

JEDEC PUBLICATION

Part Model Package Guidelines for Electronic-Device Packages – XML Requirements

JEP30-P100E

(Revision of JEP30-P100D, February 2024)

August 2024

JEDEC SOLID STATE TECHNOLOGY ASSOCIATION



NOTICE

JEDEC standards and publications contain material that has been prepared, reviewed, and approved through the JEDEC Board of Directors level and subsequently reviewed and approved by the JEDEC legal counsel.

JEDEC standards and publications are designed to serve the public interest through eliminating misunderstandings between manufacturers and purchasers, facilitating interchangeability and improvement of products, and assisting the purchaser in selecting and obtaining with minimum delay the proper product for use by those other than JEDEC members, whether the standard is to be used either domestically or internationally.

JEDEC standards and publications are adopted without regard to whether or not their adoption may involve patents or articles, materials, or processes. By such action JEDEC does not assume any liability to any patent owner, nor does it assume any obligation whatever to parties adopting the JEDEC standards or publications.

The information included in JEDEC standards and publications represents a sound approach to product specification and application, principally from the solid state device manufacturer viewpoint. Within the JEDEC organization there are procedures whereby a JEDEC standard or publication may be further processed and ultimately become an ANSI standard.

No claims to be in conformance with this standard may be made unless all requirements stated in the standard are met.

Inquiries, comments, and suggestions relative to the content of this JEDEC standard or publication should be addressed to JEDEC at the address below, or refer to www.jedec.org under Standards and Documents for alternative contact information.

Published by
©JEDEC Solid State Technology Association 2024
3103 10th Street North
Suite 240S
Arlington, VA 22201

JEDEC retains the copyright on this material. By downloading this file the individual agrees not to charge for or resell the resulting material.

PRICE: Contact JEDEC

Printed in the U.S.A.
All rights reserved

DO NOT VIOLATE
THE
LAW!

This document is copyrighted by JEDEC and may not be
reproduced without permission.

For information, contact:

JEDEC Solid State Technology Association
3103 10th Street North
Suite 240S
Arlington, VA 22201
<https://www.jedec.org/contact>

This page intentionally left blank.

PART MODEL PACKAGE GUIDELINE FOR ELECTRONIC-DEVICE PACKAGES - XML REQUIREMENTS

Contents

| | | Page |
|---------|---|------|
| 1 | Scope..... | 1 |
| 1.1 | Purpose | 1 |
| 2 | Applicable Documents..... | 1 |
| 2.1 | JEDEC (www.jedec.org)..... | 1 |
| 2.2 | IPC (www.ipc.org) | 1 |
| 2.3 | JEDEC/IPC (www.jedec.org)..... | 2 |
| 2.4 | ASME (www.asme.org)..... | 2 |
| 3 | Requirements..... | 2 |
| 3.1 | Terms and Definitions | 2 |
| 3.2 | XML Schema Key Terms and Definitions | 2 |
| 4 | Part Model Schema Definition | 3 |
| 4.1 | Part Model - Package Section | 3 |
| 4.2 | Manufacturer Part Number-Array | 5 |
| 4.3 | Linking the MPN to a specific Package Family Data set | 6 |
| 4.3.1 | Linking the Manufacturing Part Number to Package Content | 7 |
| 4.3.2 | Linking the Manufacturing Part Number to Physical Model Content | 8 |
| 4.3.3 | Linking the Manufacturing Part Number to Die Content..... | 9 |
| 5 | Package Section - Package | 10 |
| 5.1 | Package Terminal Position..... | 14 |
| 5.1.1 | Bottom | 15 |
| 5.1.2 | Dual | 16 |
| 5.1.2.1 | Dual Left-to-Right and Back-to-Front..... | 17 |
| 5.1.2.2 | Dual Bottom-to-Upper | 17 |
| 5.1.2.3 | Dual Corners..... | 18 |
| 5.1.3 | Diagonal..... | 19 |
| 5.1.4 | Quad | 20 |
| 5.1.5 | Radial..... | 21 |
| 5.1.6 | Single | 22 |
| 5.1.6.1 | Single Side..... | 23 |
| 5.1.6.2 | Corner - Internal..... | 23 |
| 5.1.6.3 | Corner - External..... | 24 |
| 5.1.7 | Triple..... | 25 |
| 5.1.7.1 | Triple – Multi Side position | 26 |
| 5.1.8 | Upper | 26 |
| 5.2 | Package Outline Style..... | 27 |
| 5.2.1 | Microelectronics Assembly | 28 |
| 5.2.2 | Post Mount..... | 28 |
| 5.3 | Package Terminal Code..... | 29 |
| 5.3.1 | Ball..... | 30 |
| 5.3.2 | Lug..... | 30 |
| 5.3.3 | Flat..... | 31 |
| 5.3.4 | Gull-wing..... | 31 |
| 5.3.5 | Post Terminal..... | 31 |
| 5.3.6 | L-bend..... | 32 |

Contents (Cont'd)

| | | |
|------------------|------------------------------------|----|
| 5.3.7 | Column | 32 |
| 5.3.8 | Surface Terminal..... | 33 |
| 5.3.9 | Pressfit..... | 34 |
| 5.3.10 | Pin..... | 34 |
| 5.3.11 | Wraparound | 35 |
| 5.3.12 | S-Bend..... | 36 |
| 5.3.13 | Through-Hole | 36 |
| 5.3.14 | Terminal Wire..... | 37 |
| 5.3.15 | Screw..... | 37 |
| 5.4 | Package Mounting Preparation | 37 |
| 5.5 | Package Body Direction..... | 38 |
| 5.6 | Mass | 38 |
| 5.7 | Package Shape..... | 39 |
| 5.7.1 | Contour | 41 |
| 5.7.2 | Shape Center..... | 41 |
| 5.7.3 | Package Vertical Dimension..... | 42 |
| 5.8 | Part Height..... | 42 |
| 5.9 | Clearance Region - Array | 44 |
| 5.10 | Assembly Technology | 45 |
| 5.11 | CTE - Array | 46 |
| 5.11.1 | CTE - Graph..... | 47 |
| 5.11.1.1 | Formatting..... | 48 |
| 5.12 | Young Modulus - Array..... | 51 |
| 5.13 | Terminal Groups | 52 |
| 5.13.1 | Terminal Group-Array..... | 53 |
| 5.13.1.1 | Terminal Position | 54 |
| 5.13.1.2 | Terminal..... | 55 |
| 5.13.1.3 | Terminal Material | 56 |
| 5.13.1.4 | Terminal Function | 57 |
| 5.13.1.5 | Terminal Location..... | 57 |
| 5.13.1.5.1 | Standard Array..... | 58 |
| 5.13.1.5.2 | Circular Array | 59 |
| 5.13.1.5.3 | Random Array | 61 |
| 5.13.1.6 | Pattern Group | 61 |
| 5.13.1.6.1 | Pattern Relationship..... | 62 |
| 5.13.1.6.1.1 | Relationship Transformations..... | 63 |
| 5.13.1.6.1.2 | Duplicate..... | 64 |
| 5.13.1.7 | Terminal Shape..... | 65 |
| 5.13.1.7.1 | Ball Types | 73 |
| 5.13.1.7.2 | C-bend | 74 |
| 5.13.1.7.3 | Lug..... | 74 |
| 5.13.1.7.4 | Flat..... | 75 |
| 5.13.1.7.5 | Gull-wing..... | 76 |
| 5.13.1.7.5.1 | Modified Corner..... | 77 |
| 5.13.1.7.5.1.1 | Impacted Terminal | 77 |
| 5.13.1.7.5.1.2 | Impact – to – Terminal Group..... | 78 |
| 5.13.1.7.5.1.2.1 | Apply – to – all - Terminals..... | 78 |
| 5.13.1.7.5.1.2.2 | Rotation | 79 |
| 5.13.1.7.5.1.2.3 | Reflection..... | 79 |
| 5.13.1.7.5.1.3 | Shape Impacted Corner | 80 |

Contents (Cont'd)

| | | |
|-------------------|--|-----|
| 5.13.1.7.5.1.4 | Corner Type | 81 |
| 5.13.1.7.5.2 | Reference Terminal Contour | 82 |
| 5.13.1.7.5.2.1 | Contour Tolerance Type..... | 82 |
| 5.13.1.7.5.3 | Neck..... | 83 |
| 5.13.1.7.5.3.1 | Neck Transition | 83 |
| 5.13.1.7.5.3.1.1 | Tapered | 84 |
| 5.13.1.7.5.3.1.2 | Right Angled | 84 |
| 5.13.1.7.5.3.2 | Configuration..... | 85 |
| 5.13.1.7.5.4 | Dambar | 86 |
| 5.13.1.7.6 | Post (Stud) Terminal | 87 |
| 5.13.1.7.7 | J-bend..... | 88 |
| 5.13.1.7.8 | L-bend..... | 89 |
| 5.13.1.7.9 | Column Types..... | 90 |
| 5.13.1.7.10 | Surface-terminal..... | 91 |
| 5.13.1.7.11 | Pin..... | 92 |
| 5.13.1.7.11.1 | Swage Fastening Pin | 93 |
| 5.13.1.7.11.1.1 | Tail Style | 93 |
| 5.13.1.7.11.1.1.1 | Slotted..... | 94 |
| 5.13.1.7.11.1.1.2 | Turret | 94 |
| 5.13.1.7.12 | Wraparound | 95 |
| 5.13.1.7.13 | S-bend Terminal..... | 96 |
| 5.13.1.7.14 | Through - Hole | 97 |
| 5.13.1.7.15 | J-inverted | 98 |
| 5.13.1.7.16 | Terminal Wire..... | 99 |
| 5.13.1.7.17 | Screw..... | 100 |
| 5.13.1.7.18 | Shoulder | 101 |
| 5.13.1.7.18.1 | Shoulder Transition | 101 |
| 5.13.1.7.18.1.1 | Tapered | 102 |
| 5.13.1.7.18.1.2 | Right Angled | 102 |
| 5.13.1.7.18.2 | Configuration..... | 103 |
| 5.13.1.7.18.3 | TH - Shoulder Cutout | 103 |
| 5.13.1.7.19 | Kinked..... | 104 |
| 5.13.1.7.20 | Terminal End Shape | 104 |
| 5.13.1.7.21 | Terminal Shape versus Dimensions | 105 |
| 5.13.1.7.22 | Terminal Void - Array | 106 |
| 5.13.1.7.22.1 | Standard Array..... | 108 |
| 5.13.1.7.22.2 | Circular Array | 109 |
| 5.13.1.7.22.3 | Void Status..... | 111 |
| 5.13.1.7.22.4 | Random Array..... | 112 |
| 5.13.1.8 | Terminal Span..... | 112 |
| 5.13.1.9 | Terminal Spacing | 113 |
| 5.13.1.10 | Terminal Group to Body Relationship..... | 113 |
| 5.13.1.10.1 | Terminal Group to Body X-Offset | 114 |
| 5.13.1.10.2 | Terminal Group to Body Y-Offset | 117 |
| 5.13.1.10.3 | Terminal Group to Body Z-Offset | 120 |
| 5.13.2 | Terminal Group to Terminal Group Relationship Array..... | 123 |
| 5.13.3 | Terminal Detail Array | 124 |
| 5.13.3.1 | Terminal Detail..... | 125 |
| 5.13.3.1.1 | Terminal Center | 126 |

 Contents (Cont'd)

| | | |
|------------------|--|-----|
| 5.13.3.1.2 | Terminal Status | 128 |
| 5.13.3.2 | Terminal Number Pattern | 138 |
| 5.13.3.2.1 | First Terminal Location | 139 |
| 5.13.3.2.2 | Sequential | 141 |
| 5.13.3.2.3 | Grid | 145 |
| 5.13.3.2.4 | Terminal Detail Exceptions | 149 |
| 5.13.3.2.4.1 | Terminal Index | 150 |
| 5.13.3.2.4.2 | Terminal Number | 151 |
| 5.13.4 | Via Array | 153 |
| 5.13.4.1 | Standard Array | 154 |
| 5.13.4.2 | Circular Array | 155 |
| 5.13.4.3 | Random Array | 155 |
| 5.13.4.4 | Via Detail | 156 |
| 5.14 | Fiducial Marking | 157 |
| 5.14.1 | Fiducial Shape | 157 |
| 5.14.1.1 | Vertex - Array | 158 |
| 5.14.1.1.1 | Edge | 159 |
| 5.14.1.2 | Primitive Fiducial Shape | 160 |
| 5.14.2 | Graphical Format - Array | 160 |
| 5.14.2.1 | Graphical Format | 161 |
| 5.14.2.1.1 | Color Type | 162 |
| 5.15 | Geometric Dimensioning and Tolerancing | 163 |
| 5.15.1 | Datum Array | 164 |
| 5.15.2 | Feature Control Array | 166 |
| 5.15.2.1 | Feature Control Type | 166 |
| 5.15.2.1.1 | Form | 167 |
| 5.15.2.1.2 | Orientation | 167 |
| 5.15.2.1.3 | Profile | 168 |
| 5.15.2.1.4 | Location | 168 |
| 5.15.2.1.5 | Runout | 169 |
| 5.15.3 | Modifier | 170 |
| 5.15.4 | Datum Reference Array | 173 |
| 5.15.5 | Feature Control Placement Array | 174 |
| 5.15.6 | GD&T XML Example | 175 |
| 5.15.7 | GD and T Datum-to-Element Map | 176 |
| 5.16 | Recommended Footprint - Array | 177 |
| 5.16.1 | Interconnect Technology - Array | 178 |
| 5.16.2 | Recommended Pad Or Hole Shape - Array | 178 |
| 5.16.2.1 | Recommended Pad or Hole Shape | 179 |
| 5.16.2.1.1 | Surface Mount | 180 |
| 5.16.2.1.1.1 | Pad Void - Array | 181 |
| 5.16.2.1.1.1.1 | Pad Void Shape | 181 |
| 5.16.2.1.1.1.2 | Pad Void Location | 182 |
| 5.16.2.1.1.1.2.1 | Standard Array | 183 |
| 5.16.2.1.1.1.2.2 | Circular Array | 184 |
| 5.16.2.1.2 | Through Hole | 185 |
| 5.16.2.1.2.1 | Hole Shape | 185 |
| 5.16.2.1.2.2 | Mount Side Pad | 186 |
| 5.16.2.1.2.3 | Internal Pad | 187 |
| 5.16.2.1.2.4 | Opposite Side Pad | 188 |

 Contents (Cont'd)

| | | |
|----------------|---|-----|
| 5.16.2.1.3 | Location | 188 |
| 5.16.2.1.3.1 | Standard Array | 189 |
| 5.16.2.1.3.2 | Circular Array | 189 |
| 5.16.2.1.3.3 | Pad-or-Hole Status..... | 191 |
| 5.16.2.1.3.4 | Random Array | 192 |
| 5.16.2.1.4 | Pattern Groups..... | 192 |
| 5.16.2.1.4.1 | Pattern Relationship..... | 193 |
| 5.16.2.1.4.1.1 | Relationship Transformations..... | 193 |
| 5.16.2.1.5 | Land Pattern Span | 194 |
| 5.16.2.1.6 | Land Pattern Spacing..... | 194 |
| 5.16.2.2 | Pad Group To Pad Group Relationship | 195 |
| 5.16.3 | Thermal Relief - Array | 195 |
| 5.16.3.1 | Spoke Location | 196 |
| 5.16.4 | Assembly Outline Layer | 196 |
| 5.16.5 | Conductive Area - Array | 197 |
| 5.16.5.1 | Conductive Area Void - Array | 197 |
| 5.16.5.1.1 | Conductive Area Void Shape | 198 |
| 5.16.5.1.2 | Conductive Area Void Location | 198 |
| 5.16.6 | Placement Outline..... | 199 |
| 5.16.7 | Keep-in Layer - Array | 200 |
| 5.16.7.1 | Restrictive Layer | 201 |
| 5.16.7.2 | From Outer Layer..... | 201 |
| 5.16.7.3 | Restriction | 202 |
| 5.16.8 | Keepout Region - Array..... | 202 |
| 5.16.9 | Soldermask Layer - Array..... | 203 |
| 5.16.9.1 | Soldermask Gang Relief- Array..... | 203 |
| 5.16.9.1.1 | Terminal Center - Array..... | 204 |
| 5.16.9.2 | Soldermask Shape..... | 205 |
| 5.16.9.2.1 | SM - Shape Location..... | 206 |
| 5.16.9.2.1.1 | Standard Array..... | 206 |
| 5.16.9.2.1.2 | Circular Array | 207 |
| 5.16.9.2.1.3 | SM - Shape Status | 207 |
| 5.16.9.2.1.3.1 | SM – Shape Center - Array | 208 |
| 5.16.9.2.1.3.2 | SM - Shape Index | 209 |
| 5.16.9.2.1.4 | Random Array | 210 |
| 5.16.9.2.2 | Pattern Group | 210 |
| 5.16.9.2.2.1 | RelationshipTransformations..... | 211 |
| 5.16.9.2.2.2 | RelationshipTransformations..... | 211 |
| 5.16.9.2.3 | SM – Shape Pattern Span..... | 212 |
| 5.16.9.2.4 | SM – Shape Pattern Spacing | 212 |
| 5.16.10 | Pastemask Layer- Array | 213 |
| 5.16.10.1 | Aperture Shape..... | 214 |
| 5.16.10.1.1 | Aperture Location..... | 215 |
| 5.16.10.1.1.1 | Standard Array | 215 |
| 5.16.10.1.1.2 | Circular Array | 216 |
| 5.16.10.1.1.3 | Status..... | 217 |
| 5.16.10.1.1.4 | RandomArray..... | 217 |
| 5.16.10.1.2 | Pattern Groups..... | 218 |
| 5.16.10.1.2.1 | Relationship Transformations..... | 218 |

Contents (Cont'd)

| | | |
|-----------------------|---|-----|
| 5.16.10.1.2.2 | Duplicate | 219 |
| 5.16.10.2 | Pastemask Layer To Pastemask Layer Relationship | 219 |
| 5.17 | Physical Model | 220 |
| 6 | Package Section - Die-Array | 221 |
| 6.1 | Die Zones | 223 |
| 6.2 | Stepping Distance | 225 |
| 6.3 | Mass | 226 |
| 6.4 | Part Height | 226 |
| 6.5 | Terminal Groups | 227 |
| 6.5.1 | Terminal Group - Array | 228 |
| 6.5.1.1 | Terminal Group | 229 |
| 6.5.1.1.1 | Terminal Pattern | 230 |
| 6.5.1.1.2 | Pattern Group | 234 |
| 6.5.1.2 | Relationship – to – Active Zone | 241 |
| 6.5.1.2.1 | X-Offset | 242 |
| 6.5.1.2.2 | Y-Offset | 245 |
| 6.5.1.3 | Terminal Group to Terminal Group Relationships | 248 |
| 6.5.2 | Terminal Regions - Array | 248 |
| 6.5.2.1 | Area Selection | 249 |
| 6.5.2.2 | Terminal Selection | 249 |
| 6.5.2.2.1 | Select | 250 |
| 6.5.3 | Terminal Materials - Array | 251 |
| 6.5.4 | CTE - Array | 252 |
| 6.5.5 | Terminal Shape - Array | 253 |
| 6.5.6 | Terminal Specification - Array | 254 |
| 6.5.7 | Terminal Detail - Array | 254 |
| 6.5.7.1 | Terminal Detail | 255 |
| 6.5.7.2 | Terminal Number Pattern | 257 |
| 6.5.7.2.1 | First Terminal Location | 258 |
| 6.5.7.2.2 | Terminal Details Exception | 260 |
| Annex A (informative) | Shape Definitions & Dimensions | 266 |
| A.1 | Shape Definitions | 266 |
| A.1.1 | Rectangle | 266 |
| A.1.2 | Rounded Rectangle | 267 |
| A.1.3 | Rounded Concave | 267 |
| A.1.4 | Rectangle with Shoulder | 268 |
| A.1.5 | Rounded Rectangle with Tab | 269 |
| A.1.6 | Modified Rectangle | 270 |
| A.1.7 | Modified Rectangle with Shoulder | 272 |
| A.1.8 | Modified Rectangle with Tab | 272 |
| A.1.9 | Circle | 273 |
| A.1.10 | Circle with Shoulder | 273 |
| A.1.11 | Circle with Tab | 273 |
| A.1.12 | D-Shape | 274 |
| A.1.13 | Double-D | 274 |
| A.1.14 | Regular Polygon | 275 |
| A.1.15 | Segment | 275 |
| A.1.16 | Para-truncated Circle | 276 |
| A.1.17 | Rounded Diamond | 276 |
| A.1.18 | Rounded Chamfered Diamond | 276 |

