version below is the definitive version.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://schemas.microsoft.com/xps/2005/06/discard-control"
xmlns="http://schemas.microsoft.com/xps/2005/06/discard-control"
xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
blockDefault="#all">

  <xs:complexType name="CT_DiscardControl">
    <xs:sequence>
      <xs:element ref="Discard" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="CT_Discard">
    <xs:attribute name="SentinelPage" type="xs:anyURI" use="required"/>
    <xs:attribute name="Target" type="xs:anyURI" use="required"/>
  </xs:complexType>

  <xs:element name="DiscardControl" type="CT_DiscardControl"/>

  <xs:element name="Discard" type="CT_Discard"/>

</xs:schema>
```
F. Abbreviated Geometry Syntax Algorithm

A path geometry specified using the abbreviated geometry syntax (see §11.2.3) is equivalent to a path specified using a path geometry. The following algorithm describes how the abbreviated path syntax can be transformed into a path geometry containing path figures that, in turn, contain various segments.

This algorithm assumes that the presented string is well-formed according to the markup schema. Whitespace skipping is assumed without being explicitly spelled out in the algorithm.

```plaintext
Let CURRENTPOINT = 0,0
Create a new PathGeometry PG
PG.FillRule = EvenOdd
Let CURRENTPATHFIGURE = undefined
Read input character CH
if ( CH == 'F' )
    
    Read input character CH
    if ( CH == '0' )
        
        PG.FillRule = EvenOdd
    
    else
        
        PG.FillRule = NonZero
    
} else
    
    GOTO label_first
label_repeat:
Read input character CH
label_first:
if ( CH == 'm' )
    
    Read relative coordinate pair DX,DY
    Let CURRENTPOINT.X = CURRENTPOINT.X + DX
    Let CURRENTPOINT.Y = CURRENTPOINT.Y + DY
    Create a new PathFigure CURRENTPATHFIGURE and add to PG
    Let attribute CURRENTPATHFIGURE.StartPoint = CURRENTPOINT
} else if ( CH == 'M' )
    
    Read coordinate pair X,Y
    Let CURRENTPOINT.X = X
    Let CURRENTPOINT.Y = Y
    Create a new PathFigure CURRENTPATHFIGURE and add to PG
    Let attribute CURRENTPATHFIGURE.StartPoint = CURRENTPOINT
} else if ( CH == 'l' )
    
    Create new PolyLineSegment S
    Add S to CURRENTPATHFIGURE
label_1:
```
Read relative coordinate pair DX, DY
Let CURRENTPOINT.X = CURRENTPOINT.X + DX
Let CURRENTPOINT.Y = CURRENTPOINT.Y + DY
Add CURRENTPOINT.X, CURRENTPOINT.Y to end of S.Points attribute list
if ( next character is not a letter )
{
    GOTO label_1
}
else if ( CH == 'L' )
{
    Create new PolyLineSegment S
    Add S to CURRENTPATHFIGURE
}
lable_2:
Read coordinate pair X, Y
Let CURRENTPOINT.X = X
Let CURRENTPOINT.Y = Y
Add CURRENTPOINT.X, CURRENTPOINT.Y to end of S.Points attribute list
if ( next character is not a letter )
{
    GOTO label_2
}
else if ( CH == 'h' )
{
    Create new PolyLineSegment S
    Add S to CURRENTPATHFIGURE
}
lable_3:
Read relative coordinate value DX
Let CURRENTPOINT.X = CURRENTPOINT.X + DX
Add CURRENTPOINT.X, CURRENTPOINT.Y to end of S.Points attribute list
if ( next character is not a letter )
{
    GOTO label_3
}
else if ( CH == 'H' )
{
    Create new PolyLineSegment S
    Add S to CURRENTPATHFIGURE
}
lable_4:
Read coordinate value X
Let CURRENTPOINT.X = X
Add CURRENTPOINT.X, CURRENTPOINT.Y to end of S.Points attribute list
if ( next character is not a letter )
{
    GOTO label_4
}
else if ( CH == 'v' )
{
    Create new PolyLineSegment S
    Add S to CURRENTPATHFIGURE
}
lable_5:
Read relative coordinate value DY
Let CURRENTPOINT.Y = CURRENTPOINT.Y + DY
Add CURRENTPOINT.X, CURRENTPOINT.Y to end of S.Points attribute list
if ( next character is not a letter )
{
    GOTO label_5
}
else if ( CH == 'V' )
{
    Create new PolyLineSegment S
    Add S to CURRENTPATHFIGURE
}
lable_6:
Read coordinate value Y
Let CURRENTPOINT.Y = Y
Add CURRENTPOINT.X,CURRENTPOINT.Y to end of S.Points attribute list

if ( next character is not a letter )
{
  GOTO label_6
}

else if ( CH == 'c' )
{
  Create new PolyBezierSegment S
  Add S to CURRENTPATHFIGURE

label_7:
  Read relative coordinate pair DX,DY
  Let POINT.X = CURRENTPOINT.X + DX
  Let POINT.Y = CURRENTPOINT.Y + DY
  Add POINT.X, POINT.Y to end of S.Points attribute list
  Read coordinate pair DX,DY
  Let POINT.X = CURRENTPOINT.X + DX
  Let POINT.Y = CURRENTPOINT.Y + DY
  Add POINT.X, POINT.Y to end of S.Points attribute list
  Read coordinate pair DX,DY
  Let CURRENTPOINT.X = CURRENTPOINT.X + DX
  Let CURRENTPOINT.Y = CURRENTPOINT.Y + DY
  Add CURRENTPOINT.X,CURRENTPOINT.Y to end of S.Points attribute list
  if ( next character is not a letter )
  {
    GOTO label_7
  }

else if ( CH == 'C' )
{
  Create new PolyBezierSegment S
  Add S to CURRENTPATHFIGURE

label_8:
  Read coordinate pair X,Y
  Let POINT.X = X
  Let POINT.Y = Y
  Add POINT.X, POINT.Y to end of S.Points attribute list
  Read coordinate pair X,Y
  Let POINT.X = X
  Let POINT.Y = Y
  Add POINT.X, POINT.Y to end of S.Points attribute list
  Read coordinate pair X,Y
  Let CURRENTPOINT.X = X
  Let CURRENTPOINT.Y = Y
  Add CURRENTPOINT.X,CURRENTPOINT.Y to end of S.Points attribute list
  if ( next character is not a letter )
  {
    GOTO label_8
  }

else if ( CH == 'q' )
{
  Create new PolyQuadraticBezierSegment S
  Add S to CURRENTPATHFIGURE

label_9:
  Read relative coordinate pair DX,DY
  Let POINT.X = CURRENTPOINT.X + DX
  Let POINT.Y = CURRENTPOINT.Y + DY
  Add POINT.X, POINT.Y to end of S.Points attribute list
  Read relative coordinate pair DX,DY
  Let CURRENTPOINT.X = CURRENTPOINT.X + DX
  Let CURRENTPOINT.Y = CURRENTPOINT.Y + DY
  Add CURRENTPOINT.X,CURRENTPOINT.Y to end of S.Points attribute list
  if ( next character is not a letter )
  {
    GOTO label_9
  }
}
else ( if CH == 'Q' )
{
    Create new PolyQuadraticBezierSegment S
    Add S to CURRENTPATHFIGURE
}

label_10:
Read coordinate pair X,Y
Let POINT.X = X
Let POINT.Y = Y
Add POINT.X, POINT.Y to end of S.Points attribute list
Read coordinate pair X,Y
Let CURRENTPOINT.X = X
Let CURRENTPOINT.Y = Y
Add CURRENTPOINT.X,CURRENTPOINT.Y to end of S.Points attribute list
if ( next character is not a letter )
{
    GOTO label_10
}
}
el  
else if ( CH == 's' )
{
    Create new PolyBezierSegment S
    Add S to CURRENTPATHFIGURE
label_11:
if ( S.Points is non-empty )
{
    Let LASTCTRLPOINT = Point before last point in S.Points
    Let POINT.X = 2 * CURRENTPOINT.X - LASTCTRLPOINT.X
    Let POINT.Y = 2 * CURRENTPOINT.Y - LASTCTRLPOINT.Y
}
else if ( segment before CURRENTPATHSEGMENT is a PolyBezierSegment )
{
    Let LASTCTRLPOINT = Point before last point in previous PolyBezierSegment
    Let POINT.X = 2 * CURRENTPOINT.X - LASTCTRLPOINT.X
    Let POINT.Y = 2 * CURRENTPOINT.Y - LASTCTRLPOINT.Y
}
else
{
    Let POINT = CURRENTPOINT
}
Add POINT.X, POINT.Y to end of S.Points attribute list
Read relative coordinate pair DX,DY
Let POINT.X = CURRENTPOINT.X + DX
Let POINT.Y = CURRENTPOINT.Y + DY
Add POINT.X, POINT.Y to end of S.Points attribute list
Read relative coordinate pair DX,DY
Let CURRENTPOINT.X = CURRENTPOINT.X + DX
Let CURRENTPOINT.Y = CURRENTPOINT.Y + DY
Add CURRENTPOINT.X,CURRENTPOINT.Y to end of S.Points attribute list
if ( next character is not a letter )
{
    GOTO label_11
}
}
else if ( CH == 'S' )
{
    Create new PolyBezierSegment S
    Add S to CURRENTPATHFIGURE
label_12:
if ( S.Points is non-empty )
{
    Let LASTCTRLPOINT = Point before last point in S.Points
    Let POINT.X = 2 * CURRENTPOINT.X - LASTCTRLPOINT.X
    Let POINT.Y = 2 * CURRENTPOINT.Y - LASTCTRLPOINT.Y
}
else if ( segment before CURRENTPATHSEGMENT is a PolyBezierSegment )
{
    Let LASTCTRLPOINT = S.Point before last point in previous PolyBezierSegment
    Let POINT.X = 2 * CURRENTPOINT.X - LASTCTRLPOINT.X
}

Let \( \text{POINT}.Y = 2 \times \text{CURRENTPOINT}.Y - \text{LASTCTRLPOINT}.Y \)

else

\[
\begin{align*}
&\text{Let POINT} = \text{CURRENTPOINT} \\
&\text{Add POINT}.X, \text{POINT}.Y \text{ to end of S.Points attribute list} \\
&\text{Read coordinate pair } X,Y \\
&\text{Let POINT}.X = X \\
&\text{Let POINT}.Y = Y \\
&\text{Add POINT}.X, \text{POINT}.Y \text{ to end of S.Points attribute list} \\
&\text{Read coordinate pair } X,Y \\
&\text{Let CURRENTPOINT}.X = X \\
&\text{Let CURRENTPOINT}.Y = Y \\
&\text{Add CURRENTPOINT}.X,\text{CURRENTPOINT}.Y \text{ to end of S.Points attribute list} \\
&\text{if ( next character is not a letter )} \\
&\quad \text{GOTO label}_12 \\
&\}
\]

else if ( CH == 'a' )

\[
\begin{align*}
&\text{label}_13: \\
&\quad \text{Create new ArcSegment } S \\
&\quad \text{Add } S \text{ to CURRENTPATHFIGURE} \\
&\quad \text{Read Radius Pair } RX,RY \\
&\quad \text{Read Rotation } ROT \\
&\quad \text{Read integer } FLAG1 \\
&\quad \text{Read integer } FLAG2 \\
&\quad \text{Read relative coordinate pair } DX,DY \\
&\quad \text{Let CURRENTPOINT}.X = \text{CURRENTPOINT}.X + DX \\
&\quad \text{Let CURRENTPOINT}.Y = \text{CURRENTPOINT}.Y + DY \\
&\quad \text{Let S.Point} = \text{CURRENTPOINT}.X,\text{CURRENTPOINT}.Y \\
&\quad \text{Let S.IsLargeArc} = (\text{FLAG1} == 1 ? \text{true} : \text{false}) \\
&\quad \text{Let S.SweepDirection} = (\text{FLAG2} == 1 ? \text{Clockwise} : \text{Counterclockwise}) \\
&\quad \text{Let S.RotationAngle} = ROT \\
&\quad \text{Let S.Size} = RX, RY \\
&\quad \text{if ( next character is not a letter )} \\
&\quad \quad \text{GOTO label}_13 \\
&\quad \}
\]

else if ( CH == 'A' )

\[
\begin{align*}
&\text{label}_14: \\
&\quad \text{Create new ArcSegment } S \\
&\quad \text{Add } S \text{ to CURRENTPATHFIGURE} \\
&\quad \text{Read Radius Pair } RX,RY \\
&\quad \text{Read Rotation } ROT \\
&\quad \text{Read integer } FLAG1 \\
&\quad \text{Read integer } FLAG2 \\
&\quad \text{Read coordinate pair } X,Y \\
&\quad \text{Let CURRENTPOINT}.X = X \\
&\quad \text{Let CURRENTPOINT}.Y = Y \\
&\quad \text{Let } S.\text{Point} = \text{CURRENTPOINT}.X,\text{CURRENTPOINT}.Y \\
&\quad \text{Let } S.\text{IsLargeArc} = (\text{FLAG1} == 1 ? \text{true} : \text{false}) \\
&\quad \text{Let } S.\text{SweepDirection} = (\text{FLAG2} == 1 ? \text{Clockwise} : \text{Counterclockwise}) \\
&\quad \text{Let } S.\text{RotationAngle} = ROT \\
&\quad \text{Let } S.\text{Size} = RX, RY \\
&\quad \text{if ( next character is not a letter )} \\
&\quad \quad \text{GOTO label}_14 \\
&\quad \}
\]

else if ( CH == 'z' or CH == 'Z' )

\[
\begin{align*}
&\quad \text{Let attribute CURRENTPATHFIGURE.IsClosed} = \text{true} \\
&\quad \text{Let CURRENTPOINT} = \text{first point of first segment of CURRENTPATHFIGURE}
\end{align*}
\]
Let CURRENTPATHFIGURE = undefined

/* This case can not occur, because the input is assumed to be well-formed according to the markup schema */
else
    ERROR: Invalid input character
*/

if ( End of input reached )
{
    Terminate algorithm, return PG
}
else
{
    GOTO label_repeat
}

GOTO label_repeat
G. scRGB Gamut Boundary Definition

<?xml version="1.0" encoding="utf-8" ?>
<cdm:ColorDeviceModel
    ID="http://schemas.microsoft.com/windows/2005/02/color/wcsRGB.cdmp"
    xmlns:cdm="http://schemas.microsoft.com/windows/2005/02/color/
            ColorDeviceModel"
    xmlns:wcs="http://schemas.microsoft.com/windows/2005/02/color/
            WcsCommonProfileTypes"
    xmlns:xs='http://www.w3.org/2001/XMLSchema-instance'>
    <cdm:ProfileName>
        wcsRGB virtual device model profile</wcs:Text>
    </cdm:ProfileName>
    <cdm:Description>
        wcsRGB virtual s2.13 device model system profile</wcs:Text>
    </cdm:Description>
    <cdm:Author>
        Microsoft Corporation</wcs:Text>
    </cdm:Author>
    <cdm:MeasurementConditions>
        <cdm:ColorSpace>CIEXYZ</cdm:ColorSpace>
        <cdm:WhitePointName>D65</cdm:WhitePointName>
    </cdm:MeasurementConditions>
    <cdm:SelfLuminous>true</cdm:SelfLuminous>
    <cdm:MaxColorant>4.0</cdm:MaxColorant>
    <cdm:MinColorant>-4.0</cdm:MinColorant>
    <cdm:RGBVirtualDevice>
        <cdm:MeasurementData TimeStamp="2005-02-09T22:00:00">
            <cdm:WhitePrimary X="95.05" Y="100.00" Z="108.90" />
            <cdm:RedPrimary X="41.24" Y="21.26" Z="1.93" />
            <cdm:GreenPrimary X="35.76" Y="71.52" Z="11.92" />
            <cdm:BluePrimary X="18.05" Y="7.22" Z="95.05" />
            <cdm:BlackPrimary X="0" Y="0" Z="0" />
            <cdm:Gamma value="1.0" />
            <cdm:GamutBoundarySamples>
                <cdm:RGB R="0.166433" G="-0.016114" B="0.027918" />
                <cdm:RGB R="0.257134" G="-0.026542" B="0.041950" />
                <cdm:RGB R="0.373396" G="-0.039395" B="0.060310" />
                <cdm:RGB R="0.516590" G="-0.054207" B="0.083665" />
                <cdm:RGB R="0.161885" G="-0.013701" B="0.015224" />
                <cdm:RGB R="0.254873" G="-0.024242" B="0.023134" />
                <cdm:RGB R="0.370591" G="-0.036207" B="0.031873" />
                <cdm:RGB R="0.645317" G="-0.053422" B="0.061566" />
                <cdm:RGB R="0.246052" G="-0.020570" B="0.009804" />
                <cdm:RGB R="0.366951" G="-0.033595" B="0.013350" />
                <cdm:RGB R="0.246052" G="-0.020570" B="0.009804" />
                <cdm:RGB R="0.405159" G="-0.027092" B="0.002697" />
                <cdm:RGB R="0.501415" G="-0.043030" B="0.003038" />
            </cdm:GamutBoundarySamples>
        </cdm:MeasurementData>
    </cdm:RGBVirtualDevice>
</cdm:ColorDeviceModel>
| RGB Value   | 1 | 2 | 3 | 4 | 5 | 6 | 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 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 |
<cdm:RGB R="0.249557" G="-0.030741" B="0.119834" />
<cdm:RGB R="0.357564" G="-0.043900" B="0.171719" />