Contents (Cont'd)

| | | |
|------------|---|-----|
| A.1.19 | Isosceles Trapezoid | 277 |
| A.1.20 | Castellated Shape | 277 |
| A.1.21 | Castellated D-Shape Shape | 277 |
| A.1.22 | Contour | 278 |
| A.1.23 | Reference Terminal Contour | 280 |
| A.1.24 | Modified Corner | 280 |
| A.1.24.1 | Impacted Terminal | 281 |
| A.1.24.2 | Impact – to – Terminal Group | 281 |
| A.1.24.2.1 | Apply – to – all - Terminals | 282 |
| A.1.24.2.2 | Rotation | 282 |
| A.1.24.2.3 | Reflection | 283 |
| A.1.24.3 | Shape Impacted Corner | 284 |
| A.1.24.4 | Corner Type | 285 |
| A.2 | Terminal Dimensions | 286 |
| A.2.1 | Ball | 286 |
| A.2.1.1 | Collapsing Ball | 286 |
| A.2.1.2 | Non-collapsing Ball | 286 |
| A.2.2 | C-Bend | 287 |
| A.2.3 | Lug | 288 |
| A.2.4 | Flat Terminal | 288 |
| A.2.5 | Gull-wing Terminal | 289 |
| A.2.6 | J-Bend | 291 |
| A.2.7 | L-bend | 292 |
| A.2.8 | Column | 293 |
| A.2.9 | Surface Terminal | 295 |
| A.2.10 | Pressfit | 296 |
| A.2.11 | Pin | 297 |
| A.2.12 | Wraparound | 298 |
| A.2.14 | Through-Hole | 299 |
| A.2.15 | J-Inverted | 301 |
| A.2.16 | Wire | 302 |
| A.2.17 | Screw | 302 |
| A.3 | Land Pattern Shapes | 303 |
| A.3.1 | Rectangle | 303 |
| A.3.2 | Rounded Rectangle | 303 |
| A.3.3 | Half Rounded Rectangle | 304 |
| A.3.4 | Modified Rectangle | 304 |
| A.3.5 | Circle | 305 |
| A.3.6 | D-Shape | 305 |
| A.3.7 | Rounded Rectangle D-Shape | 306 |
| A.3.8 | Double-D | 306 |
| A.3.9 | Contour | 306 |
| A.4 | Value Set Types | 306 |
| A.4.1 | Value Type | 306 |
| A.4.2 | Value Set Type | 307 |
| A.4.3 | Unspecified Value Set Type | 308 |
| A.4.4 | Dimensional Value Set Type | 309 |
| A.4.5 | Unspecified and Uncontrolled Value Set Type | 310 |
| A.4.6 | Unspecified Dimensional Value Set Type | 311 |

 Contents (Cont'd)

| | | |
|-----------------------|---|-----|
| A.4.7 | Unspecified and Uncontrolled Value Set Type | 312 |
| Annex B (informative) | Differences between JEP30-P100 and its predecessors | 313 |

 Tables

| | |
|---|-----|
| Table 1 - JESD30 Table and Section References for Package Elements..... | 13 |
| Table 2 - Package Shape versus Dimensions | 40 |
| Table 3 - Terminal Circular Array Elements Definition | 60 |
| Table 4 - Ball Dimensions | 73 |
| Table 5 - C-bend Dimensions..... | 74 |
| Table 6 - Lug Dimensions | 74 |
| Table 7 - Flat Terminal Type Dimensions..... | 75 |
| Table 8 - Gull-wing Dimensions | 76 |
| Table 9 - Post (Stud) Terminal Dimensions..... | 87 |
| Table 10 - J-bend Dimensions | 88 |
| Table 11 - L-bend Dimensions | 89 |
| Table 12 - Column Type Dimensions | 90 |
| Table 13 - Surface-terminal Dimensions | 91 |
| Table 14 - Pin Dimensions | 92 |
| Table 15 - Wraparound Dimensions..... | 95 |
| Table 16 - S-bend Dimensions..... | 96 |
| Table 17 - Through - Hole Dimensions..... | 97 |
| Table 18 - J-inverted Dimensions..... | 98 |
| Table 19 - Terminal Wire Dimensions | 99 |
| Table 20 - Screw Dimensions | 100 |
| Table 21 - Terminal Shape versus Dimensions | 105 |
| Table 22 - Void Circular Array Elements Definition..... | 110 |
| Table 23 - Terminal Group to Package Body X-Offset..... | 116 |
| Table 24 - Terminal Group to Package Body Y-Offset..... | 119 |
| Table 25 - Terminal Group to Package Body Z-Offset..... | 122 |
| Table 26 - Location Relative to Package Center | 140 |
| Table 27 - Terminal Number Sequential Pattern | 141 |
| Table 28 - Terminal Number Patterns | 142 |
| Table 29 - Grid Terminal Numbering Pattern..... | 147 |
| Table 30 - Pad-or-Hole Circular Array Elements Definition..... | 190 |
| Table 31 - Pattern Details for the UCle Instantiation | 231 |
| Table 32 - Pattern Group Construction..... | 238 |
| Table 33 - Terminal Group to Active Zone X-Offset..... | 244 |
| Table 34 - Terminal Group to Active Zone Y-Offset..... | 247 |
| Table 35 - Location Relative to Selection Center..... | 259 |

Figures

| | |
|--|-----|
| Figure 1 - Solder Mask Defined Pad | 73 |
| Figure 2 - Non Solder Mask Defined Pad | 73 |
| Figure 3 - Surface terminal With-Opening, plus Screw-Clearance Hole | 106 |
| Figure 4 - Flat Terminal with Hole | 106 |
| Figure 5 - Flat terminal with different shape Voids..... | 107 |
| Figure 6 - Terminal Group to Terminal Group Span and Spacing..... | 123 |
| Figure 7 - SOIC..... | 127 |
| Figure 8 - SOIC with Terminal 5 Missing | 128 |
| Figure 9 - SOIC with Terminal 5 Deleted..... | 129 |
| Figure 10 – Part with Special Terminal Pattern | 131 |
| Figure 11 – Terminal Numbering Pattern Samples..... | 144 |
| Figure 12 – A Multi grid Package | 146 |
| Figure 13 – Grid Array with Terminal Numbering | 146 |
| Figure 14 – Periphery Grid Array with Inner Array Matrix | 151 |
| Figure 15 – Example showing Datum and Feature Control Frames | 164 |
| Figure 16 – Constrained Degrees of Freedom for Primary Datum Features..... | 165 |
| Figure 17 – Modifying Symbols | 172 |
| Figure 18 – Sample for Representation in an XML Structure..... | 175 |
| Figure 19 - Zone shapes for Physical Dies on a Wafer..... | 225 |
| Figure 20 - Complex Array of 12 instantiations of a UCle Function | 227 |
| Figure 21 - Single UCle Instantiation in South Channel Inst 1 | 231 |
| Figure 22 - Power Nodes for South Channel Inst 1 & 2 | 231 |
| Figure 23 - Missing Terminal for Terminal Patterns..... | 256 |
| Figure 24 - Terminal Numbering Sequence for Bank 1..... | 261 |
| Figure 25 - An Array of Banks in a Zone | 263 |
| Figure 26 - An Array of Zones in One District..... | 265 |

This page intentionally left blank

PART MODEL PACKAGE GUIDELINE FOR ELECTRONIC-DEVICE PACKAGES - XML REQUIREMENTS

(From JEDEC Board Ballots JCB-24-29, JCB-17-48, JCB-23-10, JCB-23-27, JCB-23-33, and JCB-24-08 formulated under the cognizance of the JC-11 Committee on Mechanical Standardization.)

1 Scope

This standard establishes the requirements for exchanging part data between part manufacturers and their customers for electrical and electronic products. This standard applies to all forms of electronic parts. It forms part of the Part Model XML Schema, which covers the parental structure for the electrical, physical, thermal, assembly process classification data along with materials and substances that may be present in the supplied product or sub-products. This Guideline specifically focuses on the Package sub-section of the Part Model.

All releases of the *Package* sub-schema must be under the umbrella of the Part model Schema to ensure that the Part model schema is referencing the correct version of the *Package* sub-schema. In addition, this will enable the *Package* sub-schema to connect to the Manufacturer Part Number and the Manufacturer of the Part.

1.1 Purpose

This standard is intended to benefit part manufacturers and their customers by providing consistency and efficiency to the transfer of part data from part manufacturer to customers. This standard specifically covers data applicable to the package definition of the device.

2 Applicable Documents

The following documents form a part of this standard to the extent specified herein. The revision of the document in effect at the time of solicitation shall take precedence.

2.1 JEDEC (www.jedec.org)

JESD30L, *Descriptive Designation System for Electronic-device Packages*

JEP30, *Part Model Guidelines for Electronic-Device Packages – XML Requirements*

JEP95, *JEDEC Registered and Standard Outlines for Solid State Products*

JEP30-10, *Part Model Schema*

JEP30-P101, *Part Model Package Schema*

JEP30-D10, *Part Model Schema Types Dictionary* (Required to support the Part Model Schema and each of its sectional sub-schemas.)

SPP-010 Standard Procedures and Practices for Grid Array Terminal Position Numbering

2.2 IPC (www.ipc.org)

IPC-T-50, *Terms and Definitions for Interconnecting and Packaging Electronic Circuits*

2.3 JEDEC/IPC (www.jedec.org)

J-Std-609, *Marking, Symbols, and Labels of Leaded and Lead-Free Terminal Finished Materials Used in Electronic Assembly*

2.4 ASME (www.asme.org)

ASME Y14.5-2009 Dimensioning and Tolerancing

3 Requirements

The following terms and definitions are applicable to this XML Schema.

3.1 Terms and Definitions

All definitions and terms associated with the physical aspect of the part are in accordance with JESD30H or later versions. The physical details of the part are defined in the [PackageSection](#) of the XML Schema.

All common Terms and Definitions that are used by more than one sectional sub-schema, such as any of the Electrical, Package, Environmental, Assembly Process Classification, are defined in the “Part Model Schema Types Library”.

All other definitions and terms necessary to define the schema, are defined by this standard and included below.

Part Model: A Part Model is a data representation described in an XML file that conforms to the rules and structure of the Part Model XML Schema. Companies who use the Part Model XML Files and claim compliance to JEDEC, must ensure that their Part Model XML file conforms to the specific released version of the Part Model XML Schema released by JEDEC.

Section 4 will define the outline of the structure of the Package XML Schema. Specific components of the XML Schema and their hierarchy are specifically controlled by the JC-11 Standards Committee who retain the expertise for these structures.

The [PackageSection](#) of the schema forms part of the Part Model XML Schema and is not intended to act as a standalone schema. In addition, there is a “Part Model Schema Types Library” XML Schema, which is a common set of xml structures shared across the Part Model XML Schema and all its sub-section schemas.

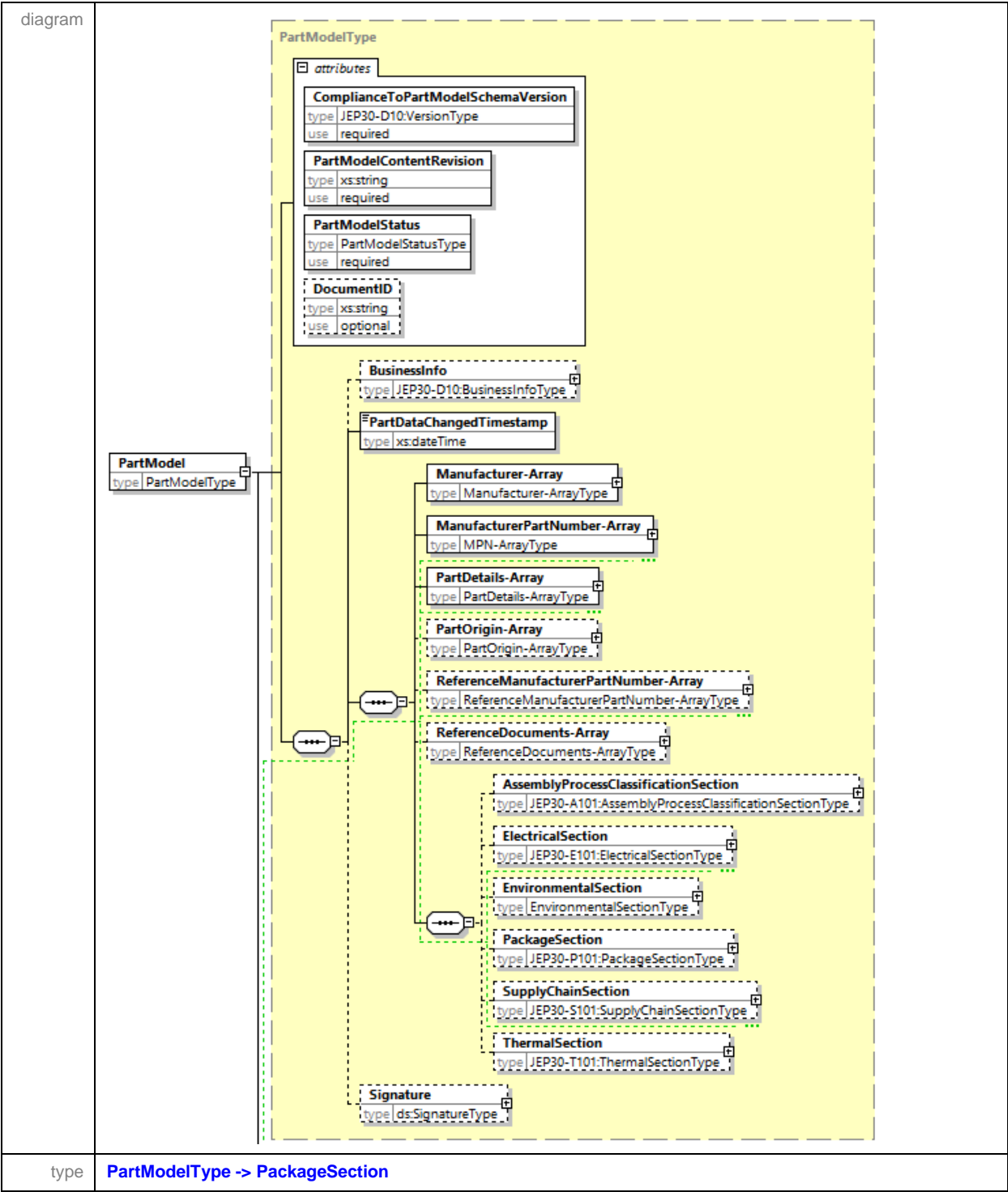
3.2 XML Schema Key Terms and Definitions

Reference JEP30 for details of the "XML Schema Key Terms and Definitions".

4 Part Model Schema Definition

The following section describes the XML Schema structure.

4.1 Part Model - Package Section



4.1 Part Model - Package Section (cont'd)

The [PartModelType](#) belongs to the “Part Model XML Schema”. The [PackageSection](#) belongs to the “Part Model Package XML Schema”. The primary purpose of the Part Model Schema is to provide the structure for identifying unique parts (Manufacturer and MPN), and the structure to include the sub schemas which define the part details, as outlined in the JEP30 publication.

This document covers the [PackageSection](#), which is referenced from its parent’s structure, the [PartModel](#). The contents under the [PackageSection](#) is tied to the Manufacturer’s name and Manufacturer’s part number.

The [ComplianceToPartModelSchemaVersion](#) indicates the version of the Schema to which the XML file is to be validated against. All new releases to this document or XML Schema is governed by the rules outlined in the JEPXX, and must be release in sync with the Part Model.

“Each time that a Sub-schema gets updated, then the part model version also gets updated in order to release that Sub-schema under the umbrella of the Part model. This is because the Part Model must now reference the new version of Sub-schema, since all subschemas have their own version number. The parent schema includes them by referring to a precise version, so a version bump in the subschema requires a version bump in the parent only at the time of release of the Parent.”

The [PartModelContentRevision](#) indicates the revision of the data for the Part that is submitted in the XML file. This enables the Component Manufacturer to provide a new XML file for a Part each time they wish to upgrade a new set of data for a part, in any of the sub-sections such as this [PackageSection](#).

4.2 Manufacturer Part Number-Array

| | |
|---------|--|
| path | PartModel/ManufacturerPartNumber-Array. |
| diagram | <p>The diagram illustrates the XML Schema Definition (XSD) for the ManufacturerPartNumber-Array. It is an array of ManufacturerPartNumbers (type MPN-ArrayType). Each ManufacturerPartNumbers element contains a ManufacturerPartNumbersType (type ManufacturerPartNumbersType). This type includes the following elements:</p> <ul style="list-style-type: none"> ID (type xs:string): A required identifier. PartNumberSeries (type PartNumberType): A sequence of part number series (0..∞). OrderedPartNumber (type OrderablePartNumberType): An optional ordered part number (0..∞). FuturePart (type FuturePartType): An optional future part (0..∞). StandardsIdentifier (type StandardsIdentifierType): An optional standards identifier (0..∞). ManufacturerID (type xs:string): An optional manufacturer ID. ManufacturerSignatureDigest (type JEP30-D10:SignatureDigestLinkType): An optional signature digest. ManufacturerPartNumbersIdentitySignature (type ds:SignatureType): An optional identity signature. <p>A constraints box is also present at the bottom of the diagram.</p> |
| type | MPN-ArrayType , ManufacturerPartNumbersType , PartNumberType , OrderablePartNumber-ArrayType , JEP30-D10:SignatureDigestLinkType , ds:SignatureType . |

The [ManufacturerPartNumber-Array/ManufacturerPartNumbers](#) provides the definition of the part number or a specific Standard, so that it can be connected to the technical specification details in the [PackageSection](#) via the [PartDetails-Array](#) section.