<cdm:RGB R="0.495437" G="-0.060672" B="0.237960" />
<cdm:RGB R="0.619457" G="-0.065550" B="0.301872" />
<cdm:RGB R="0.000000" G="0.000000" B="0.000000" />
<cdm:RGB R="0.169101" G="-0.018572" B="0.047904" />
<cdm:RGB R="0.255247" G="-0.028467" B="0.071995" />
<cdm:RGB R="0.368932" G="-0.041653" B="0.103698" />
<cdm:RGB R="0.508606" G="-0.056762" B="0.143412" />
<cdm:RGB R="0.637110" G="-0.061275" B="0.186481" />
<cdm:RGB R="1.249149" G="-0.024436" B="-0.002194" />
<cdm:RGB R="0.000475" G="-0.001355" B="0.043728" />
<cdm:RGB R="1.257972" G="-0.026125" B="0.049916" />
<cdm:RGB R="1.264195" G="-0.027432" B="0.104713" />
<cdm:RGB R="0.003291" G="-0.009926" B="0.365422" />
<cdm:RGB R="1.287764" G="-0.033804" B="0.414704" />
<cdm:RGB R="0.004536" G="-0.013745" B="0.512430" />
<cdm:RGB R="1.297183" G="-0.037024" B="0.573653" />
<cdm:RGB R="1.307374" G="-0.040929" B="0.763754" />
<cdm:RGB R="0.007818" G="-0.023805" B="0.902837" />
<cdm:RGB R="-0.001325" G="0.002612" B="-0.000199" />
<cdm:RGB R="-0.005430" G="-0.000728" B="0.906022" />
<cdm:RGB R="-0.020327" G="0.043021" B="-0.002887" />
<cdm:RGB R="-0.042091" G="0.089818" B="-0.005932" />
<cdm:RGB R="-0.117963" G="0.253649" B="-0.016500" />
<cdm:RGB R="-0.141327" G="0.262090" B="0.943868" />
<cdm:RGB R="-0.174730" G="0.376455" B="-0.024386" />
<cdm:RGB R="-0.204641" G="0.390245" B="0.954021" />
<cdm:RGB R="-0.245665" G="0.530007" B="-0.034229" />
<cdm:RGB R="-0.282073" G="0.548898" B="0.962351" />
<cdm:RGB R="1.150086" G="0.568927" B="1.001779" />
<cdm:RGB R="-0.374737" G="0.740509" B="0.968358" />
<cdm:RGB R="1.083883" G="0.766726" B="1.001175" />
<cdm:RGB R="1.002593" G="0.985026" B="-0.073045" />
<cdm:RGB R="-0.434478" G="0.938894" B="-0.060390" />
<cdm:RGB R="0.000896" G="-0.002619" B="0.089419" />
<cdm:RGB R="0.001492" G="-0.004425" B="0.156418" />
<cdm:RGB R="0.002283" G="-0.006839" B="0.247555" />
<cdm:RGB R="0.006038" G="-0.018353" B="0.690852" />
<cdm:RGB R="0.007818" G="-0.023805" B="0.902837" />
<cdm:RGB R="-0.483692" G="0.967403" B="0.971726" />
<cdm:RGB R="0.003728" G="-0.000194" B="0.002807" />
<cdm:RGB R="0.004140" G="-0.000555" B="0.015782" />
<cdm:RGB R="0.004738" G="-0.001328" B="0.043372" />
<cdm:RGB R="0.005489" G="-0.002582" B="0.089155" />
<cdm:RGB R="0.006397" G="-0.004383" B="0.156254" />
<cdm:RGB R="0.007479" G="-0.006795" B="0.247483" />
<cdm:RGB R="0.008758" G="-0.009881" B="0.365433" />
<cdm:RGB R="0.010255" G="-0.013700" B="0.512516" />
<cdm:RGB R="0.011993" G="-0.018308" B="0.691003" />
<cdm:RGB R="0.013995" G="-0.023760" B="0.903048" />
<cdm:RGB R="0.034050" G="-0.018254" B="0.692249" />

Working Draft 1.1, February 2008
<table>
<thead>
<tr>
<th>cdm:RGB R</th>
<th>G</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.023701</td>
<td>0.264902</td>
<td>0.953919</td>
</tr>
<tr>
<td>0.248626</td>
<td>-0.025756</td>
<td>0.924685</td>
</tr>
<tr>
<td>0.246801</td>
<td>-0.020644</td>
<td>0.925712</td>
</tr>
<tr>
<td>0.240032</td>
<td>-0.002034</td>
<td>0.929243</td>
</tr>
<tr>
<td>0.226828</td>
<td>0.033128</td>
<td>0.935165</td>
</tr>
<tr>
<td>0.205577</td>
<td>0.087658</td>
<td>0.942824</td>
</tr>
<tr>
<td>0.174609</td>
<td>0.164417</td>
<td>0.951306</td>
</tr>
<tr>
<td>0.132331</td>
<td>0.266234</td>
<td>0.959693</td>
</tr>
<tr>
<td>0.381162</td>
<td>-0.027864</td>
<td>0.936087</td>
</tr>
<tr>
<td>0.379554</td>
<td>-0.022644</td>
<td>0.936982</td>
</tr>
<tr>
<td>0.313886</td>
<td>0.164903</td>
<td>0.959216</td>
</tr>
<tr>
<td>0.274412</td>
<td>0.267520</td>
<td>0.966469</td>
</tr>
<tr>
<td>0.551062</td>
<td>-0.030854</td>
<td>0.948537</td>
</tr>
<tr>
<td>0.404402</td>
<td>0.399877</td>
<td>0.979322</td>
</tr>
<tr>
<td>0.761768</td>
<td>-0.034778</td>
<td>0.961424</td>
</tr>
<tr>
<td>1.016507</td>
<td>-0.039680</td>
<td>0.974324</td>
</tr>
<tr>
<td>1.247925</td>
<td>-0.024207</td>
<td>-0.006804</td>
</tr>
<tr>
<td>1.318331</td>
<td>-0.045598</td>
<td>0.986957</td>
</tr>
<tr>
<td>1.002593</td>
<td>0.985026</td>
<td>-0.073045</td>
</tr>
<tr>
<td>-0.017084</td>
<td>0.004071</td>
<td>0.105192</td>
</tr>
<tr>
<td>0.003276</td>
<td>-0.000778</td>
<td>0.024704</td>
</tr>
<tr>
<td>0.001189</td>
<td>-0.003497</td>
<td>0.124447</td>
</tr>
<tr>
<td>0.003835</td>
<td>-0.011408</td>
<td>0.425034</td>
</tr>
<tr>
<td>0.006525</td>
<td>-0.019382</td>
<td>0.730759</td>
</tr>
<tr>
<td>0.991838</td>
<td>0.854433</td>
<td>0.996305</td>
</tr>
<tr>
<td>-0.085296</td>
<td>0.980282</td>
<td>0.983420</td>
</tr>
<tr>
<td>0.997109</td>
<td>1.001108</td>
<td>1.001729</td>
</tr>
<tr>
<td>0.000038</td>
<td>-0.000093</td>
<td>0.002143</td>
</tr>
<tr>
<td>0.000087</td>
<td>-0.000230</td>
<td>0.006229</td>
</tr>
<tr>
<td>0.006227</td>
<td>-0.001818</td>
<td>0.062146</td>
</tr>
<tr>
<td>0.005522</td>
<td>-0.016416</td>
<td>0.616931</td>
</tr>
<tr>
<td>0.006525</td>
<td>-0.019382</td>
<td>0.730759</td>
</tr>
<tr>
<td>0.001838</td>
<td>-0.000115</td>
<td>0.001922</td>
</tr>
<tr>
<td>0.003835</td>
<td>-0.011408</td>
<td>0.425034</td>
</tr>
<tr>
<td>0.006525</td>
<td>-0.019382</td>
<td>0.730759</td>
</tr>
<tr>
<td>0.002147</td>
<td>-0.000441</td>
<td>0.013224</td>
</tr>
<tr>
<td>0.002370</td>
<td>-0.000761</td>
<td>0.024478</td>
</tr>
<tr>
<td>0.002634</td>
<td>-0.001207</td>
<td>0.040491</td>
</tr>
<tr>
<td>0.002944</td>
<td>-0.001796</td>
<td>0.062002</td>
</tr>
<tr>
<td>0.003302</td>
<td>-0.002545</td>
<td>0.089732</td>
</tr>
<tr>
<td>0.003715</td>
<td>-0.003473</td>
<td>0.124383</td>
</tr>
<tr>
<td>0.004188</td>
<td>-0.004596</td>
<td>0.166643</td>
</tr>
<tr>
<td>0.004728</td>
<td>-0.005932</td>
<td>0.217198</td>
</tr>
<tr>
<td>0.005521</td>
<td>-0.007497</td>
<td>0.276684</td>
</tr>
<tr>
<td>0.006813</td>
<td>-0.011384</td>
<td>0.425118</td>
</tr>
<tr>
<td>0.008657</td>
<td>-0.016392</td>
<td>0.617056</td>
</tr>
<tr>
<td>0.009736</td>
<td>-0.019358</td>
<td>0.739095</td>
</tr>
<tr>
<td>0.008059</td>
<td>-0.003029</td>
<td>0.066004</td>
</tr>
<tr>
<td>0.005786</td>
<td>-0.003070</td>
<td>0.066004</td>
</tr>
<tr>
<td>0.006041</td>
<td>-0.003495</td>
<td>0.132780</td>
</tr>
<tr>
<td>0.006363</td>
<td>-0.00795</td>
<td>0.24529</td>
</tr>
<tr>
<td>0.006745</td>
<td>-0.001225</td>
<td>0.040547</td>
</tr>
<tr>
<td>0.007184</td>
<td>-0.001801</td>
<td>0.062069</td>
</tr>
<tr>
<td>0.007679</td>
<td>-0.002542</td>
<td>0.089815</td>
</tr>
<tr>
<td>0.008231</td>
<td>-0.003463</td>
<td>0.124487</td>
</tr>
</tbody>
</table>
<cdm:RGB R="0.009523" G="-0.005912" B="0.217344" />
<cdm:RGB R="0.010273" G="-0.007474" B="0.276866" />
<cdm:RGB R="0.012011" G="-0.011356" B="0.425354" />
<cdm:RGB R="0.013011" G="-0.013710" B="0.515596" />
<cdm:RGB R="0.014108" G="-0.016361" B="0.617344" />
<cdm:RGB R="0.015308" G="-0.019327" B="0.731220" />
<cdm:RGB R="0.021793" G="-0.013690" B="0.516185" />
<cdm:RGB R="0.023073" G="-0.016337" B="0.617959" />
<cdm:RGB R="0.024454" G="-0.019298" B="0.731861" />
<cdm:RGB R="0.025943" G="-0.022591" B="0.858499" />
<cdm:RGB R="0.037804" G="-0.019298" B="0.732989" />
<cdm:RGB R="0.039522" G="-0.022583" B="0.859653" />
<cdm:RGB R="0.057988" G="-0.022628" B="0.861439" />
<cdm:RGB R="-0.012314" G="0.982157" B="0.984935" />
<cdm:RGB R="0.020932" G="0.982985" B="0.985609" />
<cdm:RGB R="0.110705" G="0.985131" B="0.987376" />
<cdm:RGB R="0.234831" G="0.987909" B="0.989706" />
<cdm:RGB R="0.311005" G="0.989512" B="0.991076" />
<cdm:RGB R="0.494043" G="0.993078" B="0.994197" />
<cdm:RGB R="0.721366" G="0.997001" B="0.997773" />
<cdm:RGB R="0.852925" G="0.999044" B="0.999710" />
<cdm:RGB R="1.001108" G="1.001729" B="1.001729" />
<cdm:RGB R="0.007645" G="-0.022679" B="0.857328" />
<cdm:RGB R="0.001742" G="-0.000069" B="-0.000016" />
<cdm:RGB R="0.010928" G="-0.022656" B="0.857495" />
<cdm:RGB R="0.005453" G="-0.000169" B="-0.000042" />
<cdm:RGB R="0.016619" G="-0.022623" B="0.857831" />
<cdm:RGB R="-0.071870" G="0.980634" B="0.983705" />
<cdm:RGB R="0.038332" G="-0.000891" B="-0.000240" />
<cdm:RGB R="0.081991" G="-0.022759" B="0.863942" />
<cdm:RGB R="0.168229" G="0.986444" B="0.988479" />
<cdm:RGB R="0.397243" G="0.991242" B="0.992576" />
<cdm:RGB R="0.601915" G="0.995906" B="0.995934" />
<cdm:RGB R="0.721366" G="0.997001" B="0.997773" />
<cdm:RGB R="0.849328" G="0.852650" B="0.994033" />
<cdm:RGB R="0.000000" G="0.000000" B="1.000000" />
</cdm:GamutBoundarySamples>
</cdm:MeasurementData>
</cdm:RGBVirtualDevice>
</cdm:ColorDeviceModel>
H. Standard Namespaces and Content Types

The following tables list the namespaces and content types used in XPS packages and XPS Documents.

### H.1 XML Namespace URIs

*Table H-1. Package-wide namespaces*

<table>
<thead>
<tr>
<th>Description</th>
<th>Namespace URI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content Types</td>
<td><a href="http://schemas.openxmlformats.org/package/2006/content-types">http://schemas.openxmlformats.org/package/2006/content-types</a></td>
</tr>
<tr>
<td>Core Properties</td>
<td><a href="http://schemas.openxmlformats.org/package/2006/metadata/core-properties">http://schemas.openxmlformats.org/package/2006/metadata/core-properties</a></td>
</tr>
<tr>
<td>Relationships</td>
<td><a href="http://schemas.openxmlformats.org/package/2006/relationships">http://schemas.openxmlformats.org/package/2006/relationships</a></td>
</tr>
</tbody>
</table>

*Table H-2. XPS Document namespaces*

<table>
<thead>
<tr>
<th>Description</th>
<th>Namespace URI</th>
</tr>
</thead>
<tbody>
<tr>
<td>DiscardControl</td>
<td><a href="http://schemas.microsoft.com/xps/2005/06/discard-control">http://schemas.microsoft.com/xps/2005/06/discard-control</a></td>
</tr>
<tr>
<td>FixedPage</td>
<td><a href="http://schemas.microsoft.com/xps/2005/06">http://schemas.microsoft.com/xps/2005/06</a></td>
</tr>
<tr>
<td>Print Schema Keywords</td>
<td><a href="http://schemas.microsoft.com/windows/2003/08/printing/print">http://schemas.microsoft.com/windows/2003/08/printing/print</a> schemakeywords</td>
</tr>
<tr>
<td>Signature Definitions</td>
<td><a href="http://schemas.microsoft.com/xps/2005/06/signature-definitions">http://schemas.microsoft.com/xps/2005/06/signature-definitions</a></td>
</tr>
<tr>
<td>Story Fragments</td>
<td><a href="http://schemas.microsoft.com/xps/2005/06/documentstructure">http://schemas.microsoft.com/xps/2005/06/documentstructure</a></td>
</tr>
</tbody>
</table>
H.2 Content Types

The content types in the tables below 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 [M12.7].

<table>
<thead>
<tr>
<th>Table H–3. Package-wide content types</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>Core Properties part</td>
</tr>
<tr>
<td>Digital Signature Certificate part</td>
</tr>
<tr>
<td>Digital Signature Origin part</td>
</tr>
<tr>
<td>Digital Signature XML Signature part</td>
</tr>
<tr>
<td>Relationships part</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table H–4. XPS Document content types</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>FixedDocument</td>
</tr>
<tr>
<td>FixedDocumentSequence</td>
</tr>
<tr>
<td>FixedPage</td>
</tr>
<tr>
<td>DiscardControl</td>
</tr>
<tr>
<td>DocumentStructure</td>
</tr>
<tr>
<td>Font</td>
</tr>
<tr>
<td>ICC profile</td>
</tr>
<tr>
<td>JPEG image</td>
</tr>
<tr>
<td>Obfuscated font</td>
</tr>
<tr>
<td>PNG image</td>
</tr>
<tr>
<td>PrintTicket</td>
</tr>
<tr>
<td>Remote resource dictionary</td>
</tr>
<tr>
<td>StoryFragments</td>
</tr>
<tr>
<td>TIFF image</td>
</tr>
<tr>
<td>Thumbnail part</td>
</tr>
<tr>
<td>Windows Media Photo image</td>
</tr>
</tbody>
</table>
### H.3 Relationship Types

**Table H–5. Package-wide relationship types**

<table>
<thead>
<tr>
<th>Description</th>
<th>Relationship type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core Properties</td>
<td><a href="http://schemas.openxmlformats.org/package/2006/relationships/metadata/core-properties">http://schemas.openxmlformats.org/package/2006/relationships/metadata/core-properties</a></td>
</tr>
<tr>
<td>Thumbnail</td>
<td><a href="http://schemas.openxmlformats.org/package/2006/relationships/metadata/thumbnail">http://schemas.openxmlformats.org/package/2006/relationships/metadata/thumbnail</a></td>
</tr>
</tbody>
</table>

**Table H–6. XPS Document relationship types**

<table>
<thead>
<tr>
<th>Description</th>
<th>Relationship type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Digital Signature Definitions</td>
<td><a href="http://schemas.microsoft.com/xps/2005/06/signature-definitions">http://schemas.microsoft.com/xps/2005/06/signature-definitions</a></td>
</tr>
<tr>
<td>DiscardControl</td>
<td><a href="http://schemas.microsoft.com/xps/2005/06/discard-control">http://schemas.microsoft.com/xps/2005/06/discard-control</a></td>
</tr>
<tr>
<td>PrintTicket</td>
<td><a href="http://schemas.microsoft.com/xps/2005/06/printticket">http://schemas.microsoft.com/xps/2005/06/printticket</a></td>
</tr>
<tr>
<td>Restricted Font</td>
<td><a href="http://schemas.microsoft.com/xps/2005/06/restricted-font">http://schemas.microsoft.com/xps/2005/06/restricted-font</a></td>
</tr>
<tr>
<td>StoryFragments</td>
<td><a href="http://schemas.microsoft.com/xps/2005/06/storyfragments">http://schemas.microsoft.com/xps/2005/06/storyfragments</a></td>
</tr>
</tbody>
</table>
I. Conformance Requirements

This annex is informative

This annex summarizes all conformance requirements for producers and consumers implementing the XML Paper Specification. 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 below, respectively containing the 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 numberthe clause in which the requirement first appears, 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 may use these IDs to report error conditions.