4.3 Linking the MPN to a specific Package Family Data set

The linking of the Parts to its technical data is done via the [PartDetails-Array](#) section as outline in the JEP30 - Part Model Guidelines for Electronic-Device Packages – XML Requirements. This consists of two sections called [PartsSelection-Array](#) and [Association-Array](#) which defines the relationship between identifying the specific set of parts and how they are associated with the supply chain content. Reference the JEP30 parent document for more details on this association.

| | |
|--------------------------------------|---|
| path | PartModel/PartDetails-Array/PartDetails/Association-Array/Association/Package-Array |
| diagram at the Association level | <p>The diagram shows a Package-Array (type: PackageAssociation-ArrayType) connected to a PackageAssociation-ArrayType container. This container includes three sub-arrays: Package (type: PackageAssociationType, 0..∞), PhysicalModel (type: PhysicalModelAssociationType, 0..∞), and Die (type: DieAssociationType, 0..∞).</p> |
| type | PackageAssociation-ArrayType , PackageAssociationType , PhysicalModelAssociationType , DieAssociationType . |
| diagram at the Package Section level | <p>The diagram shows a PackageSection (type: PackageSectionType) connected to a PackageSectionType container. This container includes two sub-sections: Package-Array (type: Package-ArrayType) and Die-Array (type: Die-ArrayType). The Package-Array section contains Package (type: PackageType, 1..∞) and PhysicalModel (type: PhysicalModelType, 0..∞), with a constraints block. The Die-Array section contains Die (type: DieType, 1..∞) and a constraints block. Dashed green lines indicate cross-section constraints between the Package and Die sections.</p> |
| type | PackageSectionType , Package-ArrayType , PackageType , PhysicalModelType , Die-ArrayType , DieType . |

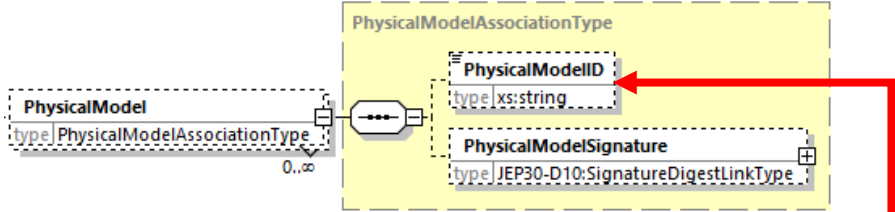
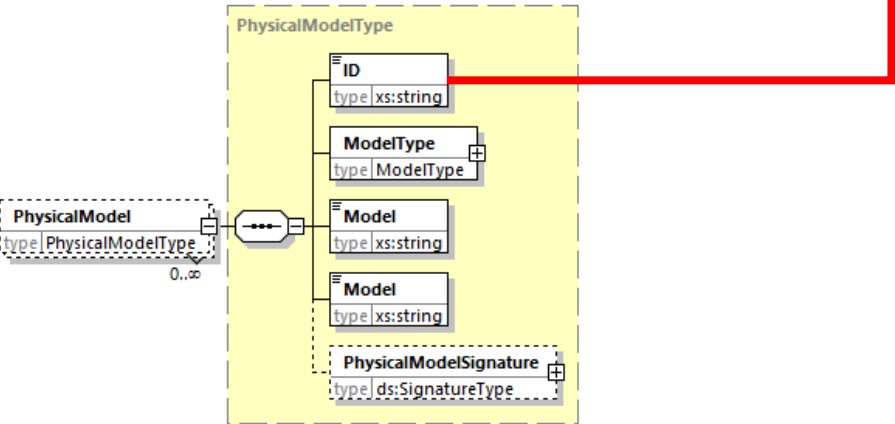
The [Package-Array](#) content is now sub-grouped into 2 major sections as shown in the diagram. This enables each section to be digitally signed independently of each other. The linkage between the 2 sections plus the [Die](#) is shown below.

4.3.1 Linking the Manufacturing Part Number to Package Content

| | |
|--|--|
| path | PartModel/PartDetails-Array/PartDetails/Association-Array/Association/Package-Array/Package |
| diagram at the Package Association level | <p>The diagram shows a PackageAssociationType container (yellow dashed box) containing a PackageID element (type xs:string) and a PackageSignature element (type JEP30-D10:SignatureDigestLinkType). To the left, a Package element (type PackageAssociationType) is shown with a cardinality of 0..∞. A red arrow originates from the PackageID element and points to the ID element in the PackageType diagram below.</p> |
| type | PackageAssociationType , JEP30-D10:SignatureDigestLinkType . |
| path | PartModel/PackageSection/Package-Array/Package |
| diagram at the Package level | <p>The diagram shows a PackageType container (yellow dashed box) containing an ID element (type xs:string), a DescriptiveDesignation element (type xs:string), a ManufacturerPackageCode element (type xs:string), and a PackageSignature element (type ds:SignatureType). To the left, a Package element (type PackageType) is shown with a cardinality of 1..∞. A red arrow originates from the ID element and points to the PackageID element in the PackageAssociationType diagram above.</p> |
| type | PackageType , ds:SignatureType , ... |

The [PackageID](#) references the [Package/ID](#) under the [PackageSection/Package-Array](#). This is enforced by the key named as [PackageKey](#) that is assigned to the [Package/ID](#) element, which is referenced by the [PackageID](#) which has a KeyRef that refers to the [JEP30-P101:PackageKey](#).

4.3.2 Linking the Manufacturing Part Number to Physical Model Content

| | | |
|---|--|--|
| path | PartModel/PartDetails-Array/PartDetails/Association-Array/Association/Package-Array/PhysicalModel | |
| diagram at the Physical Model Association level |  <p>The diagram shows a dashed box labeled PhysicalModel with a note <code>type PhysicalModelAssociationType</code> and a cardinality of <code>0..∞</code>. This is connected to a solid box labeled PhysicalModelAssociationType. Inside this box, there is an PhysicalModelID element with <code>type xs:string</code> and a PhysicalModelSignature element with <code>type JEP30-D10:SignatureDigestLinkType</code>. A red arrow points from the PhysicalModelID element to the ID element in the diagram below.</p> | |
| type | PhysicalModelAssociationType , JEP30-D10:SignatureDigestLinkType . | |
| path | PartModel/PackageSection/Package-Array/PhysicalModel | |
| diagram at the Physical Model level |  <p>The diagram shows a dashed box labeled PhysicalModel with a note <code>type PhysicalModelType</code> and a cardinality of <code>0..∞</code>. This is connected to a solid box labeled PhysicalModelType. Inside this box, there is an ID element with <code>type xs:string</code>, a ModelType element with <code>type ModelType</code>, two Model elements with <code>type xs:string</code>, and a PhysicalModelSignature element with <code>type ds:SignatureType</code>. A red arrow points from the ID element to the PhysicalModelID element in the diagram above.</p> | |
| type | PhysicalModelType , ModelType , ds:SignatureType . | |

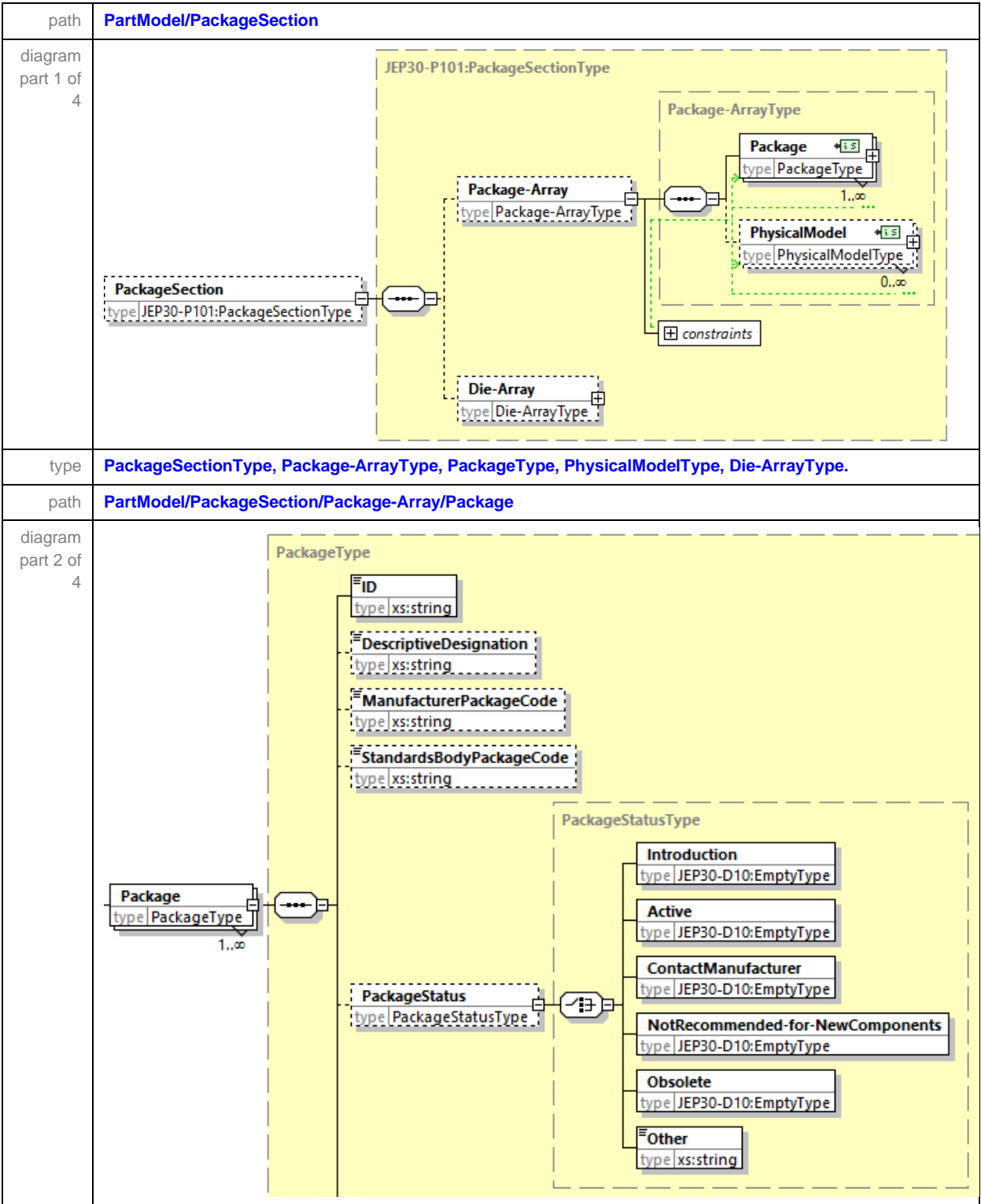
The [PhysicalModelID](#) references the [PhysicalModel/ID](#) under the [PackageSection/Package-Array](#). This is enforced by the key named as [PhysicalModelKey](#) that is assigned to the [PhysicalModel/ID](#) element, which is referenced by the [PhysicalModelID](#) which has a KeyRef that refers to the [JEP30-P101:PhysicalModelKey](#).

4.3.3 Linking the Manufacturing Part Number to Die Content

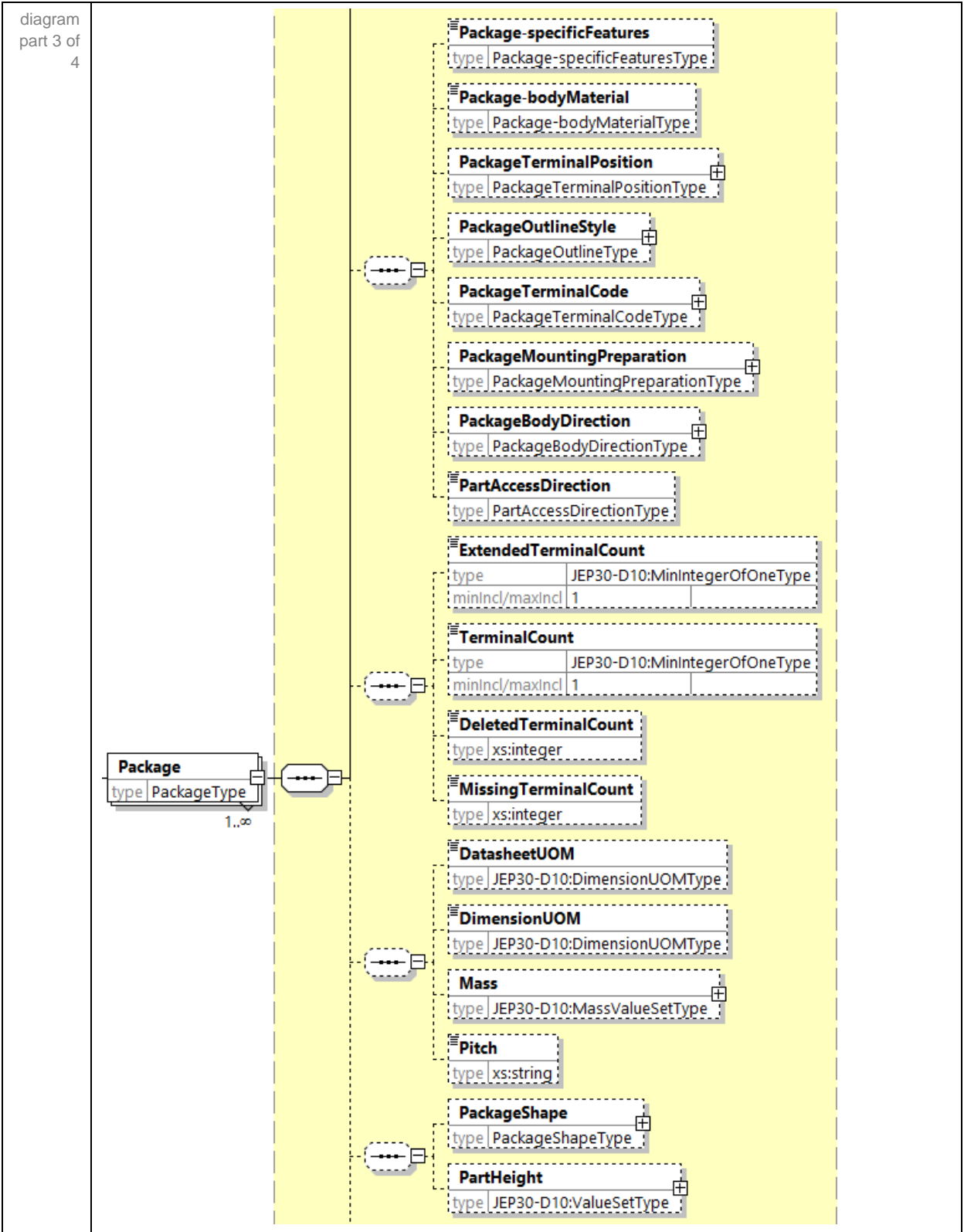
| | | |
|---|---|--|
| path | PartModel/PartDetails-Array/PartDetails/Association-Array/Association/Package-Array/Die | |
| diagram at the Physical Model Association level | <p>The diagram shows a dashed box labeled DieAssociationType. Inside, there is a DieID element with type <code>xs:string</code> and a DieSignature element with type <code>JEP30-D10:SignatureDigestLinkType</code>. A red arrow points from the DieID element to the DieSignature element. To the left of the DieAssociationType box, there is a Die element with type <code>DieAssociationType</code> and a cardinality of <code>0..∞</code>.</p> | |
| type | DieAssociationType, JEP30-D10:SignatureDigestLinkType. | |
| path | PartModel/PackageSection/Die-Array/Die | |
| diagram at the Physical Model level | <p>The diagram shows a dashed box labeled DieType. Inside, there is an ID element with type <code>xs:string</code>, followed by three unnamed elements with type <code>xs:string</code> and cardinality <code>1..∞</code>. Below these are FiducialMarking (type <code>FiducialMarkingType</code>), TerminalGroups (type <code>DieTerminalGroupsType</code>), and DieSignature (type <code>ds:SignatureType</code>). A red arrow points from the ID element to the DieSignature element. To the left of the DieType box, there is a Die element with type <code>DieType</code> and a cardinality of <code>1..∞</code>.</p> | |
| type | DieType, FiducialMarkingType, DieTerminalGroupsType, ds:SignatureType. | |

The *DieID* references the *Die/ID* under the *PackageSection/Die-Array*. This is enforced by the key named as *DieKey* that is assigned to the *Die/ID* element, which is referenced by the *DieID* which has a KeyRef that refers to the *JEP30-P101:DieKey*.

5 Package Section - Package



5 Package Section – Package (cont'd)



5 Package Section – Package (cont'd)

| | |
|------------------------------------|--|
| <p>diagram part 4 of 4</p> | |
| <p>type</p> | <p>PackageType, PackageStatusType, JEP30-D10:EmptyType, Package-specificFeaturesType, Package-bodyMaterialType, PackageTerminalPositionType, PackageOutlineType, PackageTerminalCodeType, PackageMountingPreparationType, PackageBodyDirectionType, PartAccessDirectionType, JEP30-D10:MinIntegerOfOneType, JEP30-D10:DimensionUOMType, JEP30-D10:MassValueSetType, PackageShapeType, JEP30-D10:ValueSetType, ClearanceRegion-ArrayType, AssemblyTechnologyType, CTE-ArrayType, YoungsModulus-ArrayType, TerminalGroupsType, FiducialMarkingType, JEP30-D10:GDAndTType, GDandTDatum-to-ElementMapType, RecommendedFootprint-ArrayType, ds:SignatureType.</p> |

The [DescriptiveDesignation](#) is described in JESD30, “Descriptive Designation System for Electronic-device Packages and Footprints”. The [ManufacturerPackageCode](#) is a non-standardized package Code that is assigned to the Package by the component manufacturer. The definitions of the [ExtendedTerminalCount](#), [TerminalCount](#), [DeletedTerminalCount](#), [MissingTerminalCount](#) and [Package/Pitch](#) are also described in the JESD30 document.

The [PackageStatus](#) is an optional attribute that determines the status of the Package.

5 Package Section – Package (cont'd)

The enumerate values of the *JEP30-D10:DimensionUOMType* are *nm*, *um*, *mm*, *m*, *in* and *mil*.

The enumerate values of the above lists of *Package-specificFeatures*, *Package-bodyMaterial*, and *PartAccessDirection*, are contained in JESD30, as indicated in Table 1.

Table 1 - JESD30 Table and Section References for Package Elements

| Enumerated Value List | JESD30 Tables / Section |
|----------------------------|--|
| Package-specific Features | Codes for package-specific features table |
| Package-body Material Type | Prefixes for predominant package-body material table |
| Part Access Direction | Part access Direction section |

5.1 Package Terminal Position

| | | |
|---------|--|--|
| path | PartModel/PackageSection/Package-Array/Package/PackageTerminalPosition. | |
| diagram | <p>The diagram illustrates the PackageTerminalPosition type as a disjoint union of several subtypes, all of which inherit from PackageTerminalPositionType. The subtypes are: Axial (type: JEP30-D10:EmptyType), Bottom (type: BottomUpperPositionConfigurationType), Dual (type: DualPositionConfigurationType), End (type: JEP30-D10:EmptyType), Diagonal (type: DiagonalCornerConfigurationType), Internal (type: JEP30-D10:EmptyType), MixedPosition (type: JEP30-D10:EmptyType), Quad (type: QuadPositionConfigurationType), Radial (type: RadialPositionConfigurationType), Single (type: SinglePositionConfigurationType), Triple (type: TriplePositionConfigurationType), Upper (type: BottomUpperPositionConfigurationType), and ZigZag (type: JEP30-D10:EmptyType). The PackageTerminalPosition type is shown as a dashed box containing a union symbol (a circle with a vertical line and a horizontal line) and the text PackageTerminalPosition (type: PackageTerminalPositionType).</p> | |
| type | PackageTerminalPositionType, JEP30-D10:EmptyType, BottomUpperPositionConfigurationType, DualPositionConfigurationType, DiagonalCornerConfigurationType, QuadPositionConfigurationType, RadialPositionConfigurationType, SinglePositionConfigurationType, TriplePositionConfigurationType. | |

5.1 Package Terminal Position (cont'd)

The definition of the above *PackageTerminalPosition* data elements can be found in JESD30, under the “Terminal position Prefix” section which *includes* the “Prefixes for terminal position” table, plus three sections in Annex A, namely

- Terminal position with additional definition,
- Relationship concepts between the Terminal Contact Area and the Body outline, and
- Position Images.

NOTE As a result of the *TerminalGroup/TerminalPosition* (as defined in section 4.16 Terminal Position below), the rollup of the *TerminalPosition* values for all *TerminalGroup-Array/TerminalGroups* into the *PackageTerminalPosition*, may be different than the value at the *PackageTerminalPosition* level. This is described in detail in JESD30, “Terminal-position prefix”.