Additionally, each the 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 “×” 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 “×”. 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 they 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:

<table>
<thead>
<tr>
<th>Superscript</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Only applies to producers or consumers implementing digital signature features.</td>
</tr>
<tr>
<td>C</td>
<td>Only applies if Core Properties are used.</td>
</tr>
<tr>
<td>D</td>
<td>Only applies to producers that generate document structure or consumers that use it.</td>
</tr>
<tr>
<td>E</td>
<td>Only applies to editing consumers that are also producers</td>
</tr>
<tr>
<td>F</td>
<td>Only necessary if the referenced feature is used.</td>
</tr>
<tr>
<td>ID</td>
<td>Rule</td>
</tr>
<tr>
<td>----</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>H</td>
<td>Only applies to producers that generate hyperlinks or consumers that implement hyperlinks.</td>
</tr>
<tr>
<td>K</td>
<td>Only applies to consumers that consume the XPS Document package head-first.</td>
</tr>
<tr>
<td>P</td>
<td>Only applies to printing consumers.</td>
</tr>
<tr>
<td>R</td>
<td>Only applies to producers or consumers that use DiscardControl parts.</td>
</tr>
<tr>
<td>S</td>
<td>Only applies to consumers that implement selection.</td>
</tr>
<tr>
<td>T</td>
<td>Only applies if thumbnails are used by the consumer.</td>
</tr>
<tr>
<td>U</td>
<td>Only required for a producer if the part is generated or for a consumer when accessing the part.</td>
</tr>
</tbody>
</table>

**I.1 XPS Document Format**

**I.1.1 MUST Conformance Requirements**

<table>
<thead>
<tr>
<th>ID</th>
<th>Rule</th>
<th>Reference</th>
<th>Producer</th>
<th>Consumer</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1.1</td>
<td>XPS Documents MUST observe all conformance requirements of the OPC specification, except where specifically noted otherwise in this specification.</td>
<td>Clause 8</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>M1.2</td>
<td>The XPS Document format MUST use a ZIP archive for its physical model.</td>
<td>8.2</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>

**I.2 Parts and Relationships**

**I.2.1 MUST Conformance Requirements**

<table>
<thead>
<tr>
<th>ID</th>
<th>Rule</th>
<th>Reference</th>
<th>Producer</th>
<th>Consumer</th>
</tr>
</thead>
<tbody>
<tr>
<td>M2.1</td>
<td>All content to be rendered MUST be contained in the XPS Document.</td>
<td>9.1</td>
<td>x</td>
<td>v</td>
</tr>
<tr>
<td>M2.2</td>
<td>Each part contained in an XPS Document MUST use only the appropriate content type.</td>
<td>9.1</td>
<td>x</td>
<td>v[^1]</td>
</tr>
<tr>
<td>M2.3</td>
<td>An XPS Document MUST contain exactly one FixedDocumentSequence part per fixed payload.</td>
<td>9.1</td>
<td>x</td>
<td>v</td>
</tr>
<tr>
<td>ID</td>
<td>Rule</td>
<td>Reference</td>
<td>Producer</td>
<td>Consumer</td>
</tr>
<tr>
<td>------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------</td>
<td>----------</td>
<td>----------</td>
</tr>
<tr>
<td>M2.4</td>
<td>An XPS Document MUST contain at least one FixedDocument part per fixed payload.</td>
<td>9.1</td>
<td>x</td>
<td>v</td>
</tr>
<tr>
<td>M2.5</td>
<td>An XPS Document MUST contain at least one FixedPage part per fixed payload.</td>
<td>9.1</td>
<td>x</td>
<td>v</td>
</tr>
<tr>
<td>M2.6</td>
<td>A <code>&lt;Glyphs&gt;</code> element in FixedPage markup MUST reference a Font part that exists in the XPS Document.</td>
<td>9.1</td>
<td>x</td>
<td>v</td>
</tr>
<tr>
<td>M2.7</td>
<td>An <code>&lt;ImageBrush&gt;</code> element in FixedPage markup MUST reference an Image part that exists in the XPS Document.</td>
<td>9.1</td>
<td>x</td>
<td>v</td>
</tr>
<tr>
<td>M2.8</td>
<td>If FixedPage markup references a Remote Resource Dictionary part, it MUST be included in the XPS Document.</td>
<td>9.1</td>
<td>x</td>
<td>v</td>
</tr>
<tr>
<td>M2.10</td>
<td>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.</td>
<td>9.1.1</td>
<td>x</td>
<td>v</td>
</tr>
<tr>
<td>M2.12</td>
<td>A Restricted Font relationship is REQUIRED for each print and preview font used, from the FixedDocument part to the preview and print Font part. When invoking editing functionality, an editing consumer 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 and display-only consumers MUST consider an XPS Document valid, even if the producer failed to properly set the Restricted Font relationship.</td>
<td>9.1.1, 9.1.7.2, 9.1.7.4</td>
<td>x</td>
<td>v^E</td>
</tr>
<tr>
<td>M2.13</td>
<td>Exactly one StartPart relationship is REQUIRED.</td>
<td>9.1.1</td>
<td>x</td>
<td>v</td>
</tr>
<tr>
<td>M2.14</td>
<td>The StartPart relationship MUST point from the package to the FixedDocumentSequence part that is the primary fixed payload root.</td>
<td>9.1.2</td>
<td>x</td>
<td>v</td>
</tr>
<tr>
<td>M2.15</td>
<td>The order of <code>&lt;DocumentReference&gt;</code> elements in a FixedDocumentSequence part MUST be preserved by editing consumers that are also producers.</td>
<td>9.1.2</td>
<td>x^E</td>
<td></td>
</tr>
<tr>
<td>M2.16</td>
<td>The order of <code>&lt;PageContent&gt;</code> elements in a FixedDocument MUST be preserved by editing consumers that are also producers.</td>
<td>9.1.3</td>
<td>x^E</td>
<td></td>
</tr>
<tr>
<td>M2.17</td>
<td>JPEG image parts MUST contain images that conform to the JPEG specification.</td>
<td>9.1.5.1</td>
<td>x</td>
<td>v^U</td>
</tr>
<tr>
<td>M2.18</td>
<td>PNG image parts MUST contain images that conform to the PNG specification.</td>
<td>9.1.5.2</td>
<td>x</td>
<td>v^U</td>
</tr>
<tr>
<td>M2.19</td>
<td>The PNG ancillary chunk tRNS MUST be supported.</td>
<td>9.1.5.2</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>M2.20</td>
<td>The PNG ancillary chunk iCCP MUST be supported.</td>
<td>9.1.5.2</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>M2.21</td>
<td>The PNG ancillary chunk sRGB MUST be ignored.</td>
<td>9.1.5.2</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>ID</td>
<td>Rule</td>
<td>Reference</td>
<td>Producer</td>
<td>Consumer</td>
</tr>
<tr>
<td>-------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------</td>
<td>----------</td>
<td>----------</td>
</tr>
<tr>
<td>M2.22</td>
<td>The PNG ancillary chunk cHRM MUST be ignored.</td>
<td>9.1.5.2</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>M2.23</td>
<td>The PNG ancillary chunk gAMA MUST be ignored.</td>
<td>9.1.5.2</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>M2.24</td>
<td>The PNG ancillary chunk sBIT MUST be ignored.</td>
<td>9.1.5.2</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>M2.25</td>
<td>TIFF image parts MUST contain images that conform to the TIFF specification</td>
<td>9.1.5.3</td>
<td>×</td>
<td>v</td>
</tr>
<tr>
<td>M2.26</td>
<td>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.</td>
<td>9.1.5.3</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>M2.27</td>
<td>If a TIFF file contains multiple image file directories (IFDs), consumers MUST use only the first IFD and ignore all others.</td>
<td>9.1.5.3</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>M2.28</td>
<td>XPS Document consumers MUST support TIFF images using CCITT bilevel encoding.</td>
<td>9.1.5.3</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>M2.29</td>
<td>XPS Document consumers MUST support CMYK TIFF images.</td>
<td>9.1.5.3</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>M2.30</td>
<td>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.</td>
<td>9.1.5.3</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>M2.31</td>
<td>XPS Document consumers MUST support TIFF images using LZW compression.</td>
<td>9.1.5.3</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>M2.32</td>
<td>XPS Document consumers MUST support TIFF images using differencing predictors.</td>
<td>9.1.5.3</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>M2.33</td>
<td>XPS Document consumers MUST support TIFF images using JPEG compression (compression mode 6 only).</td>
<td>9.1.5.3</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>M2.34</td>
<td>XPS Document consumers MUST support TIFF images with an embedded ICC profile.</td>
<td>9.1.5.3</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>M2.35</td>
<td>Windows Media Photo image files MUST conform to the Windows Media Photo specification.</td>
<td>9.1.5.4</td>
<td>×</td>
<td>v</td>
</tr>
<tr>
<td>M2.36</td>
<td>Each FixedPage part MUST NOT have more than one thumbnail part attached.</td>
<td>9.1.6</td>
<td>×</td>
<td>v</td>
</tr>
<tr>
<td>M2.37</td>
<td>Thumbnails MUST be either JPEG or PNG images</td>
<td>9.1.6</td>
<td>×</td>
<td>v</td>
</tr>
<tr>
<td>M2.38</td>
<td>If using a fragment in the FontURI attribute of the &lt;Glyphs&gt; 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.</td>
<td>9.1.7</td>
<td>×</td>
<td>v</td>
</tr>
<tr>
<td>M2.39</td>
<td>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.</td>
<td>9.1.7.1</td>
<td>×</td>
<td>v</td>
</tr>
<tr>
<td>M2.40</td>
<td>Producers MUST honor the licensing rights specified in OpenType fonts by following the embedding and obfuscation mechanisms.</td>
<td>9.1.7.2</td>
<td>×</td>
<td></td>
</tr>
</tbody>
</table>
### I. Conformance Requirements

<table>
<thead>
<tr>
<th>ID</th>
<th>Rule</th>
<th>Reference</th>
<th>Producer</th>
<th>Consumer</th>
</tr>
</thead>
<tbody>
<tr>
<td>M2.41</td>
<td>Consumers MUST be able to process XPS Documents using any combination of the embedding and obfuscation mechanisms described in this specification (even if produced in violation of the production requirements).</td>
<td>9.1.7.2</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>M2.42</td>
<td>For fonts with “Restricted license embedding” licensing intent, producers MUST NOT embed the font.</td>
<td>9.1.7.2</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>M2.43</td>
<td>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.</td>
<td>9.1.7.2</td>
<td>v</td>
<td>x</td>
</tr>
<tr>
<td>M2.44</td>
<td>For fonts with “Print and preview embedding” licensing intent, producers MUST perform embedded font obfuscation.</td>
<td>9.1.7.2</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>M2.45</td>
<td>For fonts with “Print and preview embedding” licensing intent, consumers MUST NOT extract or permanently install the font.</td>
<td>9.1.7.2</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>M2.46</td>
<td>For fonts with “Editable embedding” licensing intent, producers MUST perform embedded font obfuscation.</td>
<td>9.1.7.2</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>M2.47</td>
<td>For fonts with “Editable embedding” licensing intent, consumers MUST NOT extract or permanently install the font.</td>
<td>9.1.7.2</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>M2.48</td>
<td>For fonts with “No subsetting” licensing intent, producers MUST perform embedded font obfuscation.</td>
<td>9.1.7.2</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>M2.49</td>
<td>For fonts with “No subsetting” licensing intent, producers MUST NOT subset the font.</td>
<td>9.1.7.2</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>M2.50</td>
<td>For fonts with “No subsetting” licensing intent, consumers MUST NOT extract or permanently install the font.</td>
<td>9.1.7.2</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>M2.51</td>
<td>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.</td>
<td>9.1.7.2</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>M2.52</td>
<td>For fonts with “Bitmap embedding only” licensing intent, consumers MUST NOT extract or permanently install the font.</td>
<td>9.1.7.2</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>M2.53</td>
<td>Producers and consumers MUST perform font obfuscation and de-obfuscation according to the steps described in §9.1.7.3, “Embedded Font Obfuscation.”</td>
<td>9.1.7.3</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>M2.54</td>
<td>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.</td>
<td>9.1.7.3</td>
<td>x</td>
<td>v</td>
</tr>
<tr>
<td>M2.55</td>
<td>When processing &lt;Glyphs&gt; elements, the consumer MUST select a cmap table from the OpenType font according to Table 1-8 in §9.1.7.5, “Non-Standard Font Compatibility Encoding.” All further processing for that font MUST use the selected cmap table.</td>
<td>9.1.7.5</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>ID</td>
<td>Rule</td>
<td>Reference</td>
<td>Producer</td>
<td>Consumer</td>
</tr>
<tr>
<td>-----</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------</td>
<td>----------</td>
<td>----------</td>
</tr>
<tr>
<td>M2.56</td>
<td>When processing <code>&lt;Glyphs&gt;</code> elements, if a Wansung, Big5, Prc, ShiftJis, or MacRoman cmap has been selected, the consumer MUST correctly map from Unicode codepoints in the <code>UnicodeString</code> attribute to the corresponding codepoints used by the cmap before looking up glyphs.</td>
<td>9.1.7.5</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>M2.57</td>
<td>When processing <code>&lt;Glyphs&gt;</code> elements that reference a cmap (3,0) encoding font, consumers MUST handle the case where the <code>UnicodeString</code> attribute contains character codes instead of PUA codepoints by computing the correct glyph index according to the general recommendations of the OpenType specification.</td>
<td>9.1.7.5</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>M2.58</td>
<td>Consumers MUST process all <code>PrintTicket</code> parts when an XPS Document is printed.</td>
<td>9.1.9</td>
<td>x&lt;sup&gt;b&lt;/sup&gt;</td>
<td></td>
</tr>
<tr>
<td>M2.59</td>
<td>A level-specific <code>PrintTicket</code> MUST contain only settings scoped to the current level and child levels. Job-level <code>PrintTicket</code> parts MUST contain only job-, document-, and page-scoped settings; document-level <code>PrintTicket</code> parts MUST contain only document-scoped and page-scoped settings; and page-level <code>PrintTicket</code> 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 <code>FixedDocumentSequence</code>, <code>FixedDocument</code>, or <code>FixedPage</code> part MUST have no more than one attached <code>PrintTicket</code>.</td>
<td>9.1.9.2</td>
<td>x&lt;sup&gt;U&lt;/sup&gt;</td>
<td></td>
</tr>
<tr>
<td>M2.60</td>
<td>Consumers MUST process job-level, document-level and page-level settings of <code>PrintTicket</code> parts associated with <code>FixedDocumentSequence</code> parts.</td>
<td>9.1.9.2</td>
<td>x&lt;sup&gt;b&lt;/sup&gt;</td>
<td></td>
</tr>
<tr>
<td>M2.61</td>
<td>Consumers MUST process document-level and page-level settings of <code>PrintTicket</code> parts associated with <code>FixedDocument</code> parts and MUST ignore job-level settings of <code>PrintTicket</code> parts associated with <code>FixedDocument</code> parts.</td>
<td>9.1.9.2</td>
<td>x&lt;sup&gt;b&lt;/sup&gt;</td>
<td></td>
</tr>
<tr>
<td>M2.62</td>
<td>Consumers MUST process page-level settings of <code>PrintTicket</code> parts associated with <code>FixedPage</code> parts and MUST ignore job-level and document-level settings of <code>PrintTicket</code> parts associated with <code>FixedPage</code> parts.</td>
<td>9.1.9.2</td>
<td>x&lt;sup&gt;b&lt;/sup&gt;</td>
<td></td>
</tr>
<tr>
<td>M2.63</td>
<td>When processing a <code>PrintTicket</code>, consumers MUST first remove all levels of <code>PrintTicket</code> content not applicable to the current element.</td>
<td>9.1.9.3</td>
<td>x&lt;sup&gt;b&lt;/sup&gt;</td>
<td></td>
</tr>
<tr>
<td>M2.64</td>
<td>When processing a <code>PrintTicket</code>, consumers MUST second validate the <code>PrintTicket</code> according to the methods defined in the <code>PrintTicket Validation Checklist</code> of the <code>Print Schema documentation</code>.</td>
<td>9.1.9.3</td>
<td>x&lt;sup&gt;b&lt;/sup&gt;</td>
<td></td>
</tr>
<tr>
<td>M2.65</td>
<td>Following validation of a <code>PrintTicket</code>, the printing consumer MUST properly interpret the print settings according to the rules for merging two <code>PrintTicket</code> parts.</td>
<td>9.1.9.3</td>
<td>x&lt;sup&gt;b&lt;/sup&gt;</td>
<td></td>
</tr>
<tr>
<td>M2.66</td>
<td>If there is no print setting merge conflict between different <code>PrintTicket</code> 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</td>
<td>9.1.9.3</td>
<td>x&lt;sup&gt;b&lt;/sup&gt;</td>
<td></td>
</tr>
</tbody>
</table>
## Conformance Requirements

<table>
<thead>
<tr>
<th>ID</th>
<th>Rule</th>
<th>Reference</th>
<th>Producer</th>
<th>Consumer</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>identical element.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M2.67</td>
<td>If there is a print setting merge conflict between different PrintTicket levels, the setting from the most specific ticket MUST take precedence.</td>
<td>9.1.9.3</td>
<td>P</td>
<td></td>
</tr>
<tr>
<td>M2.68</td>
<td>Consumers MUST use semantic document structure provided in included DocumentStructure and StoryFragments parts in preference to any other analysis method of generating such structure.</td>
<td>9.1.11</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>M2.69</td>
<td>Consumers MUST support Markup Compatibility and Extensibility elements and attributes in DocumentStructure, FixedDocument, FixedDocumentSequence, FixedPage, Relationships, Remote Resource Dictionary, SignatureDefinitions, and StoryFragments parts. Before attempting to validate one of these parts against a schema, consumers MUST remove all Markup Compatibility and Extensibility elements and attributes, ignorable namespace declarations, and all ignored elements and attributes not defined in the expected version of XPS Document markup.</td>
<td>9.3.1</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>M2.70</td>
<td>XML content MUST be encoded using either UTF-8 or UTF-16. If any such part includes an encoding declaration (as defined in §4.3.3 of the XML specification), that declaration MUST NOT name any encoding other than UTF-8 or UTF-16.</td>
<td>9.3.2</td>
<td>x</td>
<td>v</td>
</tr>
<tr>
<td>M2.71</td>
<td>DTD content MUST NOT be used in the XML markup defined in this specification, and consumers MUST treat the presence of DTD content as an error.</td>
<td>9.3.2</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>M2.72</td>
<td>XML content MUST be valid against the corresponding W3C XSD schema defined in this specification. In particular, the XML content MUST NOT contain elements or attributes drawn from namespaces that are not explicitly defined in the corresponding XSD unless the XSD allows elements or attributes drawn from any namespace to be present in particular locations in the XML markup.</td>
<td>9.3.2</td>
<td>x</td>
<td>v</td>
</tr>
<tr>
<td>M2.73</td>
<td>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 specification.</td>
<td>9.3.2</td>
<td>x</td>
<td>v</td>
</tr>
<tr>
<td>M2.74</td>
<td>Properties MUST NOT be set more than once, regardless of the syntax used to specify the value. In certain cases, they can be specified using either property attributes or property elements. Consumers MUST treat properties that are specified in both ways as an error.</td>
<td>9.3.3.2</td>
<td>x</td>
<td>v</td>
</tr>
<tr>
<td>M2.75</td>
<td>XPS Document markup MUST NOT use the xml:space attribute.</td>
<td>9.3.4</td>
<td>x</td>
<td>v</td>
</tr>
<tr>
<td>M2.76</td>
<td>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”</td>
<td>9.3.5.1</td>
<td>x</td>
<td></td>
</tr>
</tbody>
</table>
### 1. Conformance Requirements

#### Table 1–3. Parts and Relationships SHOULD conformance requirements

<table>
<thead>
<tr>
<th>ID</th>
<th>Rule</th>
<th>Reference</th>
<th>Producer</th>
<th>Consumer</th>
</tr>
</thead>
<tbody>
<tr>
<td>S2.1</td>
<td>It is RECOMMENDED that there be exactly one Required Resource relationship from the FixedPage part for each resource referenced from the markup.</td>
<td>9.1.1</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>S2.2</td>
<td>Each <code>&lt;DocumentReference&gt;</code> element in a FixedDocumentSequence part SHOULD reference a FixedDocument part by relative URI.</td>
<td>9.1.2</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>S2.3</td>
<td>Each <code>&lt;PageContent&gt;</code> element in a FixedDocument part MUST reference a FixedPage part by relative URI.</td>
<td>9.1.3</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>S2.4</td>
<td><code>&lt;ImageBrush&gt;</code> and <code>&lt;Glyphs&gt;</code> elements MUST reference Image and Font parts by relative URI.</td>
<td>9.1.4</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>S2.5</td>
<td>Color profiles embedded in image files SHOULD be used if present and compatible with this specification.</td>
<td>9.1.5</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>S2.6</td>
<td>It is RECOMMENDED that JPEG image part names end with the extension “.jpg”.</td>
<td>9.1.5.1</td>
<td>x</td>
<td></td>
</tr>
</tbody>
</table>

---

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

**M2.78** Consumers MUST support JPEG images that contain the APP1 marker and interpret the EXIF color space correctly.

**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.

---

**M2.80** Each `<DocumentReference>` element in a FixedDocumentSequence part MUST reference a FixedDocument part by relative URI.

**M2.81** Each `<PageContent>` element in a FixedDocument part SHOULD reference a FixedPage part by relative URI.