5.1.1 Bottom

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/PackageTerminalPosition/Bottom. |
| diagram | |
| type | BottomUpperPositionConfigurationType, JEP30-D10:EmptyType, BottomDiagonalType, MatrixType, MatrixAlterationType. |

5.1.2 Dual

| | | |
|---------|--|--|
| path | PartModel/PackageSection/Package-Array/Package/PackageTerminalPosition/Dual. | |
| diagram | <p>The diagram illustrates the structure of the Dual package terminal position. It shows a hierarchy of entities and their relationships. The 'Dual' entity (type DualPositionConfigurationType) is associated with a 'Corners' entity (type DualCornerConifurationType). The 'Corners' entity is further associated with a 'DualCornerConifurationType' entity. This entity contains several relationship types: Left-to-Right (DualSidePositionRelationshipType), Back-to-Front (DualSidePositionRelationshipType), Bottom-to-Upper (Bottom-to-UpperDualPositionRelationshipType), Left (SideCornerPositionRelationshipType), Back (SideCornerPositionRelationshipType), Right (SideCornerPositionRelationshipType), Front (SideCornerPositionRelationshipType), Left-to-RightTop (DifferentSidePositionRelationshipType), Left-to-RightBottom (DifferentSidePositionRelationshipType), Back-to-FrontLeft (DifferentSidePositionRelationshipType), and Back-to-FrontRight (DifferentSidePositionRelationshipType).</p> | |
| type | DualPositionConfigurationType, DualSidePositionRelationshipType, Bottom-to-UpperDualPositionRelationshipType, DualCornerConifurationType, SideCornerPositionRelationshipType, DifferentSidePositionRelationshipType. | |

5.1.2.1 Dual Left-to-Right and Back-to-Front

| | |
|---------|--|
| path | <ol style="list-style-type: none"> 1. PartModel/PackageSection/Package-Array/Package/PackageTerminalPosition/Dual/Left-to-Right 2. PartModel/PackageSection/Package-Array/Package/PackageTerminalPosition/Dual/Back-to-Front |
| diagram | |
| type | DualSidePositionRelationshipType , JEP30-D10:EmptyType , SpanPositionRelationshipType . |

5.1.2.2 Dual Bottom-to-Upper

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/PackageTerminalPosition/Dual/Bottom-to-Upper |
| diagram | |
| type | Bottom-to-UpperDualPositionRelationshipType , JEP30-D10:EmptyType , Bottom-to-UpperSpanPositionRelationshipType |

5.1.2.3 Dual Corners

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/PackageTerminalPosition/Dual/Corners |
| diagram | <pre> classDiagram class Corners { type DualCornerConfigurationType } class DualCornerConfigurationType { Left SideCornerPositionRelationshipType Back SideCornerPositionRelationshipType Right SideCornerPositionRelationshipType Front SideCornerPositionRelationshipType Left-to-RightTop DifferentSidePositionRelationshipType Left-to-RightBottom DifferentSidePositionRelationshipType Back-to-FrontLeft DifferentSidePositionRelationshipType Back-to-FrontRight DifferentSidePositionRelationshipType } class SideCornerPositionRelationshipType { Outside JEP30-D10:EmptyType Edge-Out JEP30-D10:EmptyType Overlap JEP30-D10:EmptyType Edge-In JEP30-D10:EmptyType } class DifferentSidePositionRelationshipType { Outside JEP30-D10:EmptyType Edge-Out JEP30-D10:EmptyType Overlap JEP30-D10:EmptyType } Corners --> DualCornerConfigurationType DualCornerConfigurationType --> SideCornerPositionRelationshipType : Left DualCornerConfigurationType --> SideCornerPositionRelationshipType : Back DualCornerConfigurationType --> SideCornerPositionRelationshipType : Right DualCornerConfigurationType --> SideCornerPositionRelationshipType : Front DualCornerConfigurationType --> DifferentSidePositionRelationshipType : Left-to-RightTop DualCornerConfigurationType --> DifferentSidePositionRelationshipType : Left-to-RightBottom DualCornerConfigurationType --> DifferentSidePositionRelationshipType : Back-to-FrontLeft DualCornerConfigurationType --> DifferentSidePositionRelationshipType : Back-to-FrontRight </pre> |
| type | DualCornerConfigurationType, SideCornerPositionRelationshipType, DifferentSidePositionRelationshipType, JEP30-D10:EmptyType. |

5.1.3 Diagonal

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/PackageTerminalPosition/Diagonal |
| diagram | <p>The diagram illustrates the structure of the Diagonal configuration. It starts with a Diagonal class (type DiagonalCornerConfigurationType) which is associated with a DiagonalCornerConfigurationType container. This container holds several relationship types: Southwest-to-Northeast (type InternalCornerPositionRelationshipType), Northwest-to-Southeast (type InternalCornerPositionRelationshipType), Back-left-to-Front-right (type DifferentSidePositionRelationshipType), Front-left-to-Back-right (type DifferentSidePositionRelationshipType), Left-bottom-to-Right-top (type DifferentSidePositionRelationshipType), and Left-top-to-Right-bottom (type DifferentSidePositionRelationshipType). The Southwest-to-Northeast relationship is further detailed within a dashed box labeled InternalCornerPositionRelationshipType, showing sub-relationships: Overlap (type JEP30-D10:EmptyType), Edge-In (type JEP30-D10:EmptyType), and Inside (type JEP30-D10:EmptyType). The Back-left-to-Front-right relationship is detailed within a dashed box labeled DifferentSidePositionRelationshipType, showing sub-relationships: Outside (type JEP30-D10:EmptyType), Edge-Out (type JEP30-D10:EmptyType), and Overlap (type JEP30-D10:EmptyType).</p> |
| type | DiagonalCornerConfigurationType, InternalCornerPositionRelationshipType, JEP30-D10:EmptyType, DifferentSidePositionRelationshipType. |

5.1.4 Quad

| | | |
|---------|---|--|
| path | PartModel/PackageSection/Package-Array/Package/PackageTerminalPosition/Quad. | |
| diagram | <p>The diagram illustrates the structure of the Quad configuration types. It features a central 'Quad' class (type: QuadPositionConfigurationType) which is associated with a 'Span' class (type: SpanPositionRelationshipType) and a 'Corners' class (type: QuadCornerConfigurationType). The 'Span' class is further associated with a 'SpanPositionRelationshipType' class. The 'Corners' class is associated with a 'QuadCornerConfigurationType' class. The 'SpanPositionRelationshipType' class contains three sub-classes: 'Overlap' (type: JEP30-D10:EmptyType), 'Edge-In' (type: JEP30-D10:EmptyType), and 'Inside' (type: JEP30-D10:EmptyType). The 'QuadCornerConfigurationType' class contains four sub-classes: 'Overlap' (type: JEP30-D10:EmptyType), 'Edge-In' (type: JEP30-D10:EmptyType), 'Inside' (type: JEP30-D10:EmptyType), and 'Left-to-Right' (type: DifferentSidePositionRelationshipType). The 'Left-to-Right' and 'Back-to-Front' classes are also associated with the 'DifferentSidePositionRelationshipType' class.</p> | |
| type | QuadPositionConfigurationType, JEP30-D10:EmptyType, SpanPositionRelationshipType, QuadCornerConfigurationType, DifferentSidePositionRelationshipType. | |

5.1.5 Radial

| | | | |
|---------|--|--|--|
| path | PartModel/PackageSection/Package-Array/Package/PackageTerminalPosition/Radial | | |
| diagram | <p>The diagram illustrates the structure of the RadialPositionConfigurationType and its relationship to the RadialCornerPositionRelationshipType. The RadialPositionConfigurationType is a container type that includes several subtypes: Outside, Edge-Out, Overlap, Edge-In, and Inside. Each of these subtypes is of type JEP30-D10:EmptyType. Additionally, the RadialPositionConfigurationType includes a Corners subtype of type RadialCornerPositionRelationshipType. The RadialCornerPositionRelationshipType is a container type that includes subtypes Outside, Edge-Out, Overlap, Edge-In, and Inside, all of which are of type JEP30-D10:EmptyType. The Radial type is a specialization of RadialPositionConfigurationType.</p> | | |
| type | RadialPositionConfigurationType, JEP30-D10:EmptyType, RadialCornerpositionRelationshipType. | | |

5.1.6 Single

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/PackageTerminalPosition/Single. |
| diagram | <p>The diagram illustrates the structure of the Single package terminal position. It shows a Single entity (type: SinglePositionConfigurationType) connected to a SinglePositionConfigurationType container. This container includes four SingleSidePositionRelationshipType entities (Left, Back, Right, Front) and one SingleCornerConfigurationType entity. The SingleCornerConfigurationType entity further includes twelve DifferentSidePositionRelationshipType entities (Southwest, Southeast, Northeast, Northwest, Back-left, Back-right, Front-right, Front-left, Left-bottom, Right-bottom, Right-top, Left-top).</p> |
| type | SinglePositionConfigurationType, SingleSidePositionRelationshipType, SingleCornerConfigurationType, InternalCornerPositionRelationshipType, DifferentSidePositionRelationshipType. |

5.1.6.1 Single Side

| | |
|---------|--|
| path | <ol style="list-style-type: none"> 1. PartModel/PackageSection/Package-Array/Package/PackageTerminalPosition/Single/Left 2. PartModel/PackageSection/Package-Array/Package/PackageTerminalPosition/Single/Back 3. PartModel/PackageSection/Package-Array/Package/PackageTerminalPosition/Single/Right 4. PartModel/PackageSection/Package-Array/Package/PackageTerminalPosition/Single/Front |
| diagram | |
| type | SingleSidePositionRelationshipType , JEP30-D10:EmptyType. |

5.1.6.2 Corner - Internal

| | |
|---------|--|
| path | <ol style="list-style-type: none"> 1. PartModel/PackageSection/Package-Array/Package/PackageTerminalPosition/Single/Corner/Southwest 2. PartModel/PackageSection/Package-Array/Package/PackageTerminalPosition/Single/Corner/Southeast 3. PartModel/PackageSection/Package-Array/Package/PackageTerminalPosition/Single/Corner/Northeast 4. PartModel/PackageSection/Package-Array/Package/PackageTerminalPosition/Single/Corner/Northwest |
| diagram | |
| type | InternalCornerPositionRelationshipType , JEP30-D10:EmptyType. |

5.1.6.3 Corner - External

| | |
|---------|---|
| path | <div>1. PartModel/PackageSection/Package-Array/Package/PackageTerminalPosition/Single/Corner/Back-left</div> <div>2. PartModel/PackageSection/Package-Array/Package/PackageTerminalPosition/Single/Corner/Back-right</div> <div>3. PartModel/PackageSection/Package-Array/Package/PackageTerminalPosition/Single/Corner/Front-right</div> <div>4. PartModel/PackageSection/Package-Array/Package/PackageTerminalPosition/Single/Corner/Front-left</div> <div>5. PartModel/PackageSection/Package-Array/Package/PackageTerminalPosition/Single/Corner/Left-bottom</div> <div>6. PartModel/PackageSection/Package-Array/Package/PackageTerminalPosition/Single/Corner/Right-bottom</div> <div>7. PartModel/PackageSection/Package-Array/Package/PackageTerminalPosition/Single/Corner/Right-top</div> <div>8. PartModel/PackageSection/Package-Array/Package/PackageTerminalPosition/Single/Corner/Left-top</div> |
| diagram | <pre>classDiagram class BackLeft["Back-left"] { type DifferentSidePositionRelationshipType } class DifferentSidePositionRelationshipType { <<abstract>> +Outside +Edge-Out +Overlap } class Outside { type JEP30-D10:EmptyType } class EdgeOut["Edge-Out"] { type JEP30-D10:EmptyType } class Overlap { type JEP30-D10:EmptyType } BackLeft --> DifferentSidePositionRelationshipType DifferentSidePositionRelationshipType < -- Outside DifferentSidePositionRelationshipType < -- EdgeOut DifferentSidePositionRelationshipType < -- Overlap</pre> |
| type | DifferentSidePositionRelationshipType, JEP30-D10:EmptyType. |

5.1.7 Triple

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/PackageTerminalPosition/Triple |
| diagram | <p>The diagram illustrates the structure of the Triple class. The Triple class is of type TriplePositionConfigurationType. It has a composition relationship with the TripleCornerConfigurationType class. The TriplePositionConfigurationType class contains four attributes: NotLeft, NotBack, NotRight, and NotFront, all of type MultiSidePositionRelationshipType. The TripleCornerConfigurationType class contains twelve attributes: Not-Southwest, Not-Southeast, Not-Northeast, Not-Northwest, Not-Back-left, Not-Back-right, Not-Front-right, Not-Front-left, Not-Left-bottom, Not-Right-bottom, Not-Right-top, and Not-Left-top, all of type DifferentSidePositionRelationshipType.</p> |
| type | TriplePositionConfigurationType, MultiSidePositionRelationshipType, TripleCornerConfigurationType, InternalCornerPositionRelationshipType, DifferentSidePositionRelationshipType. |

5.1.7.1 Triple – Multi Side position

| | |
|---------|--|
| path | <ol style="list-style-type: none"> 1. PartModel/PackageSection/Package-Array/Package/PackageTerminalPosition/Triple/NotLeft 2. PartModel/PackageSection/Package-Array/Package/PackageTerminalPosition/Triple/NotBack 3. PartModel/PackageSection/Package-Array/Package/PackageTerminalPosition/Triple/NotRight 4. PartModel/PackageSection/Package-Array/Package/PackageTerminalPosition/Triple/NotFront |
| diagram | |
| type | MultiSidePositionRelationshipType , JEP30-D10:EmptyType , SpanPositionRelationshipType . |

5.1.8 Upper

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/PackageTerminalPosition/Upper. |
| diagram | |
| type | BottomUpperPositionConfigurationType , JEP30-D10:EmptyType , BottomUpperDiagonalType , MatrixType , MatrixAlterationType . |

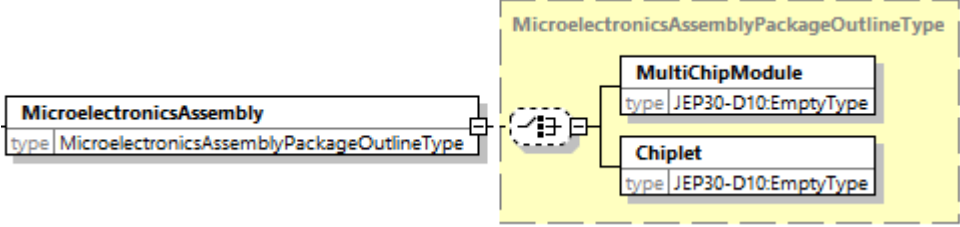
5.2 Package Outline Style

| | | |
|---------|--|--|
| path | PartModel/PackageSection/Package-Array/Package/PackageOutlineStyle. | |
| diagram | | |
| type | PackageOutlineType, JEP30-D10:EmptyType, MicroelectronicsAssemblyPackageOutlineType, PostMountPackageOutlineType | |

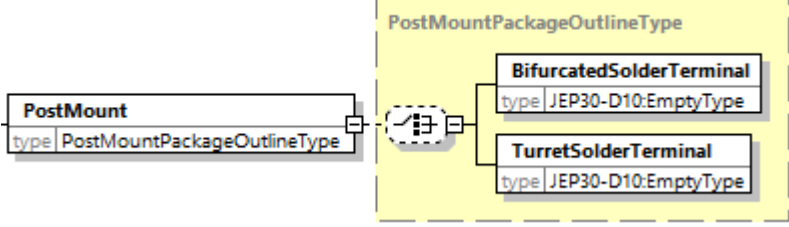
5.2 Package Outline Style (cont'd)

The definition of the above *PackageOutlineStyle* data elements can be found in the JESD30 publication, under the “Package-outline style codes” section which includes a table with the definitions of each of the above data elements. In addition, in JESD30, there is an “Illustrations of Package Outline Styles” in Annex A.

5.2.1 Microelectronics Assembly

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/PackageOutlineStyle/MicroelectronicsAssembly |
| diagram |  |
| type | MicroelectronicsAssemblyPackageOutlineType |

5.2.2 Post Mount

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/PackageOutlineStyle/PostMount. |
| diagram |  |
| type | PostMountPackageOutlineType |

5.3 Package Terminal Code

| | | |
|---------|---|--|
| path | PartModel/PackageSection/Package-Array/Package/PackageTerminalCode | |
| diagram | <p>The diagram illustrates the structure of the PackageTerminalCodeType data element. It is a container type that includes a list of terminal code types. The types are organized into two main groups, each enclosed in a dashed box. The left group, titled PackageTerminalCodeType, contains the following types: Ball (type: BallType), C-bend (type: JEP30-D10:EmptyType), Lug (type: LugType), Mixed-SMT (type: JEP30-D10:EmptyType), Flat (type: FlatType), Gull-wing (type: Gull-wingType), CompressedMountTechnology (type: JEP30-D10:EmptyType), PostTerminal (type: PostTerminalType), J-bend (type: JEP30-D10:EmptyType), Mixed-TH (type: JEP30-D10:EmptyType), L-bend (type: L-bendType), and Column (type: ColumnType). The right group, titled PackageTerminalCodeType, contains the following types: Column (type: ColumnType), Surface-terminal (type: Surface-terminalType), Pressfit (type: PressfitType), Pin (type: PinType), Quick-connect (type: JEP30-D10:EmptyType), Mixed-Technology (type: JEP30-D10:EmptyType), Wraparound (type: WraparoundType), S-bend (type: S-bendType), Through-Hole (type: Through-HoleType), J-inverted (type: JEP30-D10:EmptyType), TerminalWire (type: WireType), and Screw (type: ScrewType). A dashed box labeled PackageTerminalCode (type: PackageTerminalCodeType) is shown at the bottom left, connected to the Column type in the left group.</p> | |
| type | PackageTerminalCodeType, BallType, LugType, FlatType, Gull-wingType, PostTerminalType, L-BendType, ColumnType, SurfaceTerminalType, PressfitType, PinType, WraparoundType, S-BendType, Through-HoleType, WireType, ScrewType. | |

The definition of the above *PackageTerminalCode* data elements and their following sub-structures can be found in JESD30, under the “Terminal Shape Suffix” section which *includes* the “Suffixes for terminal shape” table, and two sections in Annex A, namely “Suffixes for terminal shape with additional definition”, and “Illustrations of terminal shape”.

NOTE As a result of the *TerminalGroup/Terminal* (as defined in 4.18), the rollup of the *Terminal* values for all *TerminalGroup-Array/TerminalGroups* into the *PackageTerminalCode*, may be different than the value at the *PackageTerminalCode* level. This is described in detail in JESD30, “Terminal Shape Suffix”.

5.3.1 Ball

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/PackageTerminalCode/Ball |
| diagram | <pre> classDiagram class Ball { type BallType } class BallType { Bump { type JEP30-D10:EmptyType } Collapsing { type JEP30-D10:EmptyType } Non-Collapsing { type JEP30-D10:EmptyType } } Ball --> BallType </pre> |
| type | BallType , JEP30-D10:EmptyType |

5.3.2 Lug

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/PackageTerminalCode/Lug |
| diagram | <pre> classDiagram class Lug { type LugType } class LugType { PerforatedSolderTerminal { type JEP30-D10:EmptyType } CupSolderTerminal { type JEP30-D10:EmptyType } HookSolderTerminal { type JEP30-D10:EmptyType } Crimplug { type JEP30-D10:EmptyType } RingTongueTerminal { type JEP30-D10:EmptyType } LugWithThreadedHole { type JEP30-D10:EmptyType } } Lug --> LugType </pre> |
| type | LugType , JEP30-D10:EmptyType |

5.3.3 Flat

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/PackageTerminalCode/Flat |
| diagram | |
| type | FlatType, JEP30-D10:EmptyType |

5.3.4 Gull-wing

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/PackageTerminalCode/Gull-wing |
| diagram | |
| type | Gull-wingType, JEP30-D10:EmptyType |

5.3.5 Post Terminal

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/PackageTerminalCode/PostTerminal |
| diagram | |
| type | PostTerminalType, JEP30-D10:EmptyType |

5.3.6 L-bend

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/PackageTerminalCode/L-Bend |
| diagram | |
| type | L-BendType, JEP30-D10:EmptyType |

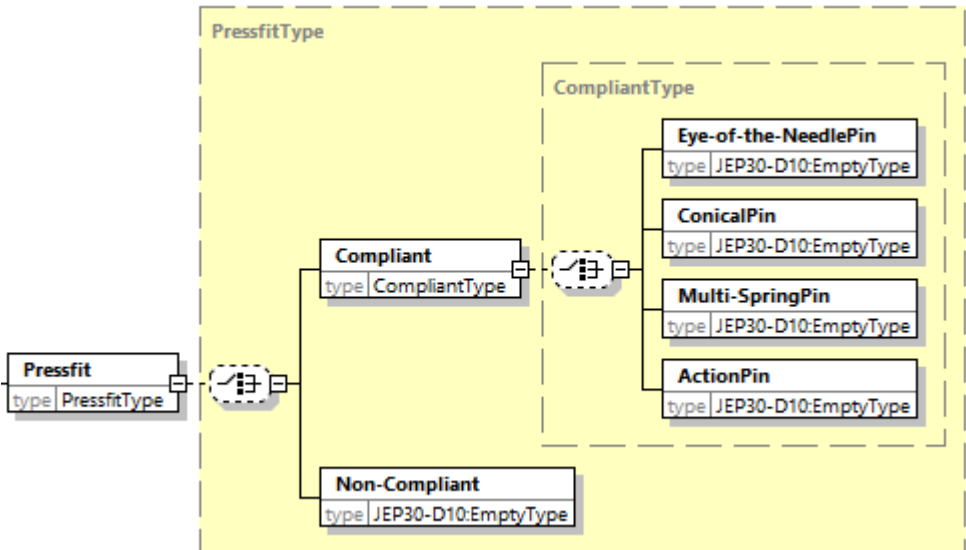
5.3.7 Column

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/PackageTerminalCode/ Column |
| diagram | |
| type | ColumnType, JEP30-D10:EmptyType |

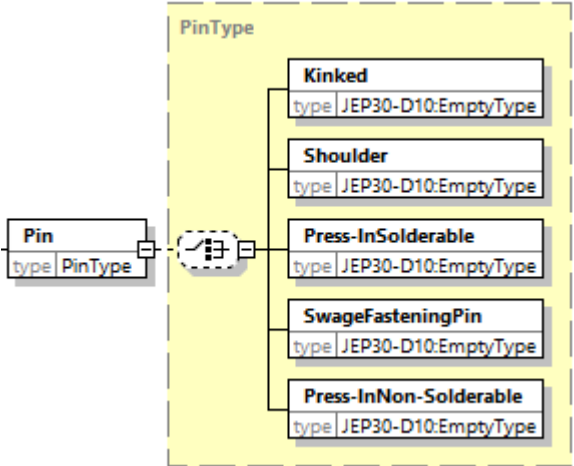
5.3.8 Surface Terminal

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/PackageTerminalCode/SurfaceTerminal |
| diagram | <p>The diagram illustrates the UML class hierarchy for SurfaceTerminalType. The hierarchy is as follows:</p> <ul style="list-style-type: none">SurfaceTerminalType (Base Class)<ul style="list-style-type: none">Castellated (type: CastellatedType)<ul style="list-style-type: none">CastellatedType (Intermediate Class)<ul style="list-style-type: none">WettableFlank (type: JEP30-D10:EmptyType)D-Shape (type: JEP30-D10:EmptyType)Pullback (type: JEP30-D10:EmptyType)Hole (type: JEP30-D10:EmptyType)With-opening (type: JEP30-D10:EmptyType)Open-Ring (type: RingType)<ul style="list-style-type: none">RingType (Intermediate Class)<ul style="list-style-type: none">Castellated (type: RingCastellatedType)<ul style="list-style-type: none">RingCastellatedType (Intermediate Class)<ul style="list-style-type: none">WettableFlank (type: JEP30-D10:EmptyType) |
| type | SurfaceTerminalType, JEP30-D10:EmptyType, CastellatedType, RingType, RingCastellatedType. |

5.3.9 Pressfit

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/PackageTerminalCode/Pressfit |
| diagram |  <p>The diagram illustrates the hierarchy of the PressfitType. A Pressfit box (type PressfitType) is connected to a dashed box labeled PressfitType. Inside this box, there are two main branches: Compliant (type CompliantType) and Non-Compliant (type JEP30-D10:EmptyType). The Compliant box is further connected to a dashed box labeled CompliantType. Inside CompliantType, there are four sub-types: Eye-of-the-NeedlePin (type JEP30-D10:EmptyType), ConicalPin (type JEP30-D10:EmptyType), Multi-SpringPin (type JEP30-D10:EmptyType), and ActionPin (type JEP30-D10:EmptyType).</p> |
| type | PressfitType, CompliantType, JEP30-D10:EmptyType |

5.3.10 Pin

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/PackageTerminalCode/Pin |
| diagram |  <p>The diagram illustrates the hierarchy of the PinType. A Pin box (type PinType) is connected to a dashed box labeled PinType. Inside this box, there are five sub-types: Kinked (type JEP30-D10:EmptyType), Shoulder (type JEP30-D10:EmptyType), Press-InSolderable (type JEP30-D10:EmptyType), SwageFasteningPin (type JEP30-D10:EmptyType), and Press-InNon-Solderable (type JEP30-D10:EmptyType).</p> |
| type | PinType, JEP30-D10:EmptyType |

5.3.11 Wraparound

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/PackageTerminalCode/Wraparound |
| diagram | <p>The diagram illustrates the structure of the WraparoundType and its associated types. The WraparoundType class is shown with a dashed border and a yellow background. It contains several nested types and associations:</p> <ul style="list-style-type: none">D-Shape: A type defined as <code>JEP30-D10:EmptyType</code>.Castellated: A type defined as <code>WraparoundCastellatedType</code>.WraparoundCastellatedType: A type defined as <code>WraparoundCastellatedType</code>.RingType: A type defined as <code>RingType</code>.Open-Ring: A type defined as <code>RingType</code>.Nibble: A type defined as <code>JEP30-D10:EmptyType</code>.ConvexE: A type defined as <code>JEP30-D10:EmptyType</code>.ConvexS: A type defined as <code>JEP30-D10:EmptyType</code>. <p>The WraparoundType class is associated with the WraparoundCastellatedType class. The WraparoundCastellatedType class is associated with the Castellated class. The Castellated class is associated with the RingType class. The RingType class is associated with the Open-Ring class. The Open-Ring class is associated with the Castellated class. The Castellated class is associated with the RingCastellatedType class. The RingCastellatedType class is associated with the WettableFlank class. The WettableFlank class is associated with the D-Shape class.</p> |
| type | WraparoundType, JEP30-D10:EmptyType, WraparoundCastellatedType, RingType, RingCastellatedType. |

5.3.12 S-Bend

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/PackageTerminalCode/S-Bend |
| diagram | |
| type | S-BendType , JEP30-D10:EmptyType |

5.3.13 Through-Hole

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/PackageTerminalCode/Through-Hole |
| diagram | |
| type | Through-HoleType , JEP30-D10:EmptyType |

5.3.14 Terminal Wire

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/PackageTerminalCode/TerminalWire |
| diagram | |
| type | TerminalWireType , JEP30-D10:EmptyType |

5.3.15 Screw

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/PackageTerminalCode/Screw |
| diagram | |
| type | ScrewType , JEP30-D10:EmptyType |

5.4 Package Mounting Preparation

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/PackageMountingPreparation |
| diagram | |
| type | PackageMountingPreparationType , JEP30-D10:EmptyType . |

5.4 Package Mounting Preparation (cont'd)

PackageMountingPreparation definition can be found in JESD30, under the “Mounting Preparation” section which includes some graphical images to describe the values.

5.5 Package Body Direction

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/PackageBodyDirection |
| diagram | |
| type | PackageBodyDirectionType, JEP30-D10:EmptyType. |

PackageBodyDirection definition can be found in the JESD30, under the “Body Direction” section which includes some graphical images to describe the values.

5.6 Mass

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/Mass |
| diagram | |
| type | JEP30-D10:MassValueSetType, |

The Mass *Units* of Measure are as follows; however, the standard preference is metric in grams.

- | | |
|-------------------|------------------|
| 1) Microgram (ug) | 4) Kilogram (kg) |
| 2) Miligram (mg) | 5) Ounce (oz) |
| 3) Gram (g) | 6) Pound (lb) |

5.7 Package Shape

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/PackageShape. |
| diagram | <p>The diagram illustrates the structure of the PackageShapeType. It is a collection of various package shapes, each represented by a class and its associated reference type. The shapes include:</p> <ul style="list-style-type: none"> Rectangle (type: ReferenceRectangleType) RectangleConcave (type: ReferenceRectangleConcaveType) RoundedRectangle (type: ReferenceRoundedRectangleType) ReferenceRoundedRectangleWithTab (type: ReferenceRoundedRectangleWithTabType) ModifiedRectangle (type: PackageReferenceModifiedRectangleType) ModifiedRectangleWithTab (type: ModifiedReferenceRectangleWithTabType) Circle (type: ReferenceCircleType) CircleWithTab (type: ReferenceCircleWithTabType) D-Shape (type: ReferenceD-ShapeType) Double-D (type: ReferenceDouble-DType) RegularPolygon (type: ReferenceRegularPolygonType) Segment (type: ReferenceSegmentType) Para-truncatedCircle (type: ReferencePara-truncatedCircleType) RoundedDiamond (type: ReferenceRoundedDiamondType) RoundedChamferedDiamond (type: ReferenceRoundedChamferedDiamondType) IsoscelesTrapezoid (type: ReferenceIsoscelesTrapezoidType) Contour (type: ContourShapeType) ShapeCenter (type: JEP30-D10:PointXYType) PackageVerticalDimension (type: PackageVerticalDimensionType) <p>A dashed box labeled PackageShape (type: PackageShapeType) is connected to the D-Shape class, indicating a specific reference or association.</p> |
| type | PackageShapeType, ReferenceRectangleType, ReferenceRectangleConcaveType, ReferenceRoundedRectangleType, ReferenceRoundedRectangleWithTabType, PackageReferenceModifiedRectangleType, ModifiedReferenceRectangleWithTabType, ReferenceCircleType, ReferenceCircleWithTabType, ReferenceD-ShapeType, ReferenceDouble-DType, ReferenceRegularPolygonType, ReferenceSegmentType, ReferencePara-truncatedCircleType, ReferenceRoundedDiamondType, ReferenceRoundedChamferedDiamondType, ReferenceIsoscelesTrapezoidType, ContourShapeType, JEP30-D10:PointXYType, PackageVerticalDimensionType. |

4.7 Package Shape (cont'd)

Table 2 shows an array of dimensions that are captured below each shape listed under [PackageShape](#).

Table 2 - Package Shape versus Dimensions

| Package Shape | Dimension 1 | Dimension 2 | Dimension 3 | Diameter | Radius | Angle | No. of Sides | Inner/Outer | Tab Length | Tab Width | Tab Orientation | Impacted Corner |
|-----------------------------|-------------|-------------|-------------|----------|--------|-------|--------------|-------------|------------|-----------|-----------------|-----------------|
| Rectangle | Y | Y | | | | Y | | | | | | |
| Rectangle Concave | Y | Y | | | Y | Y | | | | | | |
| Rounded Rectangle | Y | Y | | | Y | Y | | | | | | |
| Rounded Rectangle with Tab | Y | Y | | | Y | Y | | | Y | Y | Y | |
| Modified Rectangle | Y | Y | | | | Y | | | | | | Y |
| Modified Rectangle with Tab | Y | Y | | | | Y | | | Y | Y | Y | Y |
| Circle | | | | Y | | | | | | | | |
| Circle with Tab | | | | Y | | | | | Y | Y | Y | |
| D-Shape | Y | Y | | | | Y | | | | | | |
| Double-D | Y | Y | | | | Y | | | | | | |
| Regular Polygon | | | | Y | | Y | Y | Y | | | | |
| Segment | Y | | | Y | | Y | | | | | | |
| Para-truncated Circle | Y | | | Y | | Y | | | | | | |
| Rounded Diamond | Y | | | Y | Y | Y | | | | | | |
| Rounded Chamfered Diamond | Y | Y | | Y | | Y | | | | | | |
| Isosceles Trapezoid | Y | Y | Y | | | Y | | | | | | |

The Shape Center and the Package Vertical Dimensions are applicable to all shapes. The definition of each shape including Contour is outlined in Annex A (informative) Shape Dimensions.

5.7.1 Contour

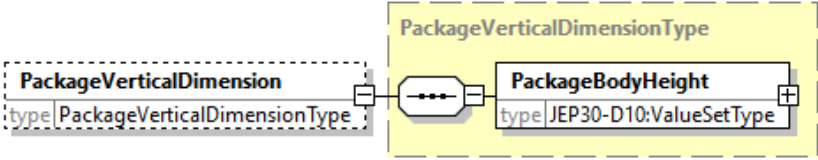
| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/PackageShape/Contour |
| diagram | |
| type | ContourShapeType, ContourToleranceType, JEP30-D10:SplitType, JEP30-D10:ToleranceUOMType, JEP30-D10:ToleranceGroup, Outline-ArrayType, OutlineSegmentType, JEP30-D10:FeatureControl-IDsType. |

The addition of the *Outline-Array* enables the GD&T to be assigned to any segment of the contour outline.

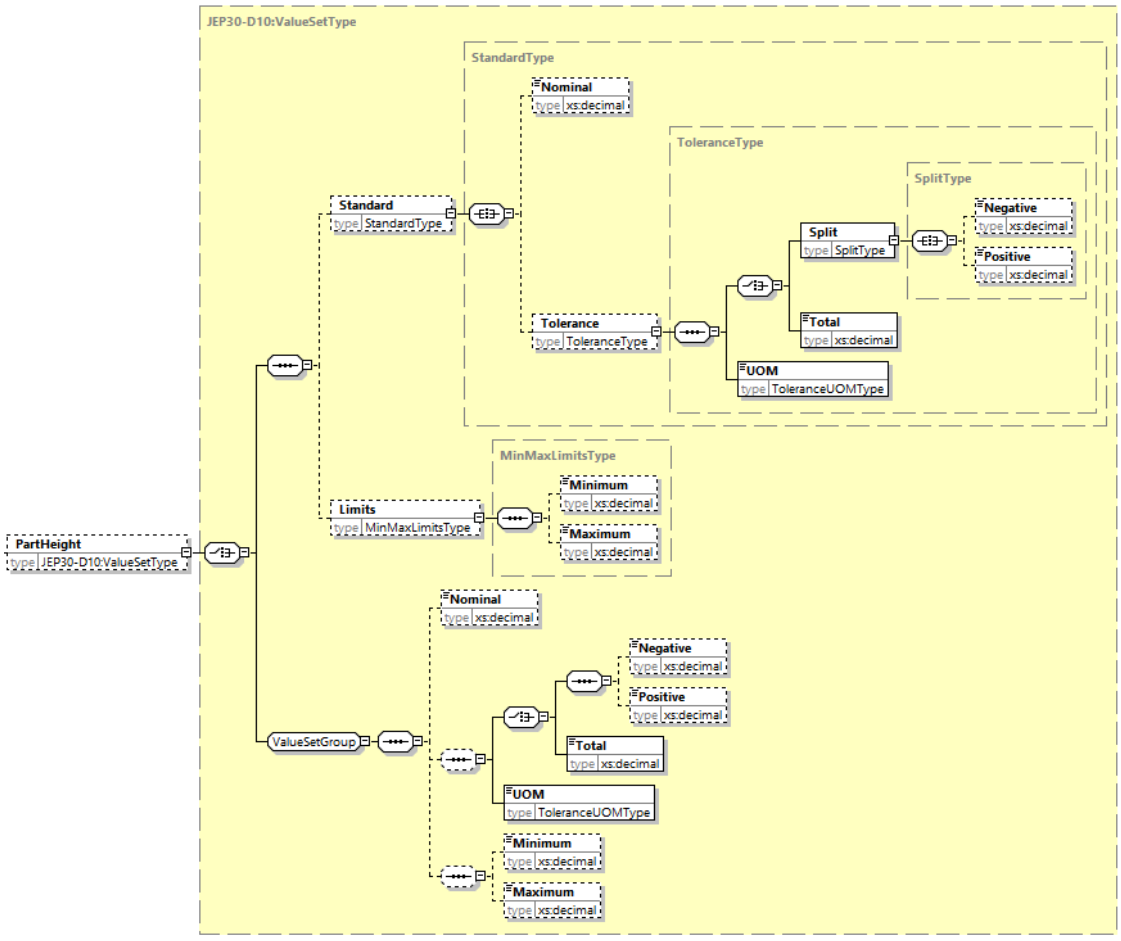
5.7.2 Shape Center

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/PackageShape/ShapeCenter |
| diagram | |
| type | JEP30-D10:PointXYType |

5.7.3 Package Vertical Dimension

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/PackageShape/PackageVerticalDimension |
| diagram |  <p>The diagram shows a dashed box labeled PackageVerticalDimension with a note <code>type PackageVerticalDimensionType</code>. This box is connected via a sequence connector (two horizontal lines with a central dot) to a solid box labeled PackageBodyHeight with a note <code>type JEP30-D10:ValueSetType</code>. The entire structure is enclosed in a yellow dashed box labeled PackageVerticalDimensionType.</p> |
| type | PackageVerticalDimensionType, JEP30-D10:ValueSetType, |

5.8 Part Height

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/PartHeight |
| diagram |  <p>The diagram illustrates the PartHeight structure, which is a dashed box with a note <code>type JEP30-D10:ValueSetType</code>. It is connected via a choice connector (a circle with a vertical line and two horizontal lines) to two main branches. The first branch is a solid box labeled Standard with a note <code>type StandardType</code>, which is further connected via an <code>or</code> connector (a circle with a horizontal line and two vertical lines) to a dashed box labeled StandardType. This StandardType box contains a Nominal box (note <code>type xs:decimal</code>) and a Tolerance box (note <code>type ToleranceType</code>). The ToleranceType box is connected via a choice connector to a SplitType box (note <code>type SplitType</code>) and a Total box (note <code>type xs:decimal</code>). The SplitType box is connected via a choice connector to Negative (note <code>type xs:decimal</code>) and Positive (note <code>type xs:decimal</code>). The second branch from the main choice connector is a solid box labeled Limits with a note <code>type MinMaxLimitsType, which is connected via a choice connector to a MinMaxLimitsType box. This box contains Minimum (note <code>type xs:decimal</code>) and Maximum (note <code>type xs:decimal</code>). The second main branch is a solid box labeled ValueSetGroup with a note <code>type ValueSetGroup</code>, which is connected via a choice connector to a Nominal box (note <code>type xs:decimal</code>) and a MinMaxLimitsType box. This MinMaxLimitsType box contains Negative (note <code>type xs:decimal</code>), Positive (note <code>type xs:decimal</code>), Total (note <code>type xs:decimal</code>), UOM (note <code>type ToleranceUOMType</code>), Minimum (note <code>type xs:decimal</code>), and Maximum (note <code>type xs:decimal</code>). The entire structure is enclosed in a yellow dashed box labeled JEP30-D10:ValueSetType.</code></p> |
| type | JEP30-D10:ValueSetType, StandardType, ToleranceType, SplitType, ToleranceUOMType, MinMaxLimitsType, ValueSetGroup. |

5.8 Part Height (cont'd)

PartHeight represents the total height of the part from the seating plane – therefore the sum of the *PackageBodyHeight* plus the *Standoff* (as defined in section 4.13 below) is normally equal to the part height. The exception is when you also have terminals exiting from the “Topside” (as defined by *PartAccessDirection*) of the part, whereby *PartHeight* is then defined as the distance from the seating plane to the top of the terminal that exits from the topside of the part.