**M2.82** `<ImageBrush>` and `<Glyphs>` elements SHOULD reference Image and Font parts by relative URI.

---

1 **I.2.2** SHOULD Conformance Requirements

2 Table 1–3. Parts and Relationships SHOULD conformance requirements
<table>
<thead>
<tr>
<th>ID</th>
<th>Rule</th>
<th>Reference</th>
<th>Producer</th>
<th>Consumer</th>
</tr>
</thead>
<tbody>
<tr>
<td>S2.7</td>
<td>The use of CMYK JPEG images is NOT RECOMMENDED. TIFF or Windows Media Photo images SHOULD be used instead to represent CMYK images.</td>
<td>9.1.5.1</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>S2.8</td>
<td>It is RECOMMENDED that PNG image part names end with the extension “.png”.</td>
<td>9.1.5.2</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>S2.9</td>
<td>It is RECOMMENDED that TIFF image part names end with the extension “.tif”.</td>
<td>0</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>S2.10</td>
<td>Consumers SHOULD ignore unsupported TIFF tags (those not described in Table 9–5 and §9.1.5.3, &quot;If no color profile is embedded in the PNG image or stored in a separate part associated with the PNG image according to the mechanisms described in §15.2.8, then the default color space MUST be treated as defined in §15.2.9 [M8.30]. TIFF Images). Producers SHOULD NOT include unsupported tags.</td>
<td>0</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>S2.11</td>
<td>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.</td>
<td>9.1.5.3</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>S2.12</td>
<td>It is RECOMMENDED that Windows Media Photo images end with the extension “.wdp”.</td>
<td>9.1.5.4</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>S2.13</td>
<td>It is RECOMMENDED that if thumbnails are used for pages, a thumbnail SHOULD be included for every page in the document.</td>
<td>9.1.6</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>S2.14</td>
<td>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.</td>
<td>9.1.6</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>S2.15</td>
<td>Producers SHOULD use Unicode-encoded fonts.</td>
<td>9.1.7,</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>9.1.7.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S2.16</td>
<td>For fonts with &quot;Installable embedding&quot; licensing intent, producers SHOULD perform embedded font obfuscation.</td>
<td>9.1.7.2</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>S2.17</td>
<td>For fonts with &quot;Installable embedding” licensing intent, consumers SHOULD NOT extract or permanently install the font.</td>
<td>9.1.7.2</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>S2.18</td>
<td>For fonts with &quot;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 &lt;Path&gt; element.</td>
<td>9.1.7.2</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>S2.19</td>
<td>Although the licensing intent allows embedding of non-</td>
<td>9.1.7.3</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
### I. Conformance Requirements

<table>
<thead>
<tr>
<th>ID</th>
<th>Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>S2.20</td>
<td>It is RECOMMENDED that the extension of an obfuscated Font part name be &quot;odttf&quot; for TrueType fonts and &quot;odttc&quot; for TrueType collections.</td>
</tr>
<tr>
<td>S2.21</td>
<td>Producers SHOULD include only PrintTicket settings that support portability of the XPS Document.</td>
</tr>
<tr>
<td>S2.22</td>
<td>Producers SHOULD only attach PrintTicket parts containing only document-level and page-level settings with FixedDocument parts.</td>
</tr>
<tr>
<td>S2.23</td>
<td>Producers SHOULD only attach PrintTicket parts containing only page-level settings with FixedPage parts.</td>
</tr>
<tr>
<td>S2.24</td>
<td>The FixedDocumentSequence part SHOULD follow the part name recommendation &quot;/&lt;FixedDocSeq&gt;.fdseq&quot; where &lt;FixedDocSeq&gt; is the name of the FixedDocumentSequence. The FixedDocumentSequence SHOULD use the extension &quot;fdseq&quot;.</td>
</tr>
<tr>
<td>S2.25</td>
<td>A FixedDocument part SHOULD follow the part name recommendation &quot;/Documents/&lt;n&gt;/FixedDocument=&lt;FixedDocument&gt;.fdoc&quot; where &lt;n&gt; is a numeral that indicates the ordinal position of the fixed document in the fixed document sequence and &lt;FixedDocument&gt; is the name of the fixed document. FixedDocument parts SHOULD use the extension &quot;.fdoc&quot;.</td>
</tr>
<tr>
<td>S2.26</td>
<td>A FixedPage part SHOULD follow the part name recommendation &quot;/Documents/&lt;n&gt;/Pages/&lt;m&gt;.fpage&quot; where &lt;n&gt; represents the document that includes this page and &lt;m&gt; is the page number. FixedPage parts SHOULD use the extension &quot;.fpage&quot;.</td>
</tr>
<tr>
<td>S2.27</td>
<td>A resource that is specific to a particular document SHOULD follow the part name recommendation &quot;/Documents/&lt;n&gt;/Resources/&lt;Resource&gt;&quot; where &lt;n&gt; is the document that uses the resource and &lt;Resource&gt; is the segments identifying the particular resource. A resource that is shared across multiple documents SHOULD follow the part name recommendation &quot;/Resources/&lt;Resource&gt;&quot;. A Font resource SHOULD use &quot;Fonts/&lt;FontName&gt;.&lt;FontExt&gt;&quot; for its &lt;Resource&gt; value, where &lt;FontExt&gt; SHOULD be either &quot;.ttf&quot; or &quot;.odttf&quot; for non-obfuscated and obfuscated fonts respectively. An Image resource SHOULD use &quot;Images/&lt;ImageName&gt;.&lt;ImageExt&gt;&quot; for its &lt;Resource&gt; value, where &lt;ImageExt&gt; SHOULD be either &quot;.jpg&quot;, &quot;.png&quot;, or other widely supported image formats.</td>
</tr>
</tbody>
</table>

---

416 Working Draft 1.1, February 2008
<table>
<thead>
<tr>
<th>ID</th>
<th>Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>“Images/&lt;ImageName&gt;.&lt;ImageExt&gt;” for its &lt;Resource&gt; value, where &lt;ImageExt&gt; is the correct extension for the image type. A Remote Resource Dictionary resource SHOULD use “Dictionaries/&lt;DictName&gt;.dict” for its &lt;Resource&gt; value. A Remote Resource Dictionary SHOULD use “.dict” as its extension. &lt;fontName&gt;, &lt;imageName&gt;, and &lt;dictname&gt; SHOULD be a string representation of a GUID value if the resource is a shared resource.</td>
</tr>
<tr>
<td>S2.28</td>
<td>A DocumentStructure part SHOULD follow the part name recommendation “/Documents/&lt;n&gt;/Structure/&lt;DocStruct&gt;.struct” where &lt;n&gt; is the fixed document that the document structure applies to. DocumentStructure parts SHOULD use the extension “.struct”.</td>
</tr>
<tr>
<td>S2.29</td>
<td>A StoryFragments part SHOULD follow the part name recommendation “/Documents/&lt;n&gt;/Structure/Fragments/&lt;m&gt;.frag” where &lt;n&gt; is the fixed document that contains the story fragments and &lt;m&gt; is the page number the StoryFragments part applies to. StoryFragments parts SHOULD use the extension “.frag”.</td>
</tr>
<tr>
<td>S2.30</td>
<td>ICC profile parts SHOULD follow the part name recommendation “/Documents/&lt;n&gt;/Metadata/&lt;ProfileName&gt;.&lt;ProfileExt&gt;” where &lt;n&gt; is the fixed document that contains the profile. ICC profile parts shared across multiple documents SHOULD follow the part name recommendation “/Metadata/&lt;ProfileName&gt;.&lt;ProfileExt&gt;”. In this case, &lt;ProfileName&gt; SHOULD be a string representation of a GUID value. The &lt;ProfileExt&gt; SHOULD be appropriate to the color profile type, such as “.icm”.</td>
</tr>
<tr>
<td>S2.31</td>
<td>Thumbnail parts SHOULD follow the part name recommendation “/Documents/&lt;n&gt;/Metadata/&lt;ThumbName&gt;.&lt;ThumbExt&gt;” where &lt;n&gt; is the fixed document that contains the profile. If the Thumbnail part represents the package as a whole, it SHOULD follow the part name recommendation “/Metadata/&lt;ThumbName&gt;.&lt;ThumbExt&gt;”. In this case, &lt;ThumbName&gt; SHOULD be a string representation of a GUID value. The &lt;ThumbExt&gt; SHOULD be appropriate to the image type,</td>
</tr>
</tbody>
</table>
### I. Conformance Requirements

<table>
<thead>
<tr>
<th>ID</th>
<th>Rule</th>
<th>Reference</th>
<th>Producer</th>
<th>Consumer</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>9.2</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>S2.32</td>
<td>PrintTicket part names associated with the entire job SHOULD be associated via relationship with the FixedDocumentSequence part and follow the part name recommendation “/Metadata/&lt;PrintTicketName&gt;.xml”. PrintTicket parts associated with a particular fixed document or fixed page SHOULD follow the part name recommendation “/Documents/&lt;n&gt;/Metadata/&lt;PrintTicketName&gt;.xml”, where &lt;n&gt; is the fixed document that uses these parts. PrintTicket parts SHOULD use the extension “.xml”.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S2.33</td>
<td>The names of any non-standard parts that are associated with a particular fixed document SHOULD follow the part name recommendation “/Documents/&lt;n&gt;/Other/&lt;PartName&gt;”, where &lt;n&gt; is the fixed document to which the part belongs.</td>
<td>9.2</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>S2.34</td>
<td>Consumers SHOULD support JPEG images that contain JFIF-specified APP0 and ICC-specified APP2 markers APP0, APP2, APP13, and APP14 markers.</td>
<td>9.1.5.1</td>
<td>x</td>
<td></td>
</tr>
</tbody>
</table>

### I.2.3 OPTIONAL Conformance Requirements

#### Table I–4. Parts and Relationships OPTIONAL conformance requirements

<table>
<thead>
<tr>
<th>ID</th>
<th>Rule</th>
<th>Reference</th>
<th>Producer</th>
<th>Consumer</th>
</tr>
</thead>
<tbody>
<tr>
<td>O2.1</td>
<td>Thumbnail parts MAY be included in an XPS Document</td>
<td>9.1</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>O2.2</td>
<td>PrintTicket parts MAY be included in an XPS Document.</td>
<td>9.1</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>O2.3</td>
<td>ICC Profile parts MAY be included in an XPS Document.</td>
<td>9.1</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>O2.4</td>
<td>DocumentStructure parts MAY be included in an XPS Document.</td>
<td>9.1</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>O2.5</td>
<td>StoryFragments parts MAY be included in an XPS Document.</td>
<td>9.1</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>O2.6</td>
<td>SignatureDefinitions parts MAY be included in an XPS Document.</td>
<td>9.1</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>O2.7</td>
<td>DiscardControl parts MAY be included in an XPS Document.</td>
<td>9.1</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>O2.8</td>
<td>A Core Properties relationship MAY be included in an XPS Document, from the package to the Core Properties part.</td>
<td>9.1.1</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>O2.9</td>
<td>A Digital Signatures Origin relationship MAY be included in an XPS Document, from the package to the Digital Signature Origin part.</td>
<td>9.1.1</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Requirement</td>
<td>Description</td>
<td>Conformance Requirement</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------</td>
<td>-------------</td>
<td>-------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>O2.10</td>
<td>Digital Signature relationships MAY be included in an XPS Document, from the Digital Signature Origin part to a Digital Signature XML Signature part.</td>
<td>9.1.1 ×</td>
<td></td>
<td></td>
</tr>
<tr>
<td>O2.11</td>
<td>Digital Signature Certificate relationships MAY be included in an XPS Document, from a Digital Signature XML Signature part to the Digital Signature Certificate part.</td>
<td>9.1.1 ×</td>
<td></td>
<td></td>
</tr>
<tr>
<td>O2.12</td>
<td>Digital Signature Definitions parts MAY be included in an XPS Document, from a FixedDocument part to the Digital Signature Definitions part.</td>
<td>9.1.1 ×</td>
<td></td>
<td></td>
</tr>
<tr>
<td>O2.13</td>
<td>DiscardControl relationships MAY be included in an XPS Document, from the package to a DiscardControl part.</td>
<td>9.1.1 ×</td>
<td></td>
<td></td>
</tr>
<tr>
<td>O2.15</td>
<td>PrintTicket relationships MAY be included in an XPS Document, from a FixedDocumentSequence, FixedDocument, or FixedPage part to a PrintTicket part.</td>
<td>9.1.1 ×</td>
<td></td>
<td></td>
</tr>
<tr>
<td>O2.16</td>
<td>StoryFragments relationships MAY be included in an XPS Document, from a FixedPage part to a StoryFragments part.</td>
<td>9.1.1 ×</td>
<td></td>
<td></td>
</tr>
<tr>
<td>O2.17</td>
<td>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.</td>
<td>9.1.1 ×</td>
<td></td>
<td></td>
</tr>
<tr>
<td>O2.18</td>
<td>Color Profiles MAY be embedded in image files.</td>
<td>9.1.5 ×</td>
<td></td>
<td></td>
</tr>
<tr>
<td>O2.19</td>
<td>Thumbnail images MAY be attached to a FixedPage part using a Thumbnail relationship.</td>
<td>9.1.6 ×</td>
<td></td>
<td></td>
</tr>
<tr>
<td>O2.20</td>
<td>Fonts MAY be subsetted based on glyph usage.</td>
<td>9.1.7.1 ×</td>
<td></td>
<td></td>
</tr>
<tr>
<td>O2.21</td>
<td>Producers MAY use a 128-bit random number instead of a true GUID for an obfuscated font name.</td>
<td>9.1.7.3 ×</td>
<td></td>
<td></td>
</tr>
<tr>
<td>O2.22</td>
<td>An obfuscated Font part MAY have an arbitrary extension.</td>
<td>9.1.7.3 ×</td>
<td></td>
<td></td>
</tr>
<tr>
<td>O2.23</td>
<td>Producers MAY add digital signature requests and instructions to an XPS Document in the form of signature definitions.</td>
<td>9.1.10 ×</td>
<td></td>
<td></td>
</tr>
<tr>
<td>O2.24</td>
<td>A producer MAY sign against an existing signature definition to provide additional signature information.</td>
<td>9.1.10 ×</td>
<td></td>
<td></td>
</tr>
<tr>
<td>O2.25</td>
<td>A recipient of an XPS Document MAY also sign it against a signature definition.</td>
<td>9.1.10 ×</td>
<td></td>
<td></td>
</tr>
<tr>
<td>O2.27</td>
<td>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.</td>
<td>9.1.11 ×</td>
<td></td>
<td></td>
</tr>
<tr>
<td>O2.28</td>
<td>A resource that is intended to be used across multiple fixed documents MAY be named according to the guidelines for shared resources.</td>
<td>9.2 ×</td>
<td></td>
<td></td>
</tr>
<tr>
<td>O2.29</td>
<td>Producers MAY include Markup Compatibility and Extensibility elements and attributes in DocumentStructure, FixedDocument, FixedDocumentSequence, FixedPage, Relationships, Remote Resource Dictionary, SignatureDefinitions, and StoryFragments parts.</td>
<td>9.3.1 ×</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### I.3 Documents

#### 1.3.1 MUST Conformance Requirements

<table>
<thead>
<tr>
<th>ID</th>
<th>Rule</th>
<th>Reference</th>
<th>Producer</th>
<th>Consumer</th>
</tr>
</thead>
<tbody>
<tr>
<td>M3.1</td>
<td>The order of <code>&lt;DocumentReference&gt;</code> elements MUST match the order of the documents in the fixed document sequence.</td>
<td>10.1</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>M3.2</td>
<td>The Source attribute of the <code>&lt;DocumentReference&gt;</code> element MUST specify a FixedDocument part within the XPS Document.</td>
<td>10.1.1</td>
<td>x v</td>
<td></td>
</tr>
<tr>
<td>M3.3</td>
<td>Producers MUST NOT produce a document with multiple <code>&lt;DocumentReference&gt;</code> elements that reference the same fixed document.</td>
<td>10.1.1</td>
<td>x v</td>
<td></td>
</tr>
<tr>
<td>M3.4</td>
<td>The order of <code>&lt;PageContent&gt;</code> elements MUST match the order of the pages in the document.</td>
<td>10.2</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>M3.5</td>
<td>The Source attribute of the <code>&lt;PageContent&gt;</code> element MUST specify a FixedPage part within the XPS Document.</td>
<td>10.2.1</td>
<td>x v</td>
<td></td>
</tr>
<tr>
<td>M3.6</td>
<td>Producers MUST NOT produce markup where a <code>&lt;PageContent&gt;</code> element references the same fixed page referenced by any other <code>&lt;PageContent&gt;</code> element in the entire XPS Document, even in other fixed documents within the fixed payload.</td>
<td>10.2.1</td>
<td>x v</td>
<td></td>
</tr>
<tr>
<td>M3.7</td>
<td>If the attribute is specified, the BleedBox BleedOriginX value MUST be less than or equal to 0.</td>
<td>10.3.1</td>
<td>x v</td>
<td></td>
</tr>
<tr>
<td>M3.8</td>
<td>If the attribute is specified, the BleedBox BleedOriginY value MUST be less than or equal to 0.</td>
<td>10.3.1</td>
<td>x v</td>
<td></td>
</tr>
<tr>
<td>M3.9</td>
<td>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.</td>
<td>10.3.1</td>
<td>x v</td>
<td></td>
</tr>
<tr>
<td>M3.10</td>
<td>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.</td>
<td>10.3.1</td>
<td>x v</td>
<td></td>
</tr>
</tbody>
</table>
### I.3.2 SHOULD Conformance Requirements

<table>
<thead>
<tr>
<th>ID</th>
<th>Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>S3.1</strong></td>
<td>Specifying a ContentBox attribute for the <code>&lt;FixedPage&gt;</code> element is <strong>RECOMMENDED</strong>.</td>
</tr>
<tr>
<td><strong>S3.2</strong></td>
<td>Invalid bleed box specifications <strong>SHOULD</strong> be ignored in favor of the default bleed box.</td>
</tr>
<tr>
<td><strong>S3.3</strong></td>
<td>Invalid content box specifications <strong>SHOULD NOT</strong> be rendered and <strong>SHOULD</strong></td>
</tr>
</tbody>
</table>

---

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**
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**
3. **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**
4. **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**
5. **M3.15** When rendering a fixed page for printing, consumers **MUST** be aware of the interaction between the fixed page markup and the PrintTicket settings.
   - **10.3.4**
6. **M3.16** When the PrintTicket specifies the page scaling option `FitApplicationBleedSizeToPageImageableSize`, printing consumers **MUST** scale the bleed box (producer bleed size) to the PageImageableSize, preserving the aspect ratio.
   - **10.3.4.1**
7. **M3.17** When the PrintTicket specifies the page scaling option `FitApplicationContentSizeToPageImageableSize`, printing consumers **MUST** scale the content box (producer content size) to the PageImageableSize, preserving the aspect ratio.
   - **10.3.4.2**
8. **M3.18** When the PrintTicket specifies the page scaling option `FitApplicationMediaSizeToPageImageableSize`, printing consumers **MUST** scale the height and width (producer media size) to the PageImageableSize, preserving the aspect ratio.
   - **10.3.4.3**
9. **M3.19** When the PrintTicket specifies the page scaling option `FitApplicationMediaSizeToPageMediaSize`, printing consumers **MUST** scale the height and width (producer media size) to the PageMediaSize, preserving the aspect ratio.
   - **10.3.4.4**
10. **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**

---

**Table I–6. Document SHOULD conformance requirements**

<table>
<thead>
<tr>
<th>ID</th>
<th>Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>S3.1</strong></td>
<td>Specifying a ContentBox attribute for the <code>&lt;FixedPage&gt;</code> element is <strong>RECOMMENDED</strong>.</td>
</tr>
<tr>
<td><strong>S3.2</strong></td>
<td>Invalid bleed box specifications <strong>SHOULD</strong> be ignored in favor of the default bleed box.</td>
</tr>
<tr>
<td><strong>S3.3</strong></td>
<td>Invalid content box specifications <strong>SHOULD NOT</strong> be rendered and <strong>SHOULD</strong></td>
</tr>
</tbody>
</table>
generate an error.

S3.4 In the absence of media scaling, the fixed page content is imaged directly to the physical media with the origin of the fixed page aligned with the origin of the physical media size. Any fixed page content that extends beyond the dimension of the physical media size SHOULD be clipped.

### I.4 Graphics

#### I.4.1 MUST Conformance Requirements

<table>
<thead>
<tr>
<th>ID</th>
<th>Rule</th>
<th>Reference</th>
<th>Producer</th>
<th>Consumer</th>
</tr>
</thead>
<tbody>
<tr>
<td>M4.1</td>
<td>The x:Key attribute of the &lt;Path&gt; element MUST be present when the element is defined in a resource dictionary. It MUST NOT be specified outside a resource dictionary.</td>
<td>11.1</td>
<td>×</td>
<td>v</td>
</tr>
<tr>
<td>M4.2</td>
<td>The x:Key attribute of the &lt;PathGeometry&gt; element MUST be present when the element is defined in a resource dictionary. It MUST NOT be specified outside a resource dictionary.</td>
<td>11.2.1.1</td>
<td>×</td>
<td>v</td>
</tr>
<tr>
<td>M4.3</td>
<td>A &lt;PathGeometry&gt; element contains a set of path figures specified either with the Figures attribute or with a child &lt;PathFigure&gt; element. Producers MUST NOT specify the path figures of a geometry with both the Figures attribute and a child &lt;PathFigure&gt; element.</td>
<td>11.2.1.1</td>
<td>×</td>
<td>v</td>
</tr>
</tbody>
</table>

#### I.4.2 SHOULD Conformance Requirements

<table>
<thead>
<tr>
<th>ID</th>
<th>Rule</th>
<th>Reference</th>
<th>Producer</th>
<th>Consumer</th>
</tr>
</thead>
<tbody>
<tr>
<td>S4.1</td>
<td>Line segments and curve segments SHOULD NOT be specified as zero-length.</td>
<td>11.2.2.1</td>
<td>×</td>
<td></td>
</tr>
</tbody>
</table>

#### I.4.3 OPTIONAL Conformance Requirements

<table>
<thead>
<tr>
<th>ID</th>
<th>Rule</th>
<th>Reference</th>
<th>Producer</th>
<th>Consumer</th>
</tr>
</thead>
<tbody>
<tr>
<td>O4.1</td>
<td>Consumers or viewers that perform anti-aliasing MAY &quot;snap&quot; 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</td>
<td>4.1</td>
<td>×</td>
<td></td>
</tr>
</tbody>
</table>
## I.5 Text

### I.5.1 MUST Conformance Requirements

Table I-10. Text MUST conformance requirements

<table>
<thead>
<tr>
<th>ID</th>
<th>Rule</th>
<th>Reference</th>
<th>Producer</th>
<th>Consumer</th>
</tr>
</thead>
<tbody>
<tr>
<td>M5.1</td>
<td>If the CaretStops attribute is missing from the <code>&lt;Glyphs&gt;</code> 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.</td>
<td></td>
<td>x</td>
<td>v</td>
</tr>
<tr>
<td>M5.2</td>
<td>If the UnicodeString attribute of the <code>&lt;Glyphs&gt;</code> element specifies an empty string (“” or “{”“}”) and the Indices attribute is not specified or is empty, the consumer MUST generate an error.</td>
<td>12.1</td>
<td>x</td>
<td>v</td>
</tr>
<tr>
<td>M5.3</td>
<td>The x:Key attribute of the <code>&lt;Glyphs&gt;</code> element MUST be present when the element is defined in a resource dictionary. It MUST NOT be specified outside a resource dictionary.</td>
<td>12.1</td>
<td>x</td>
<td>v</td>
</tr>
<tr>
<td>M5.4</td>
<td>The sum of the code unit counts for all the GlyphMapping entries in the Indices attribute MUST NOT exceed the number of UTF-16 code units in the UnicodeString attribute if the UnicodeString attribute is specified and does not contain an empty value (“” or “{”“}”). If a ClusterMapping is not specified within a GlyphMapping entry, the code unit count is 1. If the Indices attribute specifies a GlyphIndex that does not exist in the font, the consumer MUST generate an error.</td>
<td>12.1.3</td>
<td>x</td>
<td>v</td>
</tr>
<tr>
<td>M5.5</td>
<td>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.</td>
<td>12.1.3</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>M5.6</td>
<td>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.</td>
<td>12.1.3</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>M5.7</td>
<td>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.</td>
<td>12.1.4</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>M5.9</td>
<td>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.</td>
<td>12.1.4</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>M5.10</td>
<td>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</td>
<td>12.1.4</td>
<td></td>
<td>x</td>
</tr>
</tbody>
</table>
control marks are mapped in the CMAP table.