The Value Set Type is widely used throughout the Schema. It provides for all the following combinations in a structured way:

- 1) Nominal,
- 2) Positive and Negative Tolerances, or Total tolerance based on the same UOM or based on a percentage of the nominal value, and
- 3) Minimum and or Maximum.

For many applications, both the minimum and maximum Part Height details are required.

5.9 Clearance Region - Array

| | | |
|---------------------------|---|--|
| path | PartModel/PackageSection/Package-Array/Package/ClearanceRegion-Array. | |
| diagram part 1 of 2 | | |
| diagram part 2 of 2 | | |
| type | ClearanceRegionArrayType, JEP30-D10:ValueSetType, ClearanceRegionType, RectangleType, CircleType, ContourType, JEP30-D10:PointType, JEP30-D10:UnspecifiedValueSetType, StandardType, MinMaxLimitsType. | |

ClearanceRegion-Array represents the space available underneath the part after mounting to the printed board. This space if large enough can allow other parts to be placed in the product design underneath this part. The space also determines the rules applied to the solder fillet for any terminal whose land pattern may encroach inside the package body outline. If there is insufficient space around the terminals in this area, changes to the land pattern may be necessary.

If there is a simple clearance as for example typically found under an SOIC, then the clearance value can be captured under the element **Standoff**. However, many parts have different **ClearanceRegions** under the part. These **ClearanceRegions** can have different shapes, the most common being **Rectangle** or **Circle** shapes, but additional shapes can be captured via the **Contour** element.

NOTE The element **Unspecified** under the **ClearanceRegion-Array**. The purpose of this flag is to enable the user to specify that a Clearance does exist even though its value is not specified, and that the clearance is adequate to provide for a Heel, Toe, Side or Periphery fillet of solder around the Terminal Contact Area without the solder touching the package body, where this could lead to manufacturing defects such as solder balls.

5.9 Clearance Region - Array (cont'd)

The *Unspecified* element under *ClearanceRegion-Array/ClearanceRegion/Clearance* is used when there are some *ClearanceRegions* shapes that do not have a *Clearance* specified, but the *ClearanceRegion* shape is defined.

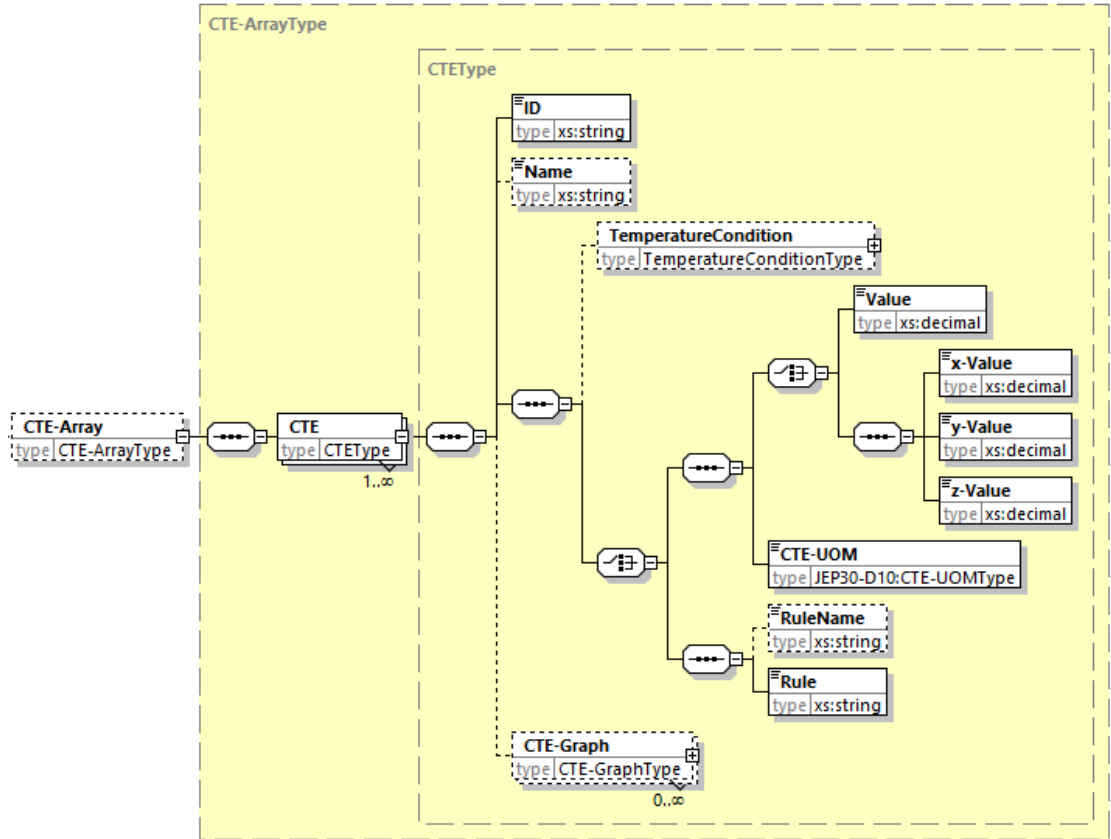
This document recommends that the clearance value is properly specified, and that the *Unspecified* field is not used.

The definition of each shape is outlined in Annex A (informative) Shape Dimensions.

5.10 Assembly Technology

| | | |
|---------|---|--|
| path | PartModel/PackageSection/Package-Array/Package/AssemblyTechnology. | |
| diagram | <pre> classDiagram class AssemblyTechnologyType { type JEP30-D10:EmptyType } class AssemblyTechnology { type AssemblyTechnologyType } class SMTMaskReflow { type JEP30-D10:EmptyType } class PasteInHole { type JEP30-D10:EmptyType } class WaveSoldering { type JEP30-D10:EmptyType } class SelectiveMiniWaveSoldering { type JEP30-D10:EmptyType } class LaserSoldering { type JEP30-D10:EmptyType } class ManualSoldering { type JEP30-D10:EmptyType } class Pressfit { type JEP30-D10:EmptyType } class WireBonding { type WireBondingType } class CopperToCopperHybridBonding { type JEP30-D10:EmptyType } class ThermoCompressionBonding { type JEP30-D10:EmptyType } class Other1 { type xs:string } class WireBondingType { type JEP30-D10:EmptyType } class ChipOnBoard { type JEP30-D10:EmptyType } class ChipOnGlass { type JEP30-D10:EmptyType } class Other2 { type xs:string } AssemblyTechnologyType "1" -- "*" AssemblyTechnology AssemblyTechnologyType "1" -- "*" SMTMaskReflow AssemblyTechnologyType "1" -- "*" PasteInHole AssemblyTechnologyType "1" -- "*" WaveSoldering AssemblyTechnologyType "1" -- "*" SelectiveMiniWaveSoldering AssemblyTechnologyType "1" -- "*" LaserSoldering AssemblyTechnologyType "1" -- "*" ManualSoldering AssemblyTechnologyType "1" -- "*" Pressfit AssemblyTechnologyType "1" -- "*" WireBonding AssemblyTechnologyType "1" -- "*" CopperToCopperHybridBonding AssemblyTechnologyType "1" -- "*" ThermoCompressionBonding AssemblyTechnologyType "1" -- "*" Other1 WireBonding "1" -- "*" WireBondingType WireBondingType "1" -- "*" ChipOnBoard WireBondingType "1" -- "*" ChipOnGlass WireBondingType "1" -- "*" Other2 </pre> | |
| type | JEP30-D10:EmptyType, WireBondingType | |

5.11 CTE - Array

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/CTE-Array. |
| diagram |  <p>The diagram illustrates the XSD structure for the CTE-Array. It shows a sequence of elements: CTE-Array (type CTE-ArrayType), CTE (type CTEType, with cardinality 1..∞), and CTE-Graph (type CTE-GraphType, with cardinality 0..∞). The CTEType is further detailed with elements: ID (type xs:string), Name (type xs:string), TemperatureCondition (type TemperatureConditionType), Value (type xs:decimal), CTE-UOM (type JEP30-D10:CTE-UOMType), RuleName (type xs:string), Rule (type xs:string), and x-Value (type xs:decimal), y-Value (type xs:decimal), and z-Value (type xs:decimal). The TemperatureConditionType is further detailed with elements: Value (type xs:decimal), x-Value (type xs:decimal), y-Value (type xs:decimal), and z-Value (type xs:decimal). The CTE-UOMType is further detailed with elements: RuleName (type xs:string) and Rule (type xs:string).</p> |
| type | CTE-ArrayType, CTEType, TemperatureConditionType, JEP30-D10:ValueSetType, TemperatureUOMType, JEP30-D10:CTE-UOMType, CTE-GraphType. |

The enumerated value for *CTE-UOM* is *ppm/degC*.

5.11.1 CTE - Graph

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/CTE-Array/CTE/CTE-Graph |
| diagram | |
| type | CTE-GraphType, CTE-ParametricGraphChartXAxisTypeType, UnitsForCTE-SpecificationType, JEP30-D10:GraphChartXAxisFormattingType, CTE-ParametricGraphChartYAxisTypeType, JEP30-D10:GraphChartYAxisFormattingType, CTE-ParameterGraphData-ArrayType, JEP30-D10:GraphFormattingType. |

A [ParameterGraph](#) has 2 axis that are defined by the [TestConditionType](#) (The X-axis definition), and the [ParameterType](#) (The Y-axis definition). Each axis is labelled by the [AxisTitle](#). When possible, the [Symbol](#) which represents the [AxisTitle](#) should be added to the PartModel file and should represent a standards-based symbol as defined in the appropriate Terms and Definitions standards. If appropriate, a more detailed [Description](#) can be used to describe the definition of the [AxisTitle](#). Each axis will also have a pre-defined set of [Units](#) but can be optionally excluded for those axis' which are unitless.

5.11.1 CTE - Graph (cont'd)

Note that the [ParameterDefinition](#) is unbounded whereas the [TestConditionDefinition](#) is bounded to a single instance. This is to cater for those graphs in which there are 2 or more y-axis, each with their own definition.

The graph can either be captured under the [Data-Array](#) or represented via a [GraphFormula](#) (A string representing the equation of the [ParameterDefinition](#) relationship to the Test [TestConditionDefinition](#)).

5.11.1.1 Formatting

The Formatting is an optional set of data that enables the user to re-create the graph for visualization purposes. Formatting applies to the following

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/CTE-Array/CTE/CTE-Graph/TestConditionDefinition/Formatting |
| diagram | |
| type | JEP30-D10:GraphChartXAxisFormattingType , GraphAxisRangeType , EmptyType , GraphAxisScaleType , GraphAxisScaleLinearType , GraphAxisScaleLogarithmicType , GraphChartXAxisPositionType . |

5.11.1.1 Formatting (cont'd)

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/CTE-Array/CTE/CTE-Graph/ParameterDefinition/Formatting |
| diagram | |
| type | JEP30-D10:GraphChartYAxisFormattingType, GraphAxisRangeType, EmptyType, GraphAxisScaleType, GraphAxisScaleLinearType, GraphAxisScaleLogarithmicType, GraphChartYAxisPositionType. |

The axis range which is usually defined from minimum to maximum can be inverted to show a graph going from maximum to minimum. The scale can be defined in either a linear step amount, a natural logarithm, or a logarithm of the specified base. The *Base* log is set to a default of *Base 10* but can be defined to any base number.

The *Position* enumerated list for the *GraphChartXAxisPositionType* is

- Top
- Bottom

And for the *GraphChartYAxisPositionType*, the enumerated values are

- Left
- Right

5.11.1.1 Formatting (cont'd)

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/CTE-Array/CTE/CTE-Graph/Formatting |
| diagram | |
| type | JEP30-D10:GraphChartYAxisFormattingType , GraphAxisRangeType , EmptyType , GraphAxisScaleType , GraphAxisScaleLinearType , GraphAxisScaleLogarithmicType , GraphChartYAxisPositionType . |

The body of the graph can be formatted under the [GraphFormattingType](#). The [DisplayType](#) enumerated list is

- Line
- Bar

The graph Legend can also be positioned around the graph in any of the following locations:

- Location
 - Inside Graph,
 - Outside Graph,
- Vertical Position
 - Top,
 - Center,
 - Bottom,
- Horizontal Position
 - Left,
 - Center,
 - Right.

5.12 Young Modulus - Array

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/YoungsModulus-Array. |
| diagram | |
| type | YoungsModulus-ArrayType, YoungsModulusType, TemperatureConditionType, JEP30-D10:ValueSetType, TemperatureUOMType, YoungsModulus-UOMType. |

The enumerated values for *YoungsModulus-UOM* are

- *N/m^2*, or
- *Giga-Pascal (GPa)*.

5.13 Terminal Groups

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups |
| diagram | |
| type | TerminalsGroupsType , TerminalGroup-ArrayType , TerminalGroupToTerminalGroupRelationships-ArrayType , TerminalDetail-ArrayType , ViaRegion-ArrayType . |

Since a package may have several terminal groups under the [TerminalGroup-Array](#), and since there may be dimensional relationships between some or all these groups, an [ID](#) is assigned to each [TerminalGroup](#) that can then be referenced in the [TerminalGroupToTerminalGroupRelationships-Array](#) branch.

5.13.1 Terminal Group-Array

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array |
| diagram | <p>The diagram illustrates the XSD structure for TerminalGroup-ArrayType. It is a sequence of TerminalGroup elements (1..∞). Each TerminalGroup contains the following elements:</p> <ul style="list-style-type: none"> ID (type: xs:string) TerminalPosition (type: TerminalPositionType) Terminal (type: TerminalType) TerminalMaterial (type: TerminalMaterialType) CTE-Array (type: CTE-ArrayType) Coplanarity (type: xs:decimal) FlexibleTerminal (type: xs:boolean) TerminalFunction (type: TerminalBasicFunctionType) TerminalLocation (type: TerminalLocationType, 0..∞) PatternGroup (type: PackageTerminalPatternGroupType, 0..∞) TerminalShape (type: TerminalShapeType) TerminalSpan (type: TerminalSpanType) TerminalSpacing (type: TerminalSpacingType) TerminalGroupToBodyRelationship (type: TerminalGroupToBodyRelationshipType) <p>A constraints box is located at the bottom of the diagram.</p> |
| type | TerminalsGroup-ArrayType, TerminalGroupType, TerminalPositionType, TerminalType, TerminalMaterialType, CTE-ArrayType, TerminalBasicFunctionType, TerminalLocationType, TerminalShapeType, TerminalSpanType, TerminalSpacingType, TerminalGroupToBodyRelationship. |

Coplanarity is defined in JESD88, JEDEC Dictionary of Terms for Solid-State Technology.

FlexibleTerminal is a new type of material applied to mostly ceramic chip component terminals for the purpose of reducing the mechanical stress between the component terminal and the printed board. This element is optional and can be set by the component manufacturer when they have incorporated this technology to the part construction.

5.13.1.1 Terminal Position

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalPosition. |
| diagram | <pre> classDiagram class TerminalPositionType { Axial Bottom Dual End Diagonal Internal Quad Radial Single Triple Upper ZigZag } class Axial { type JEP30-D10:EmptyType } class Bottom { type BottomUpperPositionConfigurationType } class Dual { type DualPositionConfigurationType } class End { type JEP30-D10:EmptyType } class Diagonal { type DiagonalCornerConfigurationType } class Internal { type JEP30-D10:EmptyType } class Quad { type QuadPositionConfigurationType } class Radial { type RadialPositionConfigurationType } class Single { type SinglePositionConfigurationType } class Triple { type TriplePositionConfigurationType } class Upper { type BottomUpperPositionConfigurationType } class ZigZag { type JEP30-D10:EmptyType } TerminalPositionType < -- Axial TerminalPositionType < -- Bottom TerminalPositionType < -- Dual TerminalPositionType < -- End TerminalPositionType < -- Diagonal TerminalPositionType < -- Internal TerminalPositionType < -- Quad TerminalPositionType < -- Radial TerminalPositionType < -- Single TerminalPositionType < -- Triple TerminalPositionType < -- Upper TerminalPositionType < -- ZigZag </pre> |
| type | TerminalPositionType , BottomUpperPositionConfigurationType , DualPositionConfigurationType , DiagonalCornerConfigurationType , QuadConfigurationType , RadialPositionConfigurationType , SinglePositionConfigurationType , TriplePositionConfigurationType . |

The definition of the above [TerminalPosition](#) data elements and their following sub-structures can be found in JESD30, under the “Terminal position Prefix” section which *includes* the “Prefixes for terminal position” table, plus three sections in Annex A, namely:

- Terminal position with additional definition,
- Relationship concepts between the Terminal Contact Area and the Body outline, and
- Position Images.

The only difference between this [TerminalPosition](#) type and the [PackageTerminalPosition](#) type is that the [MixedPosition](#) option is not available to a single [TerminalGroup](#). Refer to sub-sections of 4.5 Package Terminal Position for details of each of the sub-sections under [TerminalPosition](#).

5.13.1.2 Terminal

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/Terminal |
| diagram | <p>The diagram illustrates the structure of the TerminalType and its associated sub-structures. TerminalType is shown as a base class with a dashed box. It contains several attributes, each with a 'type' field indicating its data type. The attributes are: Ball (type: BallType), C-bend (type: JEP30-D10:EmptyType), Lug (type: LugType), Flat (type: FlatType), Gull-wing (type: Gull-wingType), CompressedMountTechnology (type: JEP30-D10:EmptyType), Post-Terminal (type: PostTerminalType), J-bend (type: JEP30-D10:EmptyType), L-bend (type: L-bendType), Column (type: ColumnType), Surface-terminal (type: Surface-terminalType), Surface-terminal (type: Surface-terminalType), Pressfit (type: PressfitType), Pin (type: PinType), Quick-connect (type: JEP30-D10:EmptyType), Wraparound (type: WraparoundType), S-bend (type: S-bendType), Through-Hole (type: Through-HoleType), J-inverted (type: JEP30-D10:EmptyType), TerminalWire (type: WireType), and Screw (type: ScrewType). A 'Terminal' attribute of type 'TerminalType' is also shown, connected to the main structure via a dashed line and a small icon.</p> |
| type | TerminalType, BallType, JEP30-D10:EmptyType, LugType, FlatType, Gull-wingType, PostTerminalType, L-BendType, ColumnType, SurfaceTerminalType, PressfitType, PinType, WraparoundType, S-BendType, Through-HoleType, WireType, ScrewType. |

The definition of the [TerminalCode](#) data elements and their following sub-structures can be found in the JESD30, under the “Terminal Shape Suffix” section which *includes* the “Suffixes for terminal shape” table, plus two sections in Annex A, namely

- Suffixes for terminal shape with additional definition, and
- Illustrations of terminal shape.

NOTE The structure for the [TerminalGroup/Terminal](#) that is under [Package/TerminalGroups/TerminalGroup-Array](#) is exactly the same as for [Package/PackageTerminalCode](#) as outlined in section 4.7 Package Terminal Code above, with the exception that the following three elements that are contained in the [Package/PackageTerminalCode](#) structure are not contained in the [TerminalGroup/Terminal](#) structure. This is because each single [TerminalGroup](#) can only represent a single [Terminal](#) Type.

5.13.1.2 Terminal (cont'd)

- Mixed-SMT,
- Mixed-TH, or
- Mixed-Technology.

Refer to 4.7 Package Terminal Code for details of each of the sub-sections under *Terminal*.

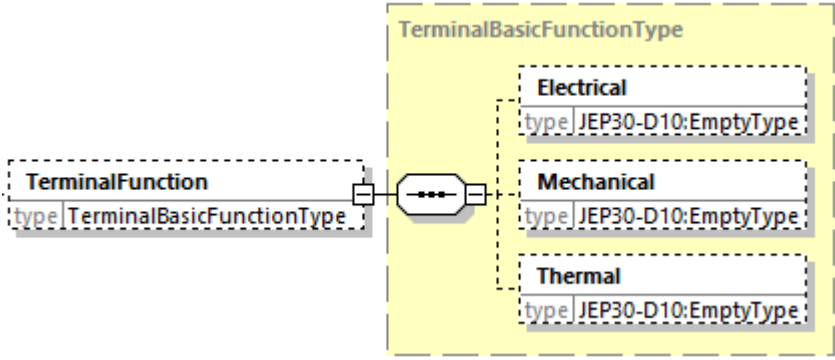
5.13.1.3 Terminal Material

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalMaterial . |
| diagram | |
| type | TerminalMaterialType , J-Std-609AlloyCompositionType , J-Std-609e-codeType , BaseMaterialType , Plating-ArrayType , PlatingType , PlatingThicknessUOMType . |

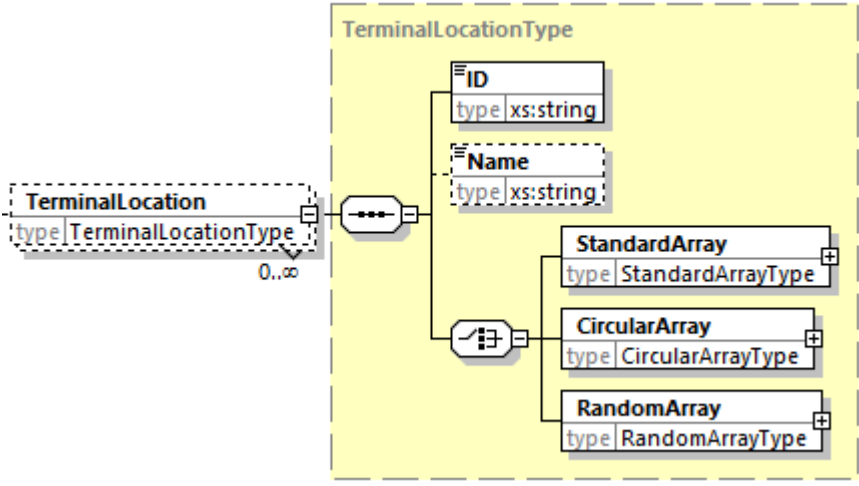
The *J-Std-609AlloyComposition* and *J-Std-609e-code* enumerated values are defined in “Annex A (informative) Example Alloys and Associated Material Codes” in the J-Std-609, “Marking, Symbols, and Labels of Leaded and Lead-Free Terminal Finished Materials Used in Electronic Assembly” standard.

The is *PlatingThicknessUOM* specified in *μm*.

5.13.1.4 Terminal Function

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalFunction. |
| diagram |  |
| type | TerminalBasicFunctionType , JEP30-D10:EmptyType . |

5.13.1.5 Terminal Location

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalLocation. |
| diagram |  |
| type | TerminalLocationType , StandardArrayType , CircularArrayType |

5.13.1.5.1 **Standard Array**

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalLocation |
| diagram | <p>The diagram illustrates the structure of the StandardArray type. It is a class with a type <code>StandardArrayType</code>. It contains several attributes and nested structures:</p> <ul style="list-style-type: none">nx: type <code>xs:integer</code>ny: type <code>xs:integer</code>dx: type <code>JEP30-D10:PitchValueSetType</code>. This attribute is associated with a JEP30-D10:PitchValueSetType structure, which includes:<ul style="list-style-type: none">Standard: type <code>StandardType</code>Limits: type <code>MinMaxLimitsType</code>PitchToleranceCumulative: type <code>xs:boolean</code>dy: type <code>JEP30-D10:PitchValueSetType</code>Angle: type <code>xs:integer</code>TerminalGroupLowerLeftTerminalCenter: type <code>JEP30-D10:PointXYType</code> |
| type | StandardArrayType, JEP30-D10:PitchValueSetType, StandardType, MinMaxLimitsType, JEP30-D10:PointType. |


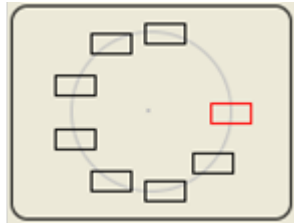
5.13.1.5.2 Circular Array

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalLocation/CircularArray. |
| diagram | <p>The diagram illustrates the structure of the CircularArray type and its associated components. On the left, a class box for CircularArray (type CircularArrayType) is connected to a large dashed yellow box representing the CircularArrayType structure. Inside this box, a central node branches into several attributes: PitchRadius (type xs:decimal), Center (type JEP30-D10:PointXYType), StartAngle (type xs:integer), NumberOfTerminals (type xs:integer), and Rotation (type RotationType). The Rotation attribute is further detailed in a separate dashed box labeled RotationType, which contains two options: RotateWithCircle (type JEP30-D10:EmptyType) and RotationAngle (type xs:integer). Additionally, a group of three attributes (AngleToFill and AngleBetweenTerminals, both type xs:integer) is shown connected to the central node via a choice relationship.</p> |
| type | CircularArrayType, JEP30-D10:PointXYType, RotationType, JEP30-D10:EmptyType. |

5.13.1.5.2 Circular Array (cont'd)

Under the [CircularArray](#), Table 3 provides the definition of each elements.

Table 3 - Terminal Circular Array Elements Definition

| Element | Explanation |
|-------------------------|--|
| Pitch Radius | The radius of the circle of terminals. |
| Center | The center of the circle of terminals. |
| Start Angle | The angle at which to place the first terminal in the array. By default, the terminal is at the right-most point on the circle (i.e. the number 3 position on a clock face). This angle specifies a rotation clockwise around the circle from this point. |
| Angle to Fill | The angle through which the terminals are distributed, starting from the first terminal and continuing clockwise, until the last terminal is reached. |
| Angle between Terminals | Instead of specifying Angle to fill , you can specify the angle between the centers of each terminal. |
| Number of Terminals | The number of terminals to be arrayed around the circle. The first terminal is displayed in red. This number includes any missing or deleted terminals in the array. |
| Rotate with Circle |  <p>If the terminals are rotated so that they are oriented perpendicular to the circle. This does not apply to circular or contour terminal shapes.</p> |
| Rotation Angle |  <p>If the terminals are not oriented perpendicular to the circle, then the terminals could be rotated around the center of the terminal itself. In this image, each terminal has a “0” degree rotation with respect to the “3 O’clock position”. This does not apply to circular or contour terminal shapes.</p> |

5.13.1.5.3 Random Array

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalLocation/RandomArray |
| diagram | |
| type | RandomArrayType , JEP30-D10:PointXYType . |

5.13.1.6 Pattern Group

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/PackageGroup |
| diagram | |
| type | PackageTerminalPatternGroupType , PackageTerminalPatternRelationshipType . |

5.13.1.6.1 Pattern Relationship

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/PackageGroup/PatternRelationship. |
| diagram | |
| type | PackageTerminalPatternRelationshipType, PackageRelationshipTransformationsType, TransformDuplicateType. |

5.13.1.6.1.1 Relationship Transformations

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/PackageGroup/PatternRelationship/RelationshipTransformations. |
| diagram | <p>The diagram illustrates the XSD structure for RelationshipTransformations. The root element, RelationshipTransformations (type PackageRelationshipTransformationsType), is composed of two main parts: Mirror and Rotate.</p> <ul style="list-style-type: none"> Mirror (type TransformMirrorType): This element includes a TransformMirrorType child, which contains: <ul style="list-style-type: none"> Origin (type JEP30-D10:EmptyType) SelectionCenter (type JEP30-D10:EmptyType) Horizontal (type JEP30-D10:EmptyType) Vertical (type JEP30-D10:EmptyType) Coordinate (type JEP30-D10:PointXYType) Rotate (type TransformRotateType): This element includes a TransformRotateType child, which contains: <ul style="list-style-type: none"> Origin (type JEP30-D10:EmptyType) SelectionCenter (type JEP30-D10:EmptyType) Coordinate (type JEP30-D10:PointXYType) Angle (type xs:decimal) <p>Additionally, an Offset element (type JEP30-D10:PointXYType) is shown with x and y attributes of type xs:decimal.</p> |
| type | PackageRelationshipTransformationsType, TransformMirrorType, TransformRotateType, JEP30-D10:EmptyType, JEP30-D10:PointXYType, |

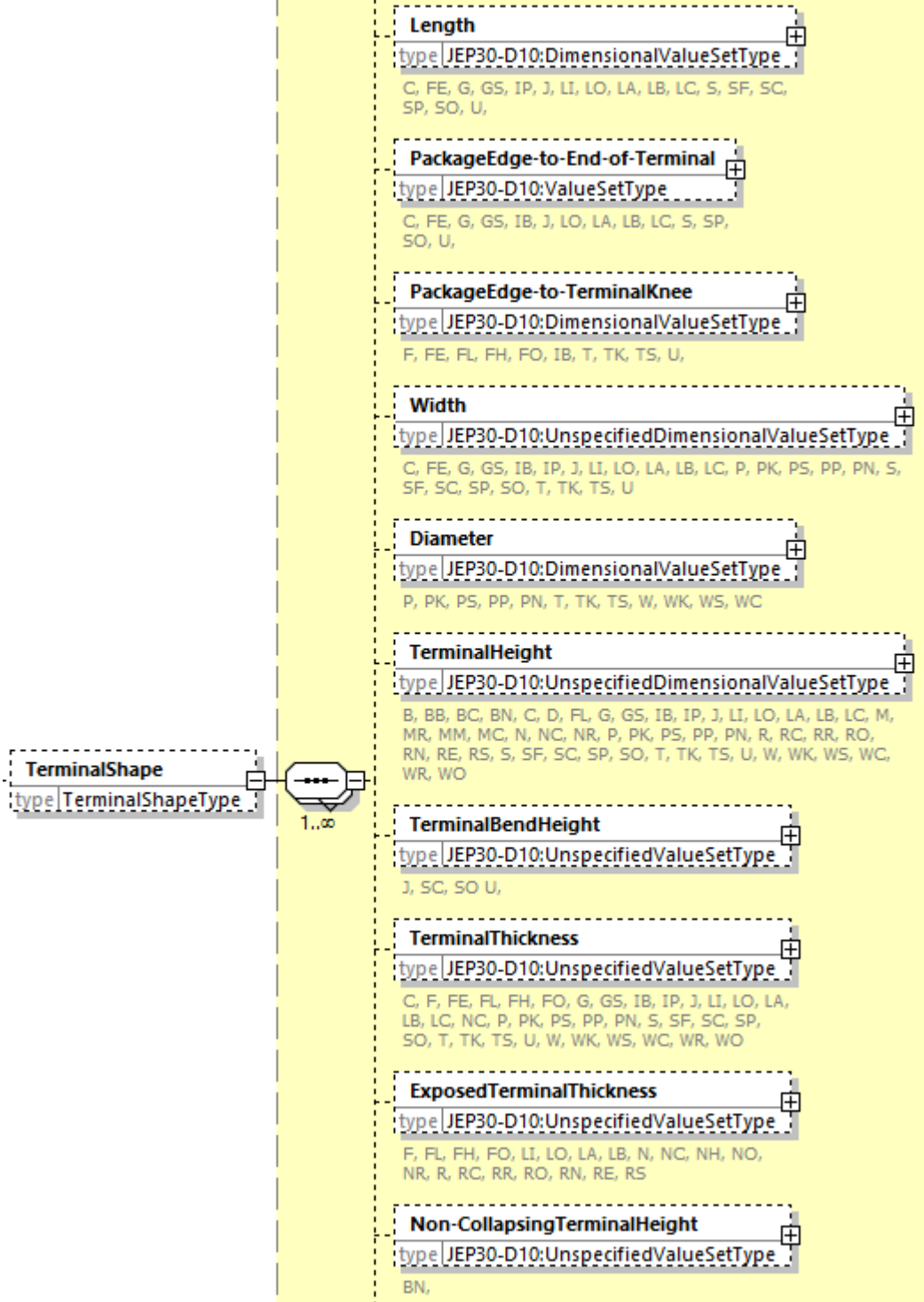
5.13.1.6.1.2 Duplicate

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/PatternGroup/PatternRelationship/Duplicate. |
| diagram | <p>The diagram, titled "TransformDuplicateType", illustrates the structure of a Duplicate relationship. It features several elements and their relationships:</p> <ul style="list-style-type: none">nx (type xs:integer) and ny (type xs:integer) are connected to a central node via a relationship with multiplicity 1..2.nx is connected to X-PrefixCode (type xs:string) and X-SuffixCode (type xs:string) via a relationship with multiplicity 1..2.ny is connected to Y-PrefixCode (type xs:string) and Y-SuffixCode (type xs:string) via a relationship with multiplicity 1..2.X-PrefixCode is connected to X-PrefixRowCode (type RowTerminalNumberOrderingType) and X-SuffixCode is connected to X-SuffixRowCode (type RowTerminalNumberOrderingType) via a relationship with multiplicity 1..2.Y-PrefixCode is connected to Y-PrefixColumnCode (type ColumnTerminalNumberOrderingType) and Y-SuffixCode is connected to Y-SuffixColumnCode (type ColumnTerminalNumberOrderingType) via a relationship with multiplicity 1..2.PrefixCode (type xs:string) is connected to PrefixNumericalSequence (type NumericalSequenceType) and PrefixAlphabeticalSequence (type AlphabeticalSequenceType) via a relationship with multiplicity 1..2.SuffixCode (type xs:string) is connected to SuffixNumericalSequence (type NumericalSequenceType) and SuffixAlphabeticalSequence (type AlphabeticalSequenceType) via a relationship with multiplicity 1..2.PrefixSeparator (type xs:string) and SuffixSeparator (type xs:string) are shown as dashed boxes, indicating they are optional or not used in all cases.dx (type xs:decimal) and dy (type xs:decimal) are shown as dashed boxes at the bottom, indicating they are optional or not used in all cases. |
| type | TransformDuplicateType, RowTerminalNumberOrderingType, ColumnTerminalNumberOrderingType, NumericalSequenceType, AlphabeticalSequenceType. |

5.13.1.7 Terminal Shape

| | |
|---------------------------|---|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalShape |
| diagram part 1 of 9 | <p>The diagram illustrates the structure of the TerminalShapeType class. It is a base class that defines the shape of a terminal. The subclasses include SMD, NSMD, Terminal, Core, SoldermaskOpening, and Pad. The Terminal class has two subclasses, LowerSide-of-Column and UpperSide-of-Column. The diagram also shows a relationship between TerminalShape and TerminalShapeType, where TerminalShape is a subclass of TerminalShapeType. The diagram is enclosed in a dashed box labeled TerminalShapeType.</p> |
| type | TerminalShapeType, JEP30-D10:EmptyType |

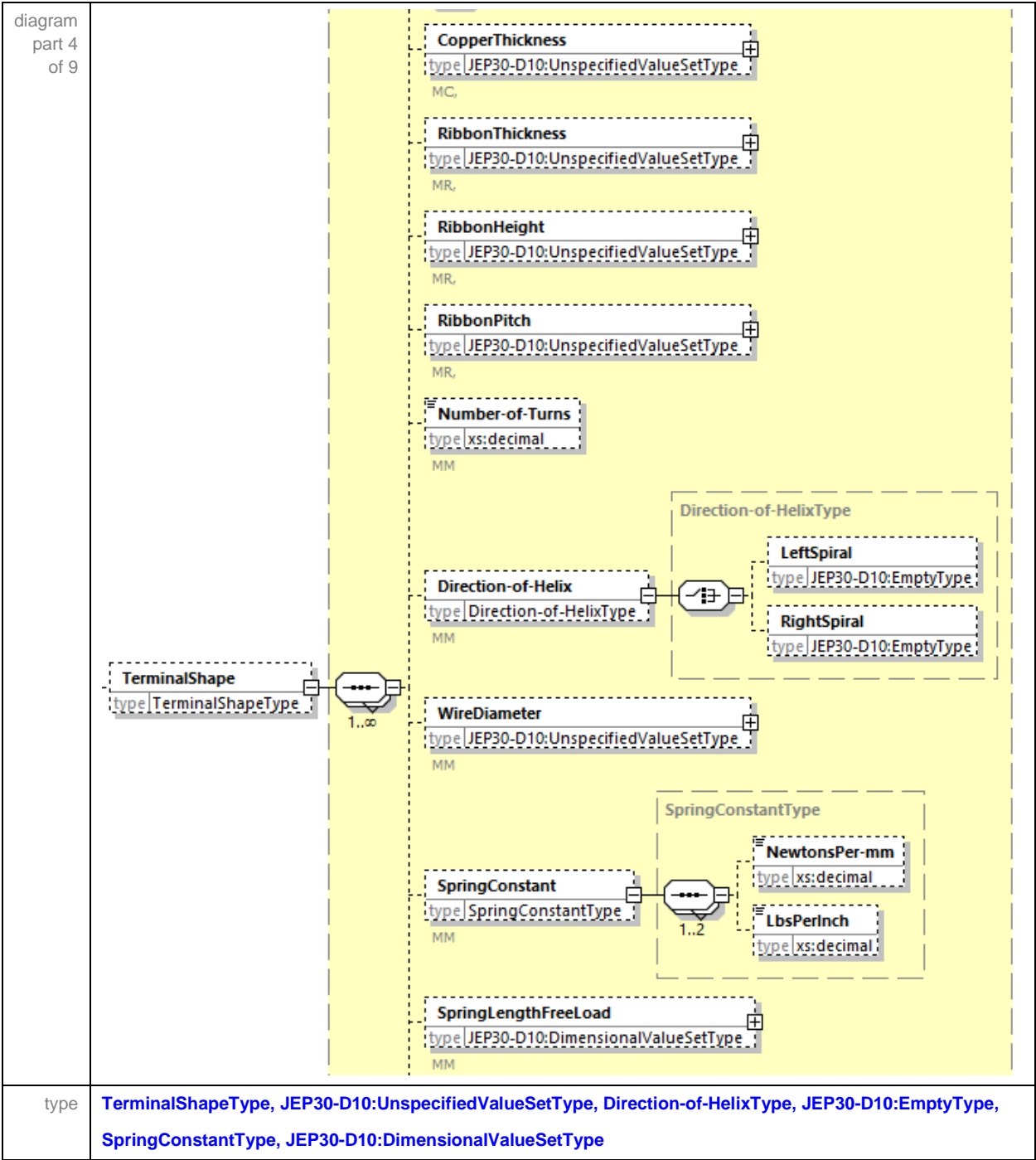
5.13.1.6 Terminal Shape (cont'd)

| | |
|------------------------------------|--|
| <p>diagram part 2 of 9</p> |  <p>TerminalShape type TerminalShapeType</p> <p>Length type JEP30-D10:DimensionalValueSetType C, FE, G, GS, IP, J, LI, LO, LA, LB, LC, S, SF, SC, SP, SO, U,</p> <p>PackageEdge-to-End-of-Terminal type JEP30-D10:ValueSetType C, FE, G, GS, IB, J, LO, LA, LB, LC, S, SP, SO, U,</p> <p>PackageEdge-to-TerminalKnee type JEP30-D10:DimensionalValueSetType F, FE, FL, FH, FO, IB, T, TK, TS, U,</p> <p>Width type JEP30-D10:UnspecifiedDimensionalValueSetType C, FE, G, GS, IB, IP, J, LI, LO, LA, LB, LC, P, PK, PS, PP, PN, S, SF, SC, SP, SO, T, TK, TS, U</p> <p>Diameter type JEP30-D10:DimensionalValueSetType P, PK, PS, PP, PN, T, TK, TS, W, WK, WS, WC</p> <p>TerminalHeight type JEP30-D10:UnspecifiedDimensionalValueSetType B, BB, BC, BN, C, D, FL, G, GS, IB, IP, J, LI, LO, LA, LB, LC, M, MR, MM, MC, N, NC, NR, P, PK, PS, PP, PN, R, RC, RR, RO, RN, RE, RS, S, SF, SC, SP, SO, T, TK, TS, U, W, WK, WS, WC, WR, WO</p> <p>TerminalBendHeight type JEP30-D10:UnspecifiedValueSetType J, SC, SO U,</p> <p>TerminalThickness type JEP30-D10:UnspecifiedValueSetType C, F, FE, FL, FH, FO, G, GS, IB, IP, J, LI, LO, LA, LB, LC, NC, P, PK, PS, PP, PN, S, SF, SC, SP, SO, T, TK, TS, U, W, WK, WS, WC, WR, WO</p> <p>ExposedTerminalThickness type JEP30-D10:UnspecifiedValueSetType F, FL, FH, FO, LI, LO, LA, LB, N, NC, NH, NO, NR, R, RC, RR, RO, RN, RE, RS</p> <p>Non-CollapsingTerminalHeight type JEP30-D10:UnspecifiedValueSetType BN,</p> |
| <p>type</p> | <p>TerminalShapeType, JEP30-D10:DimensionalValueSetType, JEP30-D10:ValueSetType, JEP30-D10:UnspecifiedDimensionalValueSetType, JEP30-D10:UnspecifiedValueSetType.</p> |