| 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 | x |
| M5.12  | Producers MUST lay out algorithmically emboldened glyphs using advance widths that are 2% of the em size larger than when not algorithmically emboldened. | 12.1.5 | x |
| M5.13  | Consumers MUST implement the effect of algorithmic emboldening such that the black box of the glyph grows by 2% of the em size. When advance widths are omitted from the markup and the glyphs are algorithmically emboldened, the advance widths obtained from the horizontal metrics font table (if 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 | x |
| M5.14  | Producers MUST lay out algorithmically italicized glyphs using exactly the same advance widths as when not algorithmically italicized. | 12.1.5 | x |
| M5.15  | Producers MUST NOT specify text that is both right-to-left (BidiLevel attribute value of 1) and vertical (IsSideways attribute set to true). | 12.1.6.2 | v |
| 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 | x^b |
| 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 | x^b |
| 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 | x^b |
| 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 | v^fp |
| 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 | x |
| 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 | x^s |
| 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 | x |
I.5.2 SHOULD Conformance Requirements

Table I–11. Text SHOULD conformance requirements

<table>
<thead>
<tr>
<th>ID</th>
<th>Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>S5.1</td>
<td>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.</td>
</tr>
<tr>
<td>S5.2</td>
<td>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.</td>
</tr>
<tr>
<td>S5.3</td>
<td>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.</td>
</tr>
<tr>
<td>S5.4</td>
<td>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.</td>
</tr>
<tr>
<td>S5.5</td>
<td>Specifying a UnicodeString for &lt;Glyphs&gt; elements is RECOMMENDED, as it supports searching, selection, and accessibility.</td>
</tr>
</tbody>
</table>

I.5.3 OPTIONAL Conformance Requirements

Table I–12. Text OPTIONAL conformance requirements

<table>
<thead>
<tr>
<th>ID</th>
<th>Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>O5.1</td>
<td>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 specification.</td>
</tr>
<tr>
<td>O5.2</td>
<td>Producers MAY choose to generate UnicodeString attribute values that are not normalized by any Unicode-defined algorithm.</td>
</tr>
<tr>
<td>O5.3</td>
<td>Consumers that understand the device font name MAY ignore the embedded font and use the device-resident version.</td>
</tr>
<tr>
<td>O5.4</td>
<td>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.</td>
</tr>
<tr>
<td>O5.5</td>
<td>Glyph advance widths MAY be omitted from markup where the advance width desired is specified in the font tables, once adjusted for algorithmic emboldening.</td>
</tr>
<tr>
<td>O5.6</td>
<td>Glyph horizontal and vertical offsets MAY be omitted from markup</td>
</tr>
</tbody>
</table>
where the offset is 0.0.

I.6 Brushes

I.6.1 MUST Conformance Requirements

Table I–13. Brushes MUST conformance requirements

<table>
<thead>
<tr>
<th>ID</th>
<th>Rule</th>
<th>Reference</th>
<th>Producer</th>
<th>Consumer</th>
</tr>
</thead>
<tbody>
<tr>
<td>M6.1</td>
<td>The x:Key attribute of the &lt;SolidColorBrush&gt; element MUST be present when the element is defined in a resource dictionary. It MUST NOT be specified outside a resource dictionary.</td>
<td>13.1</td>
<td>x</td>
<td>v</td>
</tr>
<tr>
<td>M6.2</td>
<td>The x:Key attribute of the &lt;ImageBrush&gt; element MUST be present when the element is defined in a resource dictionary. It MUST NOT be specified outside a resource dictionary.</td>
<td>13.2</td>
<td>x</td>
<td>v</td>
</tr>
<tr>
<td>M6.3</td>
<td>An &lt;ImageBrush&gt; element MUST reference a JPEG, PNG, TIFF 6.0, or Windows Media Photo Image part within the XPS Document package.</td>
<td>13.2</td>
<td>x</td>
<td>v</td>
</tr>
<tr>
<td>M6.4</td>
<td>The x:Key attribute of the &lt;VisualBrush&gt; element MUST be present when the element is defined in a resource dictionary. It MUST NOT be specified outside a resource dictionary.</td>
<td>13.3</td>
<td>x</td>
<td>v</td>
</tr>
<tr>
<td>M6.5</td>
<td>The x:Key attribute of the &lt;LinearGradientBrush&gt; element MUST be present when the element is defined in a resource dictionary. It MUST NOT be specified outside a resource dictionary.</td>
<td>13.5</td>
<td>x</td>
<td>v</td>
</tr>
<tr>
<td>M6.6</td>
<td>The x:Key attribute of the &lt;RadialGradientBrush&gt; element MUST be present when the element is defined in a resource dictionary. It MUST NOT be specified outside a resource dictionary.</td>
<td>13.6</td>
<td>x</td>
<td>v</td>
</tr>
</tbody>
</table>

I.7 Common Properties

I.7.1 MUST Conformance Requirements

Table I–14. Common properties MUST conformance requirements

<table>
<thead>
<tr>
<th>ID</th>
<th>Rule</th>
<th>Reference</th>
<th>Producer</th>
<th>Consumer</th>
</tr>
</thead>
<tbody>
<tr>
<td>M7.1</td>
<td>Individual resource values MUST be specified within a resource dictionary.</td>
<td>14.2</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>M7.2</td>
<td>Namespace prefixes in resource definitions MUST apply in the context of the definition, rather than in the context of the resource reference.</td>
<td>14.2.3</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>M7.3</td>
<td>An xml:lang attribute within a resource definition MUST be interpreted in the context of the resource reference, not the resource definition.</td>
<td>14.2.3</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>
M7.4 A remote resource dictionary MUST follow the requirements that apply to inline resource dictionaries.

M7.5 A remote resource dictionary MUST NOT contain any resource definition children that reference another remote resource dictionary.

M7.6 A <ResourceDictionary> element that specifies a remote resource dictionary in its Source attribute MUST NOT contain any resource definition children.

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.

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.

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.

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.

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.

I.7.2 OPTIONAL Conformance Requirements

Table I–15. Common properties OPTIONAL conformance requirements

<table>
<thead>
<tr>
<th>ID</th>
<th>Rule</th>
<th>Reference</th>
<th>Producer</th>
<th>Consumer</th>
</tr>
</thead>
<tbody>
<tr>
<td>O7.1</td>
<td>Resource dictionaries MAY be specified in separate parts (called remote resource dictionaries) and referenced from within the &lt;FixedPage.Resources&gt; or &lt;Canvas.Resources&gt; property element.</td>
<td>14.2</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>O7.2</td>
<td>A resource definition MAY reference another resource defined prior to the point of reference, including a resource previously within the same resource dictionary.</td>
<td>14.2.3</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>O7.3</td>
<td>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 &lt;Canvas&gt; or &lt;FixedPage&gt; element.</td>
<td>14.2.3</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>O7.5</td>
<td>The resource dictionary of a &lt;Canvas&gt; 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 &lt;Canvas&gt; or &lt;FixedPage&gt; element.</td>
<td>14.2.5</td>
<td>x</td>
<td></td>
</tr>
</tbody>
</table>
A resource definition MAY reference a previously defined resource with the same name that is defined in an ancestor resource dictionary.

---

### I.8 Color

#### I.8.1 MUST Conformance Requirements

<table>
<thead>
<tr>
<th>ID</th>
<th>Rule</th>
<th>Reference</th>
<th>Producer</th>
<th>Consumer</th>
</tr>
</thead>
<tbody>
<tr>
<td>M8.1</td>
<td>Consumers MUST support sRGB colors (8 bit-per-channel) in vector data, with and without alpha.</td>
<td>15.1</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>M8.2</td>
<td>Consumers MUST support sRGB colors in image data, using the JPEG, PNG, TIFF, or Windows Media Photo image formats.</td>
<td>15.1</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>M8.3</td>
<td>Consumers MUST support scRGB color specification in vector data, with and without alpha.</td>
<td>15.1</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>M8.4</td>
<td>Consumers MUST support scRGB colors in image data, using the Windows Media Photo image format.</td>
<td>15.1</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>M8.5</td>
<td>Consumers MUST support CMYK colors in vector data.</td>
<td>15.1</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>M8.6</td>
<td>Consumers MUST support CMYK colors in image data, using the TIFF or Windows Media Photo image formats.</td>
<td>15.1</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>M8.7</td>
<td>Consumers MUST support N-Channel colors in vector data.</td>
<td>15.1</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>M8.8</td>
<td>Consumers MUST support N-Channel colors in image data, using the Windows Media Photo image format.</td>
<td>15.1</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>M8.9</td>
<td>Consumers MUST support ICC Version 2 profiles for 3-, 4-, 5-, 6-, 7-, and 8-channel color.</td>
<td>15.1</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>M8.10</td>
<td>Consumers MUST support ICC Version 2 profiles with a Windows Color System (WCS) profile embedded as a private tag, either by handling or ignoring the WCS profile.</td>
<td>15.1</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>M8.11</td>
<td>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.</td>
<td>15.1.7</td>
<td>x^pp</td>
<td></td>
</tr>
<tr>
<td>M8.12</td>
<td>ICC profiles MUST conform to the requirements in the ICC Color Profile specification, Version 3.4. 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.</td>
<td>15.1.8</td>
<td>x^U v</td>
<td></td>
</tr>
<tr>
<td>M8.13</td>
<td>All ICC profiles used in XPS Documents MUST be either an Input profile, an Output profile, a Monitor (RGB) profile, or a ColorSpace Conversion profile.</td>
<td>15.1.8</td>
<td>x^U v</td>
<td></td>
</tr>
<tr>
<td>M8.14</td>
<td>Real numbers specified for color channel values of scRGB and 15.1.11 x v</td>
<td>15.1.11</td>
<td>x v</td>
<td></td>
</tr>
</tbody>
</table>
ContextColor colors MUST NOT use exponent forms of numbers.

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.

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.

M8.17 For N-Channel colors, the context color MUST specify a number of channel float values equal to the number of channels in the profile, setting unused ones to 0.0.

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.

M8.19 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.

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.

M8.21 Consumers that do not understand a named color MUST compute a color approximation through ICM-compliant color management functions using the specified profile.

M8.22 The color name specified by the DocumentImpositionColorPrintTicket setting MUST be matched only to profiles containing exactly one non-zero-length colorant name in the profile's colorantTable.

M8.23 The color name specified by the DocumentImpositionColor setting serves as a label for that color only and MUST NOT be matched against any Named Colors known by the consumer.

M8.24 The comparison of the color name specified by the DocumentImpositionColorPrintTicket 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.

M8.25 For gradients, the specified blending color space in the PageBlendColorSpacePrintTicket 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 PageBlendColorSpacePrintTicket setting, the color interpolation mode of the gradient brush MUST be used instead.

M8.27 If the PageDeviceColorSpaceUsage is set to MatchToDeviceDefault and the profile specified by the PageDeviceColorSpaceURI PrintTicket setting cannot be used as a device color space profile, elements using the profile MUST be color managed like any other element using a color profile.
M8.28 If the PageBlendColorSpace PrintTicket setting is set to ICCProfile, the profile MUST be an output profile, otherwise it MUST be ignored.

M8.29 Elements using the named color identified by the DocumentImpositionColor PrintTicket setting MUST appear on all color separations.

M8.30 If an ICC profile is not embedded or associated with a raster image or if the embedded or associated profile is not compatible with the pixel format of the image, the default pixel formats for each color space MUST be treated as defined in §15.2.9.

M8.31 Channel and tint float values in CMYK, N-Channel, and Named Color syntax MUST be clamped to the valid range from 0.0 to 1.0 before further processing. If the value is used as input for an ICC profile color transformation, it MUST subsequently be linearly scaled to the range from 0 to 255 or from 0 to 65535, depending on whether the profile uses 8-bit or 16-bit input tables.

I.8.2 SHOULD Conformance Requirements

Table I–17. Color SHOULD conformance requirements

<table>
<thead>
<tr>
<th>ID</th>
<th>Rule</th>
<th>Reference</th>
<th>Producer</th>
<th>Consumer</th>
</tr>
</thead>
<tbody>
<tr>
<td>S8.1</td>
<td>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.</td>
<td>15.1.8</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>S8.2</td>
<td>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.</td>
<td>15.1.8</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>S8.3</td>
<td>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).</td>
<td>15.1.16</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>S8.4</td>
<td>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.</td>
<td>15.1.16</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>S8.5</td>
<td>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.</td>
<td>15.1.16</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>S8.6</td>
<td>Support for JPEG CMYK images varies by implementation and SHOULD NOT be used in XPS Documents.</td>
<td>15.2.4.3</td>
<td></td>
<td>x</td>
</tr>
</tbody>
</table>
S8.7 For consumers that do perform separation, the imposition named color is an indicator that the tint level supplied SHOULD be used for all device colorants.

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.

S8.9 If a consumer understands the PageBlendColorSpace PrintTicket setting, it SHOULD convert all color to the specified blending color space before performing a blend operation.

S8.10 If the PageDeviceColorSpaceUsage PrintTicket setting is set to MatchToDeviceDefault, the device's internal color profile SHOULD be used for color management of all elements not using the profile specified by the PageDeviceColorSpaceProfileURI PrintTicket setting.

S8.11 If the PageDeviceColorSpaceUsage PrintTicket setting is set to OverrideDeviceDefault and the profile specified by the PageDeviceColorSpaceProfileURI PrintTicket setting has a number of channels matching the number of primaries of the device, it SHOULD be used instead of the device's internal color management for all elements.

S8.12 If the PageBlendColorSpace PrintTicket setting is set to ICCProfile, the Uri property of the option specifies an ICC profile defining the color space that SHOULD be used for blending.

S8.13 The PageICMRenderingIntent PrintTicket setting SHOULD be ignored for elements using a profile that specifies the rendering intent in the profile.

S8.14 A consumer incapable of supporting named colors SHOULD treat the colorant table for named colors tag in an ICC profile as a user-defined custom tag, and therefore ignore it. The consumer SHOULD instead use the color tables as provided in the profile to convert the specified colors to the Profile Connection Space (PCS).

S8.15 Producers SHOULD restrict 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.

### I.8.3 OPTIONAL Conformance Requirements

<table>
<thead>
<tr>
<th>ID</th>
<th>Rule</th>
<th>Reference</th>
<th>Producer</th>
<th>Consumer</th>
</tr>
</thead>
<tbody>
<tr>
<td>O8.1</td>
<td>Consumers are not required to handle all color spaces natively, but rather MAY convert data specified in a rich color space to sRGB at an early stage (possibly resulting in reduced fidelity).</td>
<td>15.1</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>O8.2</td>
<td>An ICC profile MAY contain the private tag, “MS00”, which specifies an embedded Windows Color System (WCS) profile.</td>
<td>15.1.8</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>

---

Working Draft 1.1, February 2008
I. Conformance Requirements

O8.3 When a named color is used in a gradient brush or with transparency, the result produced by consumers that are not named-color aware MAY differ significantly from the result produced by consumers that are named-color aware.

O8.4 A named color profile MAY be used with images for spot coloring.

O8.5 Consumers MAY perform color separation, if desired.

O8.6 Consumers MAY support rich color spaces such as scRGB or CMYK. Consumers that encounter any document using non-sRGB colors MAY process those colors using the simpler sRGB color space, resulting in deviations, especially for alpha blending.

O8.7 If the PageColorManagement PrintTicket setting specifies a value of Driver, the driver MAY color manage elements or convert them to different color spaces.

O8.8 Elements using the profile specified by PageBlendColorSpace PrintTicket setting with a value of ICCProfile MAY be blended naively (channel-by-channel) without converting through PCS.

O8.9 XPS producers and consumers MAY provide color management using ICC profiles conforming to the requirements of ISO 15076-1, "Image technology colour management — Architecture, profile format, and data structure — Part 1: Based on ICC.1:2004-10" [O8.9].

1 I.9 Document Structure and Interactivity

1.9.1 MUST Conformance Requirements

Table I-19. Document structure MUST conformance requirements

<table>
<thead>
<tr>
<th>ID</th>
<th>Rule</th>
<th>Reference</th>
<th>Producer</th>
<th>Consumer</th>
</tr>
</thead>
<tbody>
<tr>
<td>M9.1</td>
<td>In order to merge the table cells and rows correctly, producers MUST specify empty &lt;TableCellStructure&gt; elements for cells that do not break across story fragments.</td>
<td>16.1.2</td>
<td>xD</td>
<td></td>
</tr>
<tr>
<td>M9.2</td>
<td>If hyperlinked &lt;Path&gt; or &lt;Glyphs&gt; 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.</td>
<td>16.2.1</td>
<td>xH</td>
<td></td>
</tr>
<tr>
<td>M9.3</td>
<td>If a producer specifies a FixedPage.NavigateUri attribute on a &lt;Canvas&gt; 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.</td>
<td>16.2.1</td>
<td>xH</td>
<td></td>
</tr>
<tr>
<td>M9.4</td>
<td>Relative internal hyperlinks between FixedPage parts MUST specify, at a minimum, the named address relative to the FixedDocument part.</td>
<td>16.2.1</td>
<td>xH</td>
<td>vH</td>
</tr>
<tr>
<td>M9.5</td>
<td>In order to be addressable by either a hyperlink or the document outline, the named address MUST appear in the &lt;PageContent.LinkTargets&gt; element in the fixed document.</td>
<td>16.2.1</td>
<td>xH</td>
<td></td>
</tr>
<tr>
<td>M9.6</td>
<td>If a named address appears in the &lt;PageContent.LinkTargets&gt; element in the fixed document but is not found in the Name attribute of an</td>
<td>16.2.1</td>
<td>xH</td>
<td></td>
</tr>
</tbody>
</table>
If a named address in a URI fragment is not found, consumers MUST ignore the fragment portion of the URI.

Internal references MUST specify a page address relative to the fixed document sequence.

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.

The Name attribute MUST NOT be specified on any children of a <ResourceDictionary> element.

## I.9.2 SHOULD Conformance Requirements

Table I–20. Document structure SHOULD conformance requirements

<table>
<thead>
<tr>
<th>ID</th>
<th>Rule</th>
<th>Reference</th>
<th>Producer</th>
<th>Consumer</th>
</tr>
</thead>
<tbody>
<tr>
<td>S9.1</td>
<td>Every meaningful element in the fixed page markup SHOULD specify a Name attribute in order for the document structure markup to refer to it</td>
<td>16.1.1</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>S9.3</td>
<td>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 &lt;Canvas&gt; element, producers SHOULD consider all descendant elements to be referenced in markup order.</td>
<td>16.1.1</td>
<td>x&lt;sup&gt;5&lt;/sup&gt;</td>
<td></td>
</tr>
<tr>
<td>S9.4</td>
<td>If a &lt;StoryBreak&gt; 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 &lt;StoryBreak&gt; element is not present at the end of the content structure markup, consumers SHOULD consider the markup a continuation to the next story fragment that must be merged to determine the cross-fragment content structure.</td>
<td>16.1.2</td>
<td>x&lt;sup&gt;5&lt;/sup&gt;</td>
<td></td>
</tr>
<tr>
<td>S9.5</td>
<td>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.</td>
<td>16.1.2.2</td>
<td>x&lt;sup&gt;5&lt;/sup&gt;</td>
<td></td>
</tr>
<tr>
<td>S9.6</td>
<td>If consumers enable user interactivity, they SHOULD support hyperlink activation and addressing.</td>
<td>16.2</td>
<td>x</td>
<td>H</td>
</tr>
<tr>
<td>S9.7</td>
<td>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.</td>
<td>16.2.1</td>
<td>x</td>
<td>H</td>
</tr>
<tr>
<td>S9.8</td>
<td>The value of the Name attribute on a &lt;FixedPage&gt;, &lt;Canvas&gt;, &lt;Path&gt;, or &lt;Glyphs&gt; element SHOULD be unique within the scope of the fixed document.</td>
<td>16.2.1</td>
<td>x</td>
<td>v</td>
</tr>
<tr>
<td>S9.9</td>
<td>It is RECOMMENDED that Name attribute values on &lt;FixedPage&gt;,</td>
<td>16.2.1</td>
<td>x</td>
<td>v</td>
</tr>
</tbody>
</table>
### I. Conformance Requirements

- `<Canvas>`, `<Path>`, and `<Glyphs>` elements be unique within an entire fixed document sequence.

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>S9.10</td>
<td>If the Name attribute is specified, producers SHOULD also create a corresponding <code>&lt;LinkTarget&gt;</code> element in the FixedDocument part within the <code>&lt;PageContent&gt;</code> element that links to the parent fixed page.</td>
</tr>
<tr>
<td>S9.11</td>
<td>A hyperlink destination in the same fixed document SHOULD be expressed as a relative URI.</td>
</tr>
<tr>
<td>S9.12</td>
<td>If selection is supported, consumers SHOULD provide a visual cue over or around selected elements.</td>
</tr>
<tr>
<td>S9.13</td>
<td>Selection order within an XPS Document SHOULD follow reading order.</td>
</tr>
<tr>
<td>S9.14</td>
<td>In the absence of document structure provided in the XPS Document, consumers SHOULD, at minimum, rely on the markup order to determine reading order.</td>
</tr>
<tr>
<td>S9.15</td>
<td>Producers SHOULD order the markup in FixedPage parts to reflect the order in which it is intended to be read.</td>
</tr>
<tr>
<td>S9.16</td>
<td>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.</td>
</tr>
<tr>
<td>S9.17</td>
<td>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 <code>&lt;StoryFragment&gt;</code> elements are specified in the StoryFragments part for that page, 3) order by <code>&lt;NamedElement&gt;</code> reference within the <code>&lt;StoryFragment&gt;</code> element, 4) append all un-referenced elements that appear in the fixed page markup, ordered by markup order.</td>
</tr>
<tr>
<td>S9.18</td>
<td>Producers SHOULD order <code>&lt;StoryFragment&gt;</code> elements in each StoryFragments part in their intended reading order.</td>
</tr>
<tr>
<td>S9.19</td>
<td>The RECOMMENDED reading order of a story-centric application is as follows: 1) Order content by story in the sequence the <code>&lt;Story&gt;</code> elements appear in the DocumentStructure part. 2) Within a story, order <code>&lt;StoryFragmentReference&gt;</code> elements in the sequence they appear in the DocumentStructure part. 3) Within a story fragment, order by <code>&lt;NamedElement&gt;</code> references in the StoryFragments part markup. 4) Append all un-referenced elements that appear in the fixed page markup, ordered by page number, then markup order.</td>
</tr>
<tr>
<td>S9.20</td>
<td>Producers SHOULD order <code>&lt;Story&gt;</code> elements in the DocumentStructure part in their intended reading order.</td>
</tr>
<tr>
<td>S9.21</td>
<td>Producers SHOULD order <code>&lt;StoryFragmentReference&gt;</code> elements within a <code>&lt;Story&gt;</code> element in their intended reading order.</td>
</tr>
<tr>
<td>S9.22</td>
<td>A screen reader consumer SHOULD read the document according to its reading order.</td>
</tr>
<tr>
<td>S9.23</td>
<td>A screen reader SHOULD use the <code>UnicodeString</code> attribute of each <code>&lt;Glyphs&gt;</code> element to determine the text to read.</td>
</tr>
<tr>
<td>S9.24</td>
<td>If a screen reader provides features to navigate the document by structural elements, such as paragraphs or table rows, it SHOULD</td>
</tr>
</tbody>
</table>
use any document structure information included in the XPS Document.

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.

S9.26 If the screen reader provides features to describe hyperlink addresses, it SHOULD read the text provided in the FixedPage.NavigateUri attribute.

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.


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.

S9.30 Children of <VisualBrush> elements SHOULD NOT be referenced by document structure markup.

1 I.9.3 OPTIONAL Conformance Requirements

2 Table I–21. Document structure OPTIONAL conformance requirements

<table>
<thead>
<tr>
<th>ID</th>
<th>Rule</th>
<th>Reference</th>
<th>Producer</th>
<th>Consumer</th>
</tr>
</thead>
<tbody>
<tr>
<td>O9.1</td>
<td>Producers MAY choose to add document structure information to XPS Documents. Consumers MAY ignore any authored document structure or hyperlinks.</td>
<td>Clause 16</td>
<td>xD</td>
<td>xD</td>
</tr>
<tr>
<td>O9.2</td>
<td>Producers MAY provide either the document outline or the document content, or both; consumers MAY ignore either or both.</td>
<td>16.1</td>
<td>xD</td>
<td>xD</td>
</tr>
<tr>
<td>O9.3</td>
<td>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.</td>
<td>16.1.1</td>
<td>xD</td>
<td>xD</td>
</tr>
<tr>
<td>O9.4</td>
<td>Consumers MAY first attempt to locate named elements for document structure directly from the FixedDocument part markup, where they might appear as &lt;LinkTarget&gt; elements if that named element is also intended as an addressable location.</td>
<td>16.1.1</td>
<td>xD</td>
<td>xD</td>
</tr>
</tbody>
</table>
I. Conformance Requirements

1.1. XPS Document Package Features

1.10 MUST Conformance Requirements

<table>
<thead>
<tr>
<th>ID</th>
<th>Rule</th>
<th>Reference</th>
<th>Producer</th>
<th>Consumer</th>
</tr>
</thead>
<tbody>
<tr>
<td>M10.1</td>
<td>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.</td>
<td>17.1</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>M10.2</td>
<td>Consumers MUST be able to consume packages regardless of their interleaving structure.</td>
<td>17.1.3</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>M10.3</td>
<td>Consumers that lack the resources to process a part MUST indicate an error condition.</td>
<td>17.1.3</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>M10.4</td>
<td>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.</td>
<td>17.1.3</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>M10.5</td>
<td>If a consumer encounters a reference to an unknown part, it MUST continue to receive further bytes of the package until the unknown</td>
<td>17.1.3</td>
<td>x</td>
<td></td>
</tr>
</tbody>
</table>

I.10.1 MUST Conformance Requirements

Table I–22. XPS Document package feature MUST conformance requirements
part has been transmitted or until the end of the package is reached (indicating an error condition).

| M10.6 | The DiscardControl part MUST NOT reference itself. | 17.1.4.1 | × | v |
| 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 | ×<sup>▲</sup> | ×<sup>▲</sup> |
| M10.9 | Consumers MUST NOT prevent an end user from taking an action solely because doing so will disable functionality based on whether or not an action will invalidate a signature. | 17.2.1 | ×<sup>▲</sup> |
| M10.10 | An XPS Document MUST be considered signed according to the XPS Document signing policy, regardless of the validity of that signature, if the signing rules described in §10.2.1.1, “Signing Rules” are observed. | 17.2.1.1 | ×<sup>▲</sup> |
| M10.11 | An XPS Document MUST NOT be considered signed according to the XPS Document signing policy if any part not covered by the signing rules is included in the signature or if any relationship not covered by the signing rules is included in the signature. | 17.2.1.1 | ×<sup>▲</sup> |
| M10.12 | An XPS Document digital signer MUST NOT sign an XPS Document that contains content (parts or relationships parts) to be signed that defines the Markup Compatibility namespace but the signer does not fully understand all elements, attributes, and alternate content representations introduced through the markup compatibility mechanisms. | 17.2.1.1 | ×<sup>▲</sup> |
| 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 | ×<sup>▲</sup> |
| 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 | ×<sup>▲</sup> |
| 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 | ×<sup>▲</sup> |
| 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 | ×<sup>▲</sup> |
| M10.17 | To prohibit additional signatures in an XPS Document, the signing application MUST sign all the Digital Signature Origin part’s relationships of relationship type Digital Signature with the same signature as the rest of the content. | 17.2.1.3 | ×<sup>▲</sup> |
I. Conformance Requirements

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.

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.

M10.20 Due to space and rendering limitations, producers MUST NOT assume that consumers will use the values specified in the <SpotLocation> element.

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.

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."

M10.23 There MUST NOT be more than one DiscardControl package relationship.

M10.24 In some cases, producers might rewrite the contents of a package so that parts are provided more than once, allowing consumers to discard a part in order to free resources for additional processing. Each instance of a part MUST be stored as a new, uniquely named part in the package.

1

I.10.2 SHOULD Conformance Requirements

2 Table I–23. XPS Document package feature SHOULD conformance requirements

<table>
<thead>
<tr>
<th>ID</th>
<th>Rule</th>
<th>Reference</th>
<th>Producer</th>
<th>Consumer</th>
</tr>
</thead>
<tbody>
<tr>
<td>S10.1</td>
<td>When interleaving, the Content Types stream SHOULD be interleaved according to the recommendations in the OPC specification.</td>
<td>17.1</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>S10.2</td>
<td>When interleaving, PrintTicket parts SHOULD be written to the package before the part to which they are attached.</td>
<td>17.1</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>S10.3</td>
<td>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.</td>
<td>17.1</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>S10.4</td>
<td>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.</td>
<td>17.1</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>S10.5</td>
<td>When interleaving, the last piece of the Relationships part for a FixedPage part SHOULD be written to the package in close</td>
<td>17.1</td>
<td>x</td>
<td></td>
</tr>
</tbody>
</table>
proximity to the first piece of the FixedPage part.

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>S10.6</td>
<td>It is RECOMMENDED that one empty PrintTicket be shared for all parts that attach an empty PrintTicket.</td>
</tr>
<tr>
<td>S10.7</td>
<td>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.</td>
</tr>
<tr>
<td>S10.8</td>
<td>DiscardControl parts that are not well-formed SHOULD NOT be processed and an error SHOULD NOT be reported.</td>
</tr>
<tr>
<td>S10.9</td>
<td>If a <code>&lt;Discard&gt;</code> 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 <code>&lt;Discard&gt;</code> 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.</td>
</tr>
<tr>
<td>S10.10</td>
<td>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.</td>
</tr>
<tr>
<td>S10.11</td>
<td>Consumers SHOULD inform the end user if an action they are going to take will invalidate an existing signature.</td>
</tr>
<tr>
<td>S10.12</td>
<td>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.</td>
</tr>
<tr>
<td>S10.13</td>
<td>If the location specified by the <code>&lt;SpotLocation&gt;</code> 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.</td>
</tr>
<tr>
<td>S10.14</td>
<td>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.</td>
</tr>
<tr>
<td>S10.15</td>
<td>It is RECOMMENDED that a signature spot be a clickable area used to launch the digital signing process.</td>
</tr>
<tr>
<td>S10.16</td>
<td>If the <code>&lt;SignBy&gt;</code> 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.</td>
</tr>
<tr>
<td>S10.17</td>
<td>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.</td>
</tr>
</tbody>
</table>
## I. Conformance Requirements

**Head-first XPS Document consumers SHOULD attempt to detect inconsistent packages as soon as possible and SHOULD generate an error message, even if they have already processed the pages that resulted in the error.**

**The viewing consumer SHOULD use the values specified in the <SpotLocation> element to place a signature spot.**

### 1.10.3 OPTIONAL Conformance Requirements

Table I–24. XPS Document package feature OPTIONAL conformance requirements

<table>
<thead>
<tr>
<th>ID</th>
<th>Rule</th>
<th>Reference</th>
<th>Producer</th>
<th>Consumer</th>
</tr>
</thead>
<tbody>
<tr>
<td>O10.1</td>
<td>Interleaving is OPTIONAL.</td>
<td>17.1</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>O10.2</td>
<td>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.</td>
<td>17.1.2</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>O10.3</td>
<td>Consumers MAY discard FixedPage parts once they have been processed.</td>
<td>17.1.3</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>O10.4</td>
<td>Consumers MAY discard FixedDocument and FixedDocumentSequence parts after all their child elements and their closing tags have been processed.</td>
<td>17.1.3</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>O10.5</td>
<td>In the absence of explicit directives to the contrary, consumers MAY discard parts as directed by the DiscardControl part.</td>
<td>17.1.3</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>O10.6</td>
<td>Some producers (typically drivers) MAY choose a suitable interleaving order by modeling the resource management behavior of the consumer.</td>
<td>17.1.4</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>O10.7</td>
<td>A consumer MAY decide to ignore a malformed DiscardControl part in its entirety or from the first malformed node onward.</td>
<td>17.1.4.1</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>O10.8</td>
<td>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.</td>
<td>17.2.1.1</td>
<td>x^A</td>
<td></td>
</tr>
<tr>
<td>O10.9</td>
<td>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.</td>
<td>17.2.1.2</td>
<td>x^A</td>
<td></td>
</tr>
<tr>
<td>O10.10</td>
<td>XPS Documents MAY be signed more than once.</td>
<td>17.2.1.3</td>
<td>x^A</td>
<td></td>
</tr>
<tr>
<td>O10.11</td>
<td>Producers MAY include the JobDigitalSignatureProcessing setting in the job-level PrintTicket within the XPS Document content.</td>
<td>17.2.1.5</td>
<td>x^A</td>
<td></td>
</tr>
<tr>
<td>O10.12</td>
<td>The SpotID attribute of the &lt;SignatureDefinition&gt; element MAY be used to link to an existing signature.</td>
<td>17.2.2.2.1</td>
<td>x^A</td>
<td></td>
</tr>
<tr>
<td>O10.13</td>
<td>Consumers MAY choose a size and shape to display a signature spot based on the desired display information and page content.</td>
<td>17.2.2.3</td>
<td>x^A</td>
<td></td>
</tr>
<tr>
<td>O10.14</td>
<td>The &lt;SigningLocation&gt; element MAY be set by the original producer of the XPS Document or by the signing party at the time.</td>
<td>17.2.2.6</td>
<td>x^A</td>
<td></td>
</tr>
</tbody>
</table>
I.11 Rendering Rules

I.11.1 MUST Conformance Requirements

Table I-25. Rendering rules MUST conformance requirements

<table>
<thead>
<tr>
<th>ID</th>
<th>Rule</th>
<th>Reference</th>
<th>Producer</th>
<th>Consumer</th>
</tr>
</thead>
<tbody>
<tr>
<td>M11.1</td>
<td>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</td>
<td>Clause 18</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>M11.2</td>
<td>If a non-invertible transform is encountered during rendering, consumers MUST omit rendering the affected element and all of its child and descendant elements.</td>
<td>18.1.3</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>M11.3</td>
<td>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.</td>
<td>18.1.3</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>M11.4</td>
<td>Producers MUST NOT assume a specific placement error for curve decomposition or rely on side-effects of a specific consumer implementation.</td>
<td>18.1.5</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>M11.5</td>
<td>Encountering markup with characteristics outside of the consumer-specific implementation limits MUST cause an error condition.</td>
<td>18.2</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>M11.6</td>
<td>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.</td>
<td>18.4.1</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>M11.7</td>
<td>Composition MUST have the same effect as the application of the rules in §18.5, “Composition Rules,” in sequence.</td>
<td>18.5</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>M11.8</td>
<td>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.</td>
<td>18.7.2</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>M11.9</td>
<td>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.</td>
<td>18.7.3</td>
<td>x</td>
<td></td>
</tr>
</tbody>
</table>
### I.11.2 SHOULD Conformance Requirements

<table>
<thead>
<tr>
<th>ID</th>
<th>Rule</th>
<th>Reference</th>
<th>Producer</th>
<th>Consumer</th>
</tr>
</thead>
<tbody>
<tr>
<td>S11.1</td>
<td>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.</td>
<td>18.1.2 Table 18–1</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>S11.2</td>
<td>An ideal consumer implementation SHOULD render pixels in an 8×8 sub-pixel space, perform an 8×8 box filter sampling, and set the pixel to the resulting color value.</td>
<td>18.1.4</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>S11.3</td>
<td>When rendering a shape, a practical 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.</td>
<td>18.1.4</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>S11.4</td>
<td>When rendering geometries, consumers SHOULD render curves so they appear smooth from a normal viewing distance.</td>
<td>18.1.5</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>S11.5</td>
<td>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.</td>
<td>18.1.7</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>S11.6</td>
<td>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 8×8 sub-pixel space and subsequently draw only those sub-pixels of a shape that correspond to “ON” sub-pixels in the mask.</td>
<td>18.1.8</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>S11.7</td>
<td>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.</td>
<td>18.1.8</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>S11.8</td>
<td>A typical consumer SHOULD be able to process markup with the implementation limit characteristics indicated in Table 18–1. Producers SHOULD produce only XPS.</td>
<td>18.2</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>Requirement</td>
<td>Description</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------</td>
<td>-------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S11.9</td>
<td>Coordinates are real numbers and SHOULD be computed with at least single floating point precision.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S11.10</td>
<td>If the nesting level of <code>&lt;VisualBrush&gt;</code> elements is higher than 16, a consumer SHOULD attempt to flatten the nested content to a bitmap representation rather than failing to draw.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S11.11</td>
<td>Gradients SHOULD be rendered according to the guidelines described in §18.3, “Gradient Computations.”</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S11.12</td>
<td>Consumers SHOULD pre-process gradient stops for all gradients using the steps described in §11.3.1.1, “Gradient Stop Pre-Processing.”</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S11.13</td>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S11.14</td>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S11.15</td>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S11.16</td>
<td>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 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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S11.17</td>
<td>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.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S11.18</td>
<td>All opacity calculations SHOULD be performed with at least 8-bit precision to provide sufficient quality for nested content.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S11.19</td>
<td>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.</td>
<td>18.4.1</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>S11.20</td>
<td>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.</td>
<td>18.5</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>S11.21</td>
<td>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.</td>
<td>18.6</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>S11.22</td>
<td>Consumers SHOULD ensure that parallel edges of strokes appear parallel.</td>
<td>18.6.1</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>S11.23</td>
<td>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.</td>
<td>18.6.2</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>S11.24</td>
<td>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).</td>
<td>18.6.3</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>S11.25</td>
<td>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.</td>
<td>18.6.8</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>S11.26</td>
<td>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.</td>
<td>18.7.1</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>S11.27</td>
<td>Producers SHOULD avoid producing extreme cases where either the height, width, or both height and width are nearly zero and SHOULD NOT rely on any specific behavior when they do</td>
<td>18.7.1</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>S11.28</td>
<td>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.</td>
<td>18.7.2</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>S11.29</td>
<td>When up-sampling an image presented at a lower resolution than the device resolution, bilinear filtering SHOULD be used.</td>
<td>18.7.2</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>S11.30</td>
<td>When down-sampling an image presented at a higher resolution than the device resolution, at least a bilinear filter SHOULD be used.</td>
<td>18.7.2</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>
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.

### I.11.3 OPTIONAL Conformance Requirements

<table>
<thead>
<tr>
<th>ID</th>
<th>Rule</th>
<th>Reference</th>
<th>Producer</th>
<th>Consumer</th>
</tr>
</thead>
<tbody>
<tr>
<td>O11.1</td>
<td>Very high resolution devices MAY use lower fractional precision than a 28.4 fixed-point representation to represent device coordinates.</td>
<td>18.1.2</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>O11.2</td>
<td>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.</td>
<td>18.1.4</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>O11.3</td>
<td>Devices MAY use sub-pixel masking.</td>
<td>18.1.4</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>O11.4</td>
<td>An implementation capable of anti-aliasing MAY draw a thin line in a way that blends with the background to varying degrees.</td>
<td>18.1.4</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>O11.5</td>
<td>A bi-tonal implementation on a printer MAY draw thin lines with or without drop-outs, or by applying half-toning, depending on the desired output quality.</td>
<td>18.1.4</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>O11.6</td>
<td>Consumers MAY apply pixel placement rules optimized for character rendering to individual glyphs in a &lt;Glyphs&gt; element.</td>
<td>18.1.6</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>O11.7</td>
<td>Behavior of blending with very close gradient stops MAY vary in an implementation-specific manner (see S11.16).</td>
<td>18.3.1.2</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>O11.8</td>
<td>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.</td>
<td>18.3.1.2</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>O11.9</td>
<td>In certain scenarios (such as when rendering 3D scenes to a bitmap), producers MAY choose to create pre-multiplied bitmap data specifying “superluminous” colors.</td>
<td>18.4.1</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>O11.10</td>
<td>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.</td>
<td>18.4.1</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>O11.11</td>
<td>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.</td>
<td>18.5</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>O11.12</td>
<td>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 <code>&lt;Canvas&gt;</code> 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.</td>
<td>18.5.1</td>
<td>☒</td>
<td></td>
</tr>
<tr>
<td>O11.13</td>
<td>When doing page composition, if an element is fully transparent (an opacity of 0.0), it MAY be skipped.</td>
<td>18.5.1</td>
<td>☒</td>
<td></td>
</tr>
<tr>
<td>O11.14</td>
<td>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.</td>
<td>18.5.1</td>
<td>☒</td>
<td></td>
</tr>
<tr>
<td>O11.15</td>
<td>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.</td>
<td>18.5.1</td>
<td>☒</td>
<td></td>
</tr>
<tr>
<td>O11.16</td>
<td>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.</td>
<td>18.5.1</td>
<td>☒</td>
<td></td>
</tr>
<tr>
<td>O11.17</td>
<td>When doing page composition, a consumer MAY use methods to achieve transparency other than creating a temporary surface. Such methods MAY include planar mapping.</td>
<td>18.5.1</td>
<td>☒</td>
<td></td>
</tr>
<tr>
<td>O11.21</td>
<td>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.</td>
<td>18.7.1</td>
<td>☒</td>
<td></td>
</tr>
<tr>
<td>O11.22</td>
<td>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.</td>
<td>18.7.2</td>
<td>☒</td>
<td></td>
</tr>
<tr>
<td>O11.23</td>
<td>Consumers MAY choose any technique desired to achieve the requirement to precisely place a tile possibly resulting in fractional device pixel placement, such as linear filtering for seams, stretching of the tile (up-sampling or down-sampling), or pre-computing multiple tiles and adjusting behavior according to how the tiles fit on a grid.</td>
<td>18.7.3</td>
<td>☒</td>
<td></td>
</tr>
<tr>
<td>O11.24</td>
<td>Temporary work canvases MAY be re-used when tiling transparent brushes.</td>
<td>18.7.4</td>
<td>☒</td>
<td></td>
</tr>
<tr>
<td>O11.25</td>
<td>Producers MAY generate a <code>&lt;Path&gt;</code> 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.</td>
<td>18.6.12</td>
<td>☒</td>
<td></td>
</tr>
<tr>
<td><strong>O11.26</strong> 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.</td>
<td>18.1.4/18.7.4</td>
<td>☒</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
I.12 Additional Conformance Requirements

I.12.1 MUST Conformance Requirements

Table I–28. Additional MUST conformance requirements

<table>
<thead>
<tr>
<th>ID</th>
<th>Rule</th>
<th>Reference</th>
<th>Producer</th>
<th>Consumer</th>
</tr>
</thead>
<tbody>
<tr>
<td>M12.1</td>
<td>FixedDocument parts MUST be referenced by &lt;DocumentReference&gt; 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.</td>
<td>–</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>M12.2</td>
<td>A FixedDocument part MUST NOT be referenced more than once by a FixedDocumentSequence part.</td>
<td>–</td>
<td>×</td>
<td>v</td>
</tr>
<tr>
<td>M12.3</td>
<td>A FixedPage part MUST NOT be referenced more than once in total, throughout all FixedDocument parts.</td>
<td>–</td>
<td>×</td>
<td>v</td>
</tr>
<tr>
<td>M12.4</td>
<td>FixedPage parts MUST be referenced by &lt;PageContent&gt; 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 &lt;PageContent&gt; elements to the FixedDocument in reverse order or by binding the right side of the page.</td>
<td>–</td>
<td>×</td>
<td></td>
</tr>
<tr>
<td>M12.5</td>
<td>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 &lt;DocumentReference&gt; or &lt;PageContent&gt; element MUST have no more than one attached PrintTicket part.</td>
<td>–</td>
<td>×</td>
<td>v</td>
</tr>
<tr>
<td>M12.6</td>
<td>Every Font part reachable from the primary fixed payload root or its related parts by relationship or by the Source attribute on a &lt;DocumentReference&gt; or &lt;PageContent&gt; element MUST be a valid OpenType font.</td>
<td>–</td>
<td>×</td>
<td>v</td>
</tr>
<tr>
<td>M12.7</td>
<td>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.</td>
<td>I.2</td>
<td>×</td>
<td>v</td>
</tr>
</tbody>
</table>

End of informative text.
# J. Bibliography

   http://www.microsoft.com/typography/otspec/os2.htm
   http://www.microsoft.com/typography/otspec/otff.htm
   http://www.microsoft.com/typography/otspec/default.htm
K. Index

In the index that follows, italic page numbers are used to indicate illustrations and examples with illustrations. Bold page numbers are used to indicate a primary references when several pages are listed. Page ranges are elided. "See" references indicate the primary index location for that topic, while "See also" references indicate related index topics.

A

abbreviated geometry syntax .................................................. 8
accessibility .............................................................................. 10
anti-aliasing ............................................................................ 34
described .................................................................................. 61
document structure, enabled by .................................................... 11
document structure, dependent on .............................................. 25
end point .................................................................................... 62
example ..................................................................................... 63
gradient stops, specifying ............................................................ 64
image brush ................................................................................ 51
image text alternative, long ........................................................ 13
image text alternative, short ......................................................... 17
importance of ............................................................................ 20
importance of the text ................................................................. 22
of page-centric application .......................................................... 28
of story-centric application ........................................................ 29
offset ......................................................................................... 49
on path ....................................................................................... 69
on canvas ................................................................................... 61
printing ....................................................................................... 61
profile ......................................................................................... 69
rendering ..................................................................................... 69
screen reader ............................................................................ 30
segment ...................................................................................... 56
solid color brush ....................................................................... 94
spread method ........................................................................... 68
start point ................................................................................... 74
specifying .................................................................................. 73
linear gradient brush .................................................................. 60
color interpolation mode ............................................................... 152
end point .................................................................................... 62
example ..................................................................................... 63
gradient stops, specifying ............................................................ 64
mapping mode ........................................................................... 65
rendering ..................................................................................... 66
specifying .................................................................................. 67
radial gradient brush .................................................................. 76
color interpolation mode ............................................................... 158
end point .................................................................................... 62
example ..................................................................................... 63
gradient stops, specifying ............................................................ 64
gradient origin ........................................................................... 81
gradient y-radius ....................................................................... 84
granient x-radius ....................................................................... 83
mappng mode ........................................................................... 85
tile ............................................................................................... 95
virtual device ............................................................................. 106
XPS Specification and Reference Guide 451

B

bleed area ................................................................................... 40
bookmark .................................................................................. 41
brush .......................................................................................... 42
content ......................................................................................... 42
described ..................................................................................... 125
end point .................................................................................... 62
example ..................................................................................... 63
gradient stops, specifying ............................................................ 64
gradient x-radius ....................................................................... 83
gradient y-radius ....................................................................... 84
mapping mode ........................................................................... 85
rendering ..................................................................................... 86
specifying .................................................................................. 87
spread method ........................................................................... 88
described ..................................................................................... 162
end point .................................................................................... 62
example ..................................................................................... 63
gradient stops, specifying ............................................................ 64
specifying .................................................................................. 93
solid color brush ....................................................................... 94
tile ............................................................................................... 95

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also

See also
<table>
<thead>
<tr>
<th>Page</th>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>document imposition color</td>
<td>224</td>
</tr>
<tr>
<td>2</td>
<td>fidelity, improved</td>
<td>209</td>
</tr>
<tr>
<td>3</td>
<td>gamma</td>
<td>218</td>
</tr>
<tr>
<td>4</td>
<td>See color, blending</td>
<td>200</td>
</tr>
<tr>
<td>5</td>
<td>pixel formats</td>
<td>218</td>
</tr>
<tr>
<td>6</td>
<td>See color, raster image support</td>
<td>200</td>
</tr>
<tr>
<td>7</td>
<td>printing</td>
<td>218</td>
</tr>
<tr>
<td>8</td>
<td>See PrintTicket keywords</td>
<td>200</td>
</tr>
<tr>
<td>9</td>
<td>raster image support</td>
<td>218</td>
</tr>
<tr>
<td>10</td>
<td>associating a color profile part</td>
<td>220</td>
</tr>
<tr>
<td>11</td>
<td>device color</td>
<td>218</td>
</tr>
<tr>
<td>12</td>
<td>named colors</td>
<td>220</td>
</tr>
<tr>
<td>13</td>
<td>n-channel</td>
<td>220</td>
</tr>
<tr>
<td>14</td>
<td>pixel format defaults</td>
<td>220</td>
</tr>
<tr>
<td>15</td>
<td>scRGB</td>
<td>218</td>
</tr>
<tr>
<td>16</td>
<td>sRGB</td>
<td>217</td>
</tr>
<tr>
<td>17</td>
<td>rendering intent, specifying</td>
<td>224</td>
</tr>
<tr>
<td>18</td>
<td>syntax</td>
<td>215</td>
</tr>
<tr>
<td>19</td>
<td>CMYK</td>
<td>216</td>
</tr>
<tr>
<td>20</td>
<td>n-channel</td>
<td>215</td>
</tr>
<tr>
<td>21</td>
<td>scRGB</td>
<td>214</td>
</tr>
<tr>
<td>22</td>
<td>sRGB</td>
<td>213</td>
</tr>
<tr>
<td>23</td>
<td>summarized</td>
<td>213</td>
</tr>
<tr>
<td>24</td>
<td>where used</td>
<td>212</td>
</tr>
<tr>
<td>25</td>
<td>See color, color profile</td>
<td>212</td>
</tr>
<tr>
<td>26</td>
<td>compositability of properties</td>
<td>212</td>
</tr>
<tr>
<td>27</td>
<td>See XPS Document format, properties, compositability</td>
<td>212</td>
</tr>
<tr>
<td>28</td>
<td>compression</td>
<td>212</td>
</tr>
<tr>
<td>29</td>
<td>See image</td>
<td>212</td>
</tr>
<tr>
<td>30</td>
<td>See Open Packaging Conventions, specification</td>
<td>212</td>
</tr>
<tr>
<td>31</td>
<td>conformity</td>
<td>17</td>
</tr>
<tr>
<td>32</td>
<td>inherited from Open Packaging Conventions</td>
<td>3</td>
</tr>
<tr>
<td>33</td>
<td>of software</td>
<td>3</td>
</tr>
<tr>
<td>34</td>
<td>requirements tables</td>
<td>407-47</td>
</tr>
<tr>
<td>35</td>
<td>consumer</td>
<td>7</td>
</tr>
<tr>
<td>36</td>
<td>described</td>
<td>17</td>
</tr>
<tr>
<td>37</td>
<td>implementation burden</td>
<td>407-47</td>
</tr>
<tr>
<td>38</td>
<td>content area</td>
<td>404</td>
</tr>
<tr>
<td>39</td>
<td>See page, content area</td>
<td>7</td>
</tr>
<tr>
<td>40</td>
<td>content type</td>
<td>7</td>
</tr>
<tr>
<td>41</td>
<td>namespace</td>
<td>404</td>
</tr>
<tr>
<td>42</td>
<td>See namespace</td>
<td>7</td>
</tr>
<tr>
<td>43</td>
<td>summarized</td>
<td>7</td>
</tr>
<tr>
<td>44</td>
<td>usage</td>
<td>21</td>
</tr>
<tr>
<td>45</td>
<td>contour intersection point</td>
<td>7</td>
</tr>
<tr>
<td>46</td>
<td>See also selection</td>
<td>227</td>
</tr>
<tr>
<td>47</td>
<td>document structure, improved by</td>
<td>227</td>
</tr>
<tr>
<td>48</td>
<td>See Open Packaging Conventions</td>
<td>227</td>
</tr>
<tr>
<td>49</td>
<td>curve</td>
<td>227</td>
</tr>
</tbody>
</table>

**D**

<table>
<thead>
<tr>
<th>Page</th>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>device</td>
<td>7</td>
</tr>
<tr>
<td>51</td>
<td>device color</td>
<td>7</td>
</tr>
<tr>
<td>52</td>
<td>See color, color space, device color</td>
<td>7</td>
</tr>
<tr>
<td>53</td>
<td>digital signature</td>
<td>7</td>
</tr>
<tr>
<td>54</td>
<td>certificate</td>
<td>7</td>
</tr>
<tr>
<td>55</td>
<td>relationship to</td>
<td>24</td>
</tr>
<tr>
<td>56</td>
<td>store</td>
<td>262</td>
</tr>
<tr>
<td>57</td>
<td>validity</td>
<td>262</td>
</tr>
<tr>
<td>58</td>
<td>co-signature</td>
<td>262</td>
</tr>
<tr>
<td>59</td>
<td>See digital signature, signature definitions</td>
<td>262</td>
</tr>
<tr>
<td>60</td>
<td>multiple signatures</td>
<td>262</td>
</tr>
<tr>
<td>61</td>
<td>namespace</td>
<td>262</td>
</tr>
<tr>
<td>62</td>
<td>See namespace</td>
<td>262</td>
</tr>
<tr>
<td>63</td>
<td>Open Packaging Conventions</td>
<td>259</td>
</tr>
<tr>
<td>64</td>
<td>origin</td>
<td>24</td>
</tr>
<tr>
<td>65</td>
<td>part</td>
<td>24</td>
</tr>
<tr>
<td>66</td>
<td>See parts</td>
<td>24</td>
</tr>
<tr>
<td>67</td>
<td>relationship to</td>
<td>24</td>
</tr>
<tr>
<td>68</td>
<td>relationships</td>
<td>24</td>
</tr>
<tr>
<td>69</td>
<td>request</td>
<td>24</td>
</tr>
<tr>
<td>70</td>
<td>See digital signature, signature definitions</td>
<td>9</td>
</tr>
<tr>
<td>71</td>
<td>described</td>
<td>39, 263</td>
</tr>
<tr>
<td>72</td>
<td>markup example</td>
<td>264</td>
</tr>
<tr>
<td>73</td>
<td>mentioned</td>
<td>22</td>
</tr>
<tr>
<td>74</td>
<td>See namespace</td>
<td>22</td>
</tr>
<tr>
<td>75</td>
<td>relationship to</td>
<td>24</td>
</tr>
<tr>
<td>76</td>
<td>sign by date and time</td>
<td>268</td>
</tr>
<tr>
<td>77</td>
<td>signer name</td>
<td>265</td>
</tr>
<tr>
<td>78</td>
<td>signing intent</td>
<td>266</td>
</tr>
<tr>
<td>79</td>
<td>signing location</td>
<td>268</td>
</tr>
<tr>
<td>80</td>
<td>specifying</td>
<td>264</td>
</tr>
<tr>
<td>81</td>
<td>spot ID</td>
<td>265</td>
</tr>
<tr>
<td>82</td>
<td>spot location</td>
<td>265, 266</td>
</tr>
<tr>
<td>83</td>
<td>signature policy</td>
<td>261, 262</td>
</tr>
<tr>
<td>84</td>
<td>conditions where policy does not apply</td>
<td>261</td>
</tr>
<tr>
<td>85</td>
<td>described</td>
<td>260</td>
</tr>
<tr>
<td>86</td>
<td>markup compatibility impact</td>
<td>260</td>
</tr>
<tr>
<td>87</td>
<td>parts to sign</td>
<td>260</td>
</tr>
<tr>
<td>88</td>
<td>optional</td>
<td>260</td>
</tr>
<tr>
<td>89</td>
<td>See relations to sign</td>
<td>260</td>
</tr>
<tr>
<td>90</td>
<td>as a group, required</td>
<td>261</td>
</tr>
<tr>
<td>91</td>
<td>conditionally required</td>
<td>261</td>
</tr>
<tr>
<td>92</td>
<td>required</td>
<td>261</td>
</tr>
<tr>
<td>93</td>
<td>signing rules</td>
<td>261</td>
</tr>
<tr>
<td>94</td>
<td>signing validity</td>
<td>262</td>
</tr>
<tr>
<td>95</td>
<td>single signature</td>
<td>261</td>
</tr>
<tr>
<td>96</td>
<td>signature spot</td>
<td>261</td>
</tr>
<tr>
<td>97</td>
<td>See digital signature, signature definitions</td>
<td>261</td>
</tr>
<tr>
<td>98</td>
<td>signature status</td>
<td>261</td>
</tr>
<tr>
<td>99</td>
<td>broken digital signature</td>
<td>262</td>
</tr>
<tr>
<td>100</td>
<td>compliant</td>
<td>262</td>
</tr>
<tr>
<td>101</td>
<td>incompliant</td>
<td>262</td>
</tr>
<tr>
<td>102</td>
<td>incompliant digital signature</td>
<td>262</td>
</tr>
<tr>
<td>103</td>
<td>questionable</td>
<td>7</td>
</tr>
<tr>
<td>104</td>
<td>questionable digital signature</td>
<td>7</td>
</tr>
<tr>
<td>105</td>
<td>valid</td>
<td>7</td>
</tr>
<tr>
<td>106</td>
<td>valid digital signature</td>
<td>7</td>
</tr>
<tr>
<td>107</td>
<td>discard control</td>
<td>262</td>
</tr>
<tr>
<td>108</td>
<td>consumer considerations</td>
<td>257</td>
</tr>
<tr>
<td>109</td>
<td>elements</td>
<td>258</td>
</tr>
<tr>
<td>110</td>
<td>markup example</td>
<td>258</td>
</tr>
<tr>
<td>111</td>
<td>namespace</td>
<td>258</td>
</tr>
</tbody>
</table>
fixed payload ............................................. See XPS Document format
FixedDocument part ............................................. 8
FixedDocumentSequence part ............................................. 8
FixedPage part ............................................. 8
font
CFF ............................................. 33, 112
compatibility encoding ............................................. 36
device font ............................................. 98, 118
embedding ............................................. 33–36, 35
 extraction ............................................. 34, 35
language impact on copy and paste ............................................. 47
licensing rights ............................................. 33, 35, 36
obfuscation ............................................. 34, 35
algorithm ............................................. 35
OpenType ............................................. 33
parts ............................................. See parts
rasterization ............................................. 34
relationships ............................................. See relationships, required resource
restricted editing ............................................. See relationships, restricted font
restricted font ............................................. 25
sharing ............................................. 33
subsetting ............................................. 33, 34
TrueType ............................................. 33
TrueType collection (TTC) ............................................. 33, 418
Unicode encoding ............................................. 33
usage of ............................................. 33

geometry
abbreviated syntax ............................................. 79, 89–95
algorithm ............................................. 391
circle ............................................. See geometry, segment, arc
described ............................................. See geometry, segment
described ............................................. 75
figure
closed ............................................. 80, 88, 92
described ............................................. 75
fill control ............................................. 80
markup ............................................. 80
reference to ............................................. 77, 79
segments, composed of ............................................. 75
start point ............................................. 80, 90
stroking of segments ............................................. 75
figures, composed of ............................................. 75
fill algorithm
described ............................................. 78
EvenOdd ............................................. 78
mentioned ............................................. 75
NonZero ............................................. 79
specifying ............................................. 77, 90
filled area ............................................. 77
segment
arc ............................................. 81–84, 91
Bézier curve ............................................. 85, 91
Bézier curve, quadratic ............................................. 87, 91
Bézier curve, smooth ............................................. 91, 93
line ............................................. 86, 90
segment, degenerate ............................................. 298
transformation ............................................. See transformation
usage described ............................................. 76
glyphs ............................................. See text
gradient ............................................. See brush, See path
graphics ............................................. See path
grouping markup ............................................. See canvas

H
hairline ............................................. See stroke, hairline
hyperlink
activation ............................................. 247
addressability
appearance in link targets ............................................. 248
mentioned ............................................. 51, 248
missing name, handling of ............................................. 248
name ............................................. 248
name uniqueness ............................................. 248
of canvas ............................................. 61, 248
of glyphs ............................................. 98, 248
of page ............................................. 53, 248
of path ............................................. 69, 248
of visual brush contents ............................................. 248
page number ............................................. 248
document-level listing ............................................. See hyperlink, target example
mentioned ............................................. 227
overlapping behavior ............................................. 247
source
base URI ............................................. 249
described ............................................. 249
from canvas ............................................. 61, 247
from glyphs ............................................. 98, 247
from path ............................................. 69, 247
inheritance of ............................................. 247
specifying ............................................. 171
support recommended ............................................. 249
target
addressability ............................................. See hyperlink, addressability
document-level definition of ............................................. 52
external ............................................. 247
internal ............................................. 247
link target, specifying as ............................................. 51
missing behavior ............................................. 248
name ............................................. See name
relative target handling ............................................. 248
relative target recommended ............................................. 249
relative to document, at minimum ............................................. 247

I
ICC ............................................. See color, color profile
image

Working Draft 1.1, February 2008
naming of parts
named color
name

mark up compatibility
digital signature, impacted by
document structure, usage in
mentioned
namespace
preprocessing requirements
processing required
property elements, usage with
resource dictionary, usage in
usage of
media size
memory management, device
control

miter
See stroke, line, join

opacity mask
brush, filling with
of canvas
of color
of glyphs
of image brush
of linear gradient brush
of path
of pixel formats
of radial gradient brush
of stroke
of visual brush
pre-multiplied alpha
superluminous colors
transparent brush tiling
value range

ordering
simple

packing model

n-channel color
See color, color space, n-channel

composition effects
computations
described
elements applied to
mask
of canvas
of color
of glyphs
of image brush
of linear gradient brush
of path
of pixel formats
of radial gradient brush
of stroke
of visual brush
pre-multiplied alpha
superluminous colors
transparent brush tiling
value range

ordering
simple

packing model

n-channel color
See color, color space, n-channel

opacity
blending
See color, blending
brush initial opacity
### K. Index

<table>
<thead>
<tr>
<th>Number</th>
<th>Description</th>
<th>Page(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>parsing head-first vs. tail first</td>
<td>253</td>
</tr>
<tr>
<td>2</td>
<td>Open Packaging Conventions</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>interleaving</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>consumer considerations</td>
<td>257</td>
</tr>
<tr>
<td>5</td>
<td>Open Packaging Conventions</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>digital signature</td>
<td>60</td>
</tr>
<tr>
<td>7</td>
<td>Open Packaging Conventions</td>
<td>61</td>
</tr>
<tr>
<td>8</td>
<td>core properties</td>
<td>268</td>
</tr>
<tr>
<td>9</td>
<td>Open Packaging Conventions</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>core properties</td>
<td>64</td>
</tr>
<tr>
<td>11</td>
<td>namespace</td>
<td>65</td>
</tr>
<tr>
<td>12</td>
<td>OpenType font</td>
<td>66</td>
</tr>
<tr>
<td>13</td>
<td>optimization</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>for streaming consumption</td>
<td>73</td>
</tr>
<tr>
<td>15</td>
<td>Conventions, interleaving</td>
<td>74</td>
</tr>
<tr>
<td>16</td>
<td>of composition rules</td>
<td>284</td>
</tr>
<tr>
<td>17</td>
<td>of digital signatures</td>
<td>259</td>
</tr>
<tr>
<td>18</td>
<td>of glyphs</td>
<td>119</td>
</tr>
<tr>
<td>19</td>
<td>of interleaving</td>
<td>254–57</td>
</tr>
<tr>
<td>20</td>
<td>of named element location</td>
<td>228</td>
</tr>
<tr>
<td>21</td>
<td>of pixel placement rules</td>
<td>271</td>
</tr>
<tr>
<td>22</td>
<td>outline</td>
<td>272</td>
</tr>
</tbody>
</table>

### XPS Specification and Reference Guide

<table>
<thead>
<tr>
<th>Number</th>
<th>Description</th>
<th>Page(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>23</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>packaging model</td>
<td>55</td>
</tr>
<tr>
<td>25</td>
<td>page</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>bleed area</td>
<td>53, 54</td>
</tr>
<tr>
<td>27</td>
<td>content area</td>
<td>53, 55, 269</td>
</tr>
<tr>
<td>28</td>
<td>height of</td>
<td>53, 269</td>
</tr>
<tr>
<td>29</td>
<td>height, advisory</td>
<td>51</td>
</tr>
<tr>
<td>30</td>
<td>imageable area</td>
<td>See PrintTicket keywords,</td>
</tr>
<tr>
<td>31</td>
<td>PagelImageableSize</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>layout</td>
<td>See also layout</td>
</tr>
<tr>
<td>33</td>
<td>markup grouping</td>
<td>See canvas</td>
</tr>
<tr>
<td>34</td>
<td>namespace</td>
<td>See namespace</td>
</tr>
<tr>
<td>35</td>
<td>order of pages</td>
<td>50</td>
</tr>
<tr>
<td>36</td>
<td>orientation of</td>
<td>55, 56, 58</td>
</tr>
<tr>
<td>37</td>
<td>reference to</td>
<td>50</td>
</tr>
<tr>
<td>38</td>
<td>root of</td>
<td>53</td>
</tr>
<tr>
<td>39</td>
<td>scaling for print</td>
<td>56</td>
</tr>
<tr>
<td>40</td>
<td>size terminology</td>
<td>55</td>
</tr>
<tr>
<td>41</td>
<td>uniqueness of</td>
<td>51</td>
</tr>
<tr>
<td>42</td>
<td>width of</td>
<td>53, 269</td>
</tr>
<tr>
<td>43</td>
<td>width, advisory</td>
<td>51</td>
</tr>
<tr>
<td>44</td>
<td>paragraph</td>
<td>See document structure, paragraph</td>
</tr>
<tr>
<td>45</td>
<td>part</td>
<td>8</td>
</tr>
<tr>
<td>46</td>
<td>part name</td>
<td>8</td>
</tr>
<tr>
<td>47</td>
<td>parts</td>
<td>See also XPS Document format</td>
</tr>
<tr>
<td>48</td>
<td>core properties</td>
<td>See Open Packaging Conventions, specification</td>
</tr>
<tr>
<td>49</td>
<td>specification</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>digital signature</td>
<td></td>
</tr>
<tr>
<td>51</td>
<td>certificate</td>
<td></td>
</tr>
<tr>
<td>52</td>
<td>part</td>
<td>See Open Packaging Conventions, specification</td>
</tr>
<tr>
<td>53</td>
<td>digital signature origin</td>
<td>See Open Packaging Conventions, specification</td>
</tr>
</tbody>
</table>

### DiscardControl

<table>
<thead>
<tr>
<th>Description</th>
<th>22, 257</th>
</tr>
</thead>
<tbody>
<tr>
<td>DocumentStructure</td>
<td>22, 40, 227</td>
</tr>
<tr>
<td>FixedDocument</td>
<td>21, 25</td>
</tr>
<tr>
<td>FixedDocumentSequence</td>
<td>21, 25</td>
</tr>
<tr>
<td>FixedPage</td>
<td>21, 26</td>
</tr>
<tr>
<td>font</td>
<td>21, 33–37</td>
</tr>
<tr>
<td>ICC profile</td>
<td>22</td>
</tr>
<tr>
<td>image</td>
<td>22</td>
</tr>
<tr>
<td>naming recommendations</td>
<td>22, 26–32</td>
</tr>
<tr>
<td>PrintTicket</td>
<td>22, 37</td>
</tr>
<tr>
<td>remote resource dictionary</td>
<td>22, 37</td>
</tr>
<tr>
<td>thumbnail</td>
<td>22, 32</td>
</tr>
<tr>
<td>thumbnail, package</td>
<td>See Open Packaging Conventions, specification</td>
</tr>
<tr>
<td>XML digital signature</td>
<td>See Open Packaging Conventions, specification</td>
</tr>
<tr>
<td>path</td>
<td></td>
</tr>
<tr>
<td>clipping</td>
<td>See clipping</td>
</tr>
<tr>
<td>described</td>
<td>67, 68</td>
</tr>
<tr>
<td>fill brush</td>
<td>69</td>
</tr>
<tr>
<td>geometry, reference to</td>
<td>69, 72</td>
</tr>
<tr>
<td>opacity</td>
<td>See opacity</td>
</tr>
<tr>
<td>opacity mask</td>
<td>See opacity mask</td>
</tr>
<tr>
<td>shape</td>
<td>See shape</td>
</tr>
<tr>
<td>stroke brush</td>
<td>69</td>
</tr>
<tr>
<td>stroke control</td>
<td>See stroke control</td>
</tr>
<tr>
<td>transformation</td>
<td>See transformation</td>
</tr>
<tr>
<td>usage described</td>
<td>72</td>
</tr>
<tr>
<td>payload</td>
<td>See XPS Document format</td>
</tr>
<tr>
<td>physical imageable size</td>
<td>8, See PrintTicket keywords, PagelImageableSize</td>
</tr>
<tr>
<td>physical media size</td>
<td>8, See PrintTicket keywords, PagelMediaSize</td>
</tr>
<tr>
<td>physical model</td>
<td>See Open Packaging Conventions</td>
</tr>
<tr>
<td>physical organization</td>
<td>See Open Packaging Conventions, interleaving; ZIP</td>
</tr>
<tr>
<td>pixel</td>
<td>See layout, pixel</td>
</tr>
<tr>
<td>pixel snapping</td>
<td>See stroke, line</td>
</tr>
<tr>
<td>PNG</td>
<td>See image, PNG</td>
</tr>
<tr>
<td>portrait orientation</td>
<td>See page, orientation of positioning content</td>
</tr>
<tr>
<td>Print Schema</td>
<td>8</td>
</tr>
<tr>
<td>primary fixed payload root</td>
<td></td>
</tr>
</tbody>
</table>

### Print Schema

<table>
<thead>
<tr>
<th>Description</th>
<th>See PrintTicket</th>
</tr>
</thead>
<tbody>
<tr>
<td>printing</td>
<td></td>
</tr>
<tr>
<td>bleed area</td>
<td>See page, bleed area</td>
</tr>
<tr>
<td>content area</td>
<td>See page, bleed area</td>
</tr>
<tr>
<td>device fonts</td>
<td>See font, device font</td>
</tr>
<tr>
<td>digital signature</td>
<td>See digital signature</td>
</tr>
<tr>
<td>discard control</td>
<td>See discard control</td>
</tr>
<tr>
<td>font, print and preview restricted</td>
<td>See font, licensing rights</td>
</tr>
<tr>
<td>interleaving</td>
<td>See Open Packaging Conventions, interleaving</td>
</tr>
<tr>
<td>layout</td>
<td>See layout, orientation, See PrintTicket keywords, PrintTicket, PrintTicket keywords, resources constraints, See discard control</td>
</tr>
</tbody>
</table>
scaling ........................................... See PrintTicket keywords
PrintTicket ........................................... 9
described ........................................... 37
empty PrintTicket, markup of .................... 254
mapping content levels to parts .................. 38
namespace ........................................... See namespace
parts .................................................. See parts
processing requirements ................................ 38
relationships ........................................... See relationships
PrintTicket keywords
DocumentImpositionColor .......................... 224
ICMRenderingIntent ................................ 224
JobDigitalSignatureProcessing ...................... 262
PrintInvalidSignatures ............................... 262
PrintInvalidSignaturesWithErrorReport .......... 263
PrintOnlyValidSignatures ........................... 263
namespace ........................................... See namespace
PageBlackGenerationProcessing .................... 224
PageBlendColorSpace ................................. 223
PageColorManagement ............................... 222
PageDeviceColorSpaceProfileURI ................... 222
PageDeviceColorSpaceUsage ........................ 222
PageImageableSize ................................ 55, 58, 59, 60
PageMediaSize ........................................ 55, 56, 58, 60, 269
PageOrientation ...................................... 55, 56, 58
PageScaling ........................................... 58–60
producer ................................................ 9
bleed size ............................................... 9, 55
content size .......................................... 9, 55
described ............................................. 17
implementation burden ............................... 407–47
media size ............................................. 9, 55
property ............................................. See XPS Document format, properties

resource dictionary .................................................. See Resource dictionary
resource definition ........................................... See resource dictionary
resource part ............................................. See resource part
remote ............................................. 9, See resource part, remote resource dictionary
resource definition ........................................... 9
described ................................................ 172, 176
key ........................................................... 172
namespace ............................................. See namespace
on canvas .............................................. 61
on geometry ............................................ 77
on glyphs ................................................. 98
on image brush ........................................ 128
on linear gradient brush .............................. 152
on matrix transformation ............................ 186
on path .................................................. 69
on radial gradient brush .............................. 158
on solid color brush .................................. 126
on visual brush ........................................ 131
uniqueness ............................................. 180
language ................................................. 177
locating .................................................. 181
namespace prefixes, interpreting .................... 177
referencing previously-defined resources ...... 176, 177, 180
usefulness of path, glyphs, and canvas as ........... 173
sharing ............................................. See resource part, remote resource dictionary
specifying ............................................. 172, 173–77
where usable ........................................... 173
resource part
color profile ........................................... See color, color profile
tFont ................................................. See font
image ...................................................... See image
reference to ........................................... 24, 26
relationships to ....................................... See relationships
remote resource dictionary ......................... 22, 24, 173, 177
required resource .................................... 24
usage of ................................................ 24
resource reference .................................... 9
described ................................................ 172, 180
example ............................................... 180
scope .................................................... 180
syntax ................................................... 180
rotating content ..................................... See transformation, matrix

R
radial gradient ........................................... See brush, radial gradient brush
raster graphics .......................................... See image
reading order ........................................... See accessibility, reading order
relationship ............................................. 9
StartPart .............................................. 10
relationships ........................................... See also XPS Document format
core properties ....................................... 24
digital signature ....................................... 24
digital signature certificate .......................... 24
digital signature definitions ........................ 24
digital signature origin ............................... 24
DiscardControl ........................................ 24
DocumentStructure ................................... 24
external prohibited ..................................... 24
namespace ........................................... See namespace
PrintTicket ........................................... 24
purpose of ............................................ 24
relationship types, reference table ............... 405
required resource .................................... 24
restricted font ........................................ 25, 34, 36
StartPart ............................................... 21, 25
StoryFragments ....................................... 25
thumbnail ............................................... 25
usage of ............................................... 25
relationships part ..................................... 9
rendering ............................................... See also layout
rules described ........................................ 269–301
required part .......................................... 9
resource ............................................. See resource dictionary, resource definition
resource constraints ................................... See discard control
resource definition ...................................... See resource dictionary, resource definition
description ........................................... 172, 176
declared .............................................. 172, 176
key ........................................................... 172
namespace ............................................. See namespace
on canvas .............................................. 61
on geometry ............................................ 77
on glyphs ................................................. 98
on image brush ........................................ 128
on linear gradient brush .............................. 152
on matrix transformation ............................ 186
on path .................................................. 69
on radial gradient brush .............................. 158
on solid color brush .................................. 126
on visual brush ........................................ 131
uniqueness ............................................. 180
language ................................................. 177
locating .................................................. 181
namespace prefixes, interpreting .................... 177
referencing previously-defined resources ...... 176, 177, 180
usefulness of path, glyphs, and canvas as ........... 173
sharing ............................................. See resource part, remote resource dictionary
specifying ............................................. 172, 173–77
where usable ........................................... 173
resource part
color profile ........................................... See color, color profile
tFont ................................................. See font
image ...................................................... See image
reference to ........................................... 24, 26
relationships to ....................................... See relationships
remote resource dictionary ......................... 22, 24, 173, 177
required resource .................................... 24
usage of ................................................ 24
resource reference .................................... 9
described ................................................ 172, 180
example ............................................... 180
scope .................................................... 180
syntax ................................................... 180
rotating content ..................................... See transformation, matrix

XPS Specification and Reference Guide

K. Index

Working Draft 1.1, February 2008
459
K. Index

1 S
2 scalable vector graphics (SVG) ... See geometry, abbreviated
3 syntax
4 scaling
5 content ... See transformation, matrix
6 for print ... See page: scaling for print
7 scRGB ... See color, color space, scRGB
8 search ... See find
9 section ... See document structure, section
10 segment ... See geometry, segment
11 selection
12 behavior ... 250
13 caret stop ... 98, 119
14 document structure, enabled by ... 227, 250
15 fragment type, behavior depending on ... 250
16 mentioned ... 110, 119, 227
17 order ... 250
18 recommended ... 250
19 shear ... See transformation, matrix
20 signature definitions ... See digital signature
21 signature spot ... 9, See digital signature, signature definitions
22 size terminology ... See page, size terminology
23 skewing content ... See transformation, matrix
24 sRGB ... See color, color space, sRGB
25 starting part ... See relationships, StartPart
26 story ... See document structure, story
27 story fragment ... See document structure, story fragment
28 stream ... 10
29 stretching content ... See transformation, matrix
30 stroke
31 brush ... See path, stroke brush
32 consistent nominal width ... See stroke, hairline
33 contour rendering ... 289
34 dash
35 cap ... 290
36 flat ... 290
37 round ... 291
38 square ... 290
39 style ... 69
40 triangle ... 291
41 offset ... 69
42 overlapping ... 292
43 style ... 69
44 drawing algorithm, symmetry of ... 289
45 edge parallelization ... 289
46 fill rule, independence from ... 299
47 hairline ... 299
48 line
49 cap ... 293
50 end ... 69
51 flat ... 293
52 for dashed stroke ... 293
53 round ... 293
54 square ... 293
55 start ... 69
56 triangle ... 293

XPS Specification and Reference Guide

57 join
58 bevel ... 295
59 miter ... 296–98
60 round ... 294
61 style ... 69
62 miter limit ... 69
63 pixel snapping ... 69
64 thin stroke anti-aliasing behavior ... 271
65 zero-length, avoided ... 81
66 multi-figure path, behavior with ... 299
67 phase control ... 289
68 segments, mixed stroked and non-stroked ... 299
69 thickness ... 69

T

70 table ... See document structure, table
71 table of contents ... See document structure, outline
72 text
73 baseline ... See text, glyph, baseline
74 bidirectional ... See text, style simulation
75 bold ... See text, style simulation
76 clipping ... See clipping
77 fill brush ... 98, 124
78 font ... See also font
79 device font, reference to ... 98, 118
80 reference to ... 98
81 glyph
82 advance width ... 103, 104, 108, 109
83 baseline ... 103
84 black box ... 103
85 black box ... 103
86 cluster map ... 104–7, 108, 109, 110
87 indices
88 described ... 108
89 reference to non-existant glyph ... 108
90 restriction of length ... 108
91 specifying ... 98
92 syntax ... 108
93 metrics ... 103
94 offset ... 104, 108, 109, 114
95 origin ... 103, 104
96 origin, sideways ... 103
97 origin, sideways ... 103
98 side-bearing, bottom ... 104
99 side-bearing, left ... 103
100 side-bearing, right ... 103
101 side-bearing, top ... 104
102 glyphs usage for ... 97
103 italic ... See text, style simulation
104 markup ... 98
105 markup examples ... 121
106 markup optimization ... 119
107 opacity ... See also text, glyph, offset
108 opacity mask ... See opacity mask
109 position ... See also text, glyph, offset
110 horizontal text ... 98