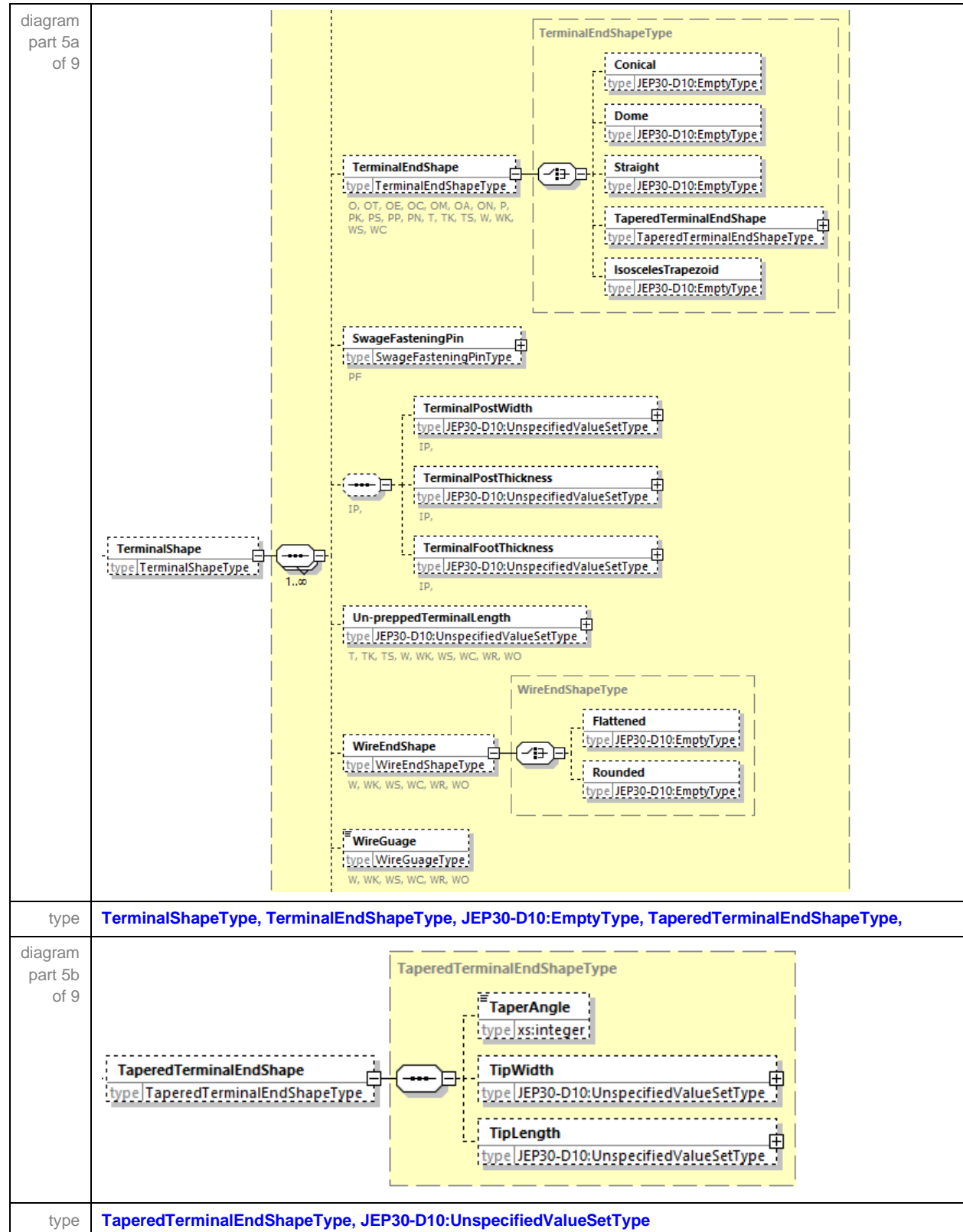
5.13.1.6 Terminal Shape (cont'd)

| | |
|------------------------------------|---|
| <p>diagram part 3 of 9</p> | |
| <p>type</p> | <p>TerminalShapeType, JEP30-D10:UnspecifiedValueType, JEP30-D10:DimensionalValueType, JEP30-D10:EmptyType,</p> |

5.13.1.6 Terminal Shape (cont'd)



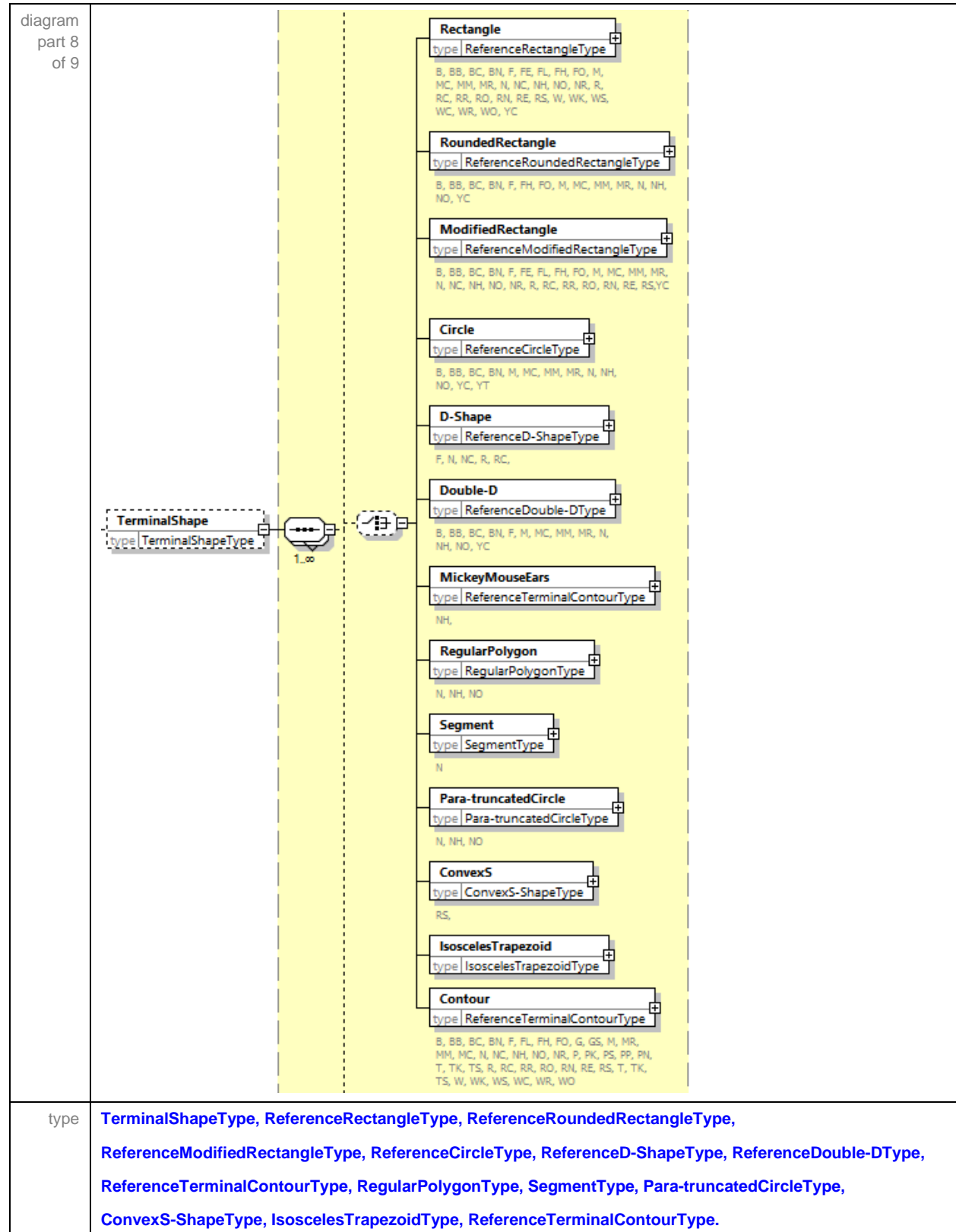
5.13.1.6 Terminal Shape (cont'd)



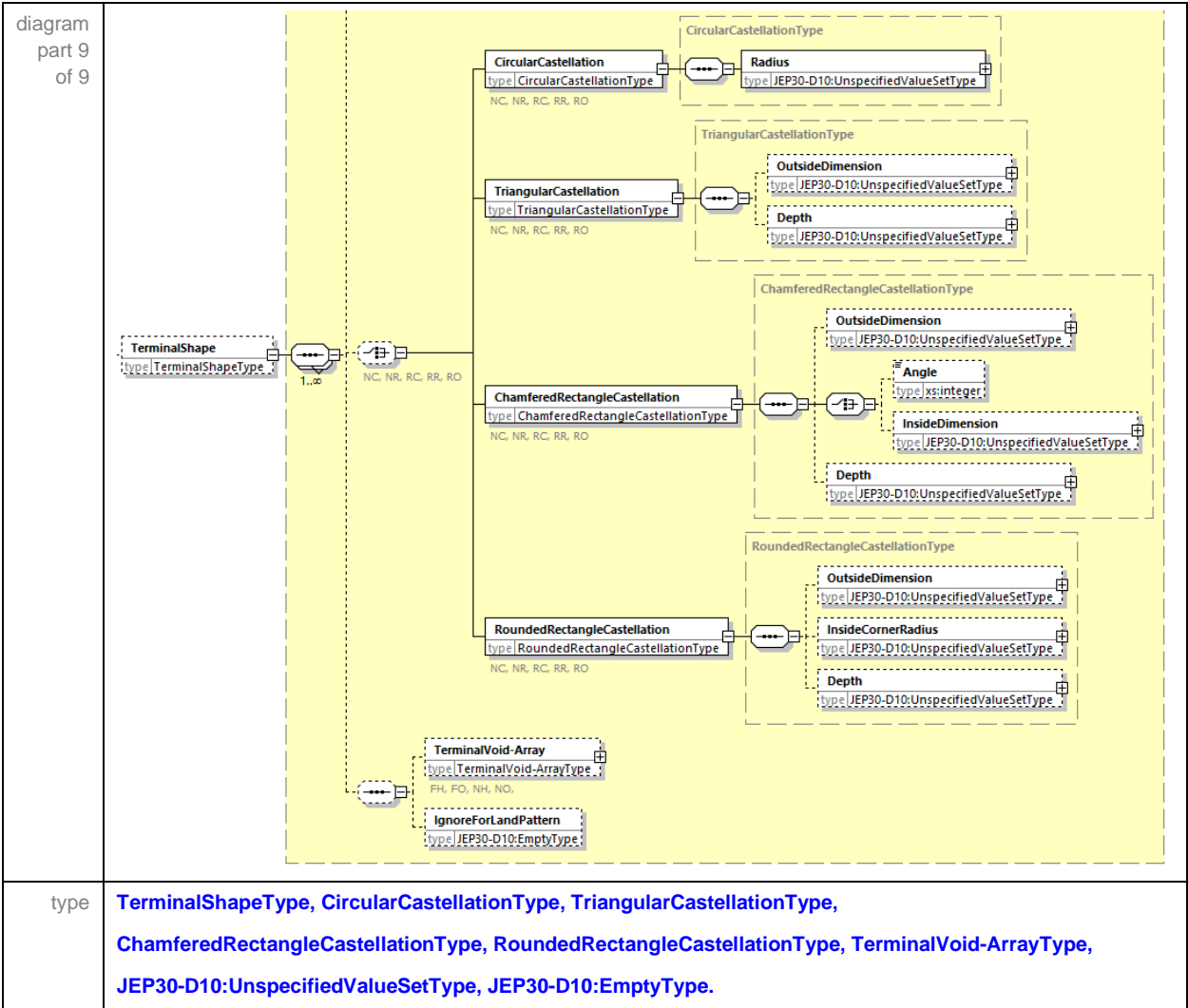
5.13.1.6 Terminal Shape (cont'd)

| | |
|------------------------------------|---|
| <p>diagram part 6 of 9</p> | |
| <p>type</p> | <p>TerminalShapeType, JEP30-D10:UnspecifiedValueType</p> |
| <p>diagram part 7 of 9</p> | |
| <p>type</p> | <p>TerminalShapeType, ModifiedCornerType, JEP30-D10:EmptyType, ConfigurableShoulderType, NeckType, DambarType, KinkedType.</p> |

5.13.1.6 Terminal Shape (cont'd)



5.13.1.6 Terminal Shape (cont'd)



5.13.1.7.1 Ball Types

Table 4 shows an array of dimensions that are applicable to this terminal:

Table 4 - Ball Dimensions

| Schema Labels | Ball Types | Applicable Shapes to capture | | | Schema Types |
|--------------------------------|----------------------------|------------------------------|--------------------|-----------|---|
| | | Terminal Shape | Soldermask Opening | Pad Shape | |
| Rectangle | All | | NSMD | SMD | ReferenceRectangleType |
| Rounded Rectangle | Bump | SMD, NSMD | SMD, NSMD | SMD, NSMD | ReferenceRoundedRectangleType |
| | Collapsing, Non-Collapsing | | NSMD | SMD | |
| Modified Rectangle | All | | NSMD | SMD | ReferenceModifiedRectangleType |
| Circle | All | SMD, NSMD | SMD, NSMD | SMD, NSMD | ReferenceCircleType |
| Double-D | All | | NSMD | SMD | ReferenceDouble-DType |
| Contour | Bump | | NSMD | SMD | ReferenceTerminalContourType |
| Terminal Height | All | SMD, NSMD | n/a | n/a | JEP30-D10: UnspecifiedDimensionalValueSetType |
| Non-Collapsing Terminal Height | All | SMD, NSMD | n/a | n/a | JEP30-D10:UnspecifiedValueSetType |

The definition of each shape is outlined in Annex A (informative) Shape Definitions & Dimensions.

The Terminal shapes is mandatory for all combinations of Ball types. The solderable surface of the terminal to the package may be defined by an opening in the soldermask resist layer as defined in Figure 1 – Solder Mask Defined Pad. For this type of SMD Pad, the Soldermask Opening shape is mandatory while the Pad shape is optional.

Alternatively, the solderable surface of the terminal to the package may be defined by the size of the metalized pad as defined in Figure 2 –Non Solder Mask Defined Pad. For this type of NSMD Pad, the Soldermask Opening shape is optional while the Pad shape is mandatory.

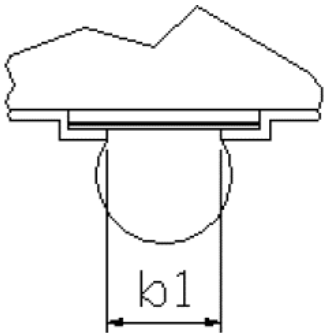


Figure 1 - Solder Mask Defined Pad

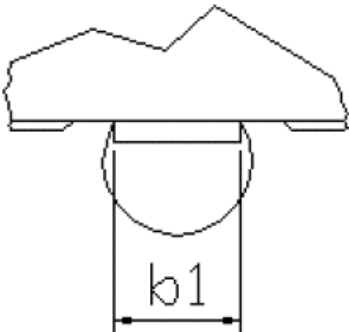


Figure 2 - Non Solder Mask Defined Pad

5.13.1.7.2 C-bend

Table 5 shows an array of dimensions that are applicable to this terminal.

Table 5 - C-bend Dimensions

| Schema Labels | Symbol | Schema Types | Data Requirements |
|--------------------------------|--------|--|---|
| Length | L | JEP30-D10:DimensionalValueSetType | Mandatory – but can be derived by other dimensions provided |
| PackageEdge-to-end-of-Terminal | L1 | JEP30-D10:ValueSetType | Mandatory – but can be derived by other dimensions provided |
| Width | b | JEP30-D10:UnspecifiedDimensionalValueSetType | Mandatory |
| TerminalHeight | A3 | JEP30-D10:UnspecifiedDimensionalValueSetType | Optional |
| TerminalThickness | c | JEP30-D10:UnspecifiedValueSetType | Mandatory |
| RadiusTop | R1 | JEP30-D10:UnspecifiedValueSetType | Optional |
| RadiusBottom | R2 | JEP30-D10:UnspecifiedValueSetType | Optional |
| SeatingAngle | Φ1 | xs:integer | Optional |
| RiseAngle | Φ2 | xs:integer | Optional |
| PlatingThickness | p | JEP30-D10:UnspecifiedValueSetType | Mandatory |

The definition of each shape is outlined in Annex A (informative) Shape Definitions & Dimensions

5.13.1.7.3 Lug

Table 6 shows an array of dimensions that are applicable to this terminal.

Table 6 - Lug Dimensions

| Schema Labels | Symbol | Schema Types | Data Requirements |
|-------------------------|--------|--|-------------------|
| TerminalHeight | A3 | JEP30-D10:UnspecifiedDimensionalValueSetType | Mandatory |
| TerminalInsertionHeight | | JEP30-D10:UnspecifiedValueSetType | Mandatory |
| Shoulder | | ConfigurableShoulderType | Optional |

In addition to this table of dimensions, any one of the 2D shapes as defined in diagram 2 of X in this section. The definition of each shape is outlined in Annex A (informative) Shape Definitions & Dimensions.

5.13.1.7.4 Flat

Table 7 shows an array of dimensions that are applicable to this terminal.

Table 7 - Flat Terminal Type Dimensions

| Schema Labels | Flat | Elevated | Flat – L-bend | Hole | With- opening | Schema Types |
|--|------|----------|------------------|------|------------------|--|
| One of the following 2D shapes is applicable to this Terminal type | | | | | | |
| Rectangle | Y | Y | Y | Y | Y | ReferenceRectangleType |
| RoundedRectangle | Y | | | Y | Y | ReferenceRoundedRectangleType |
| ModifiedRectangle | Y | Y | Y | Y | Y | ReferenceModifiedRectangleType |
| D-Shape | Y | | | | | ReferenceD-ShapeType |
| Double-D | Y | | | Y | Y | ReferenceDouble-DType |
| The dimensions below are mandatory if applicable | | | | | | |
| Terminal Height | | | Y | | | JEP30-D10: UnspecifiedDimensionalValueSetType |
| TerminalThickness | Y | Y | Y | Y | Y | JEP30-D10:UnspecifiedValueSetType |
| ExposedTerminalThickness | Opt | | Opt | Opt | Opt | JEP30-D10:UnspecifiedValueSetType |
| PackageEdge-to-End-of- Terminal | Y | Y | Y | Y | Y | JEP30-D10:ValueSetType |
| PlatingThickness | Y | Y | Y | Y | Y | JEP30-D10:UnspecifiedValueSetType |
| TerminalVoid-Array | | | | Y | Y | TerminalVoid-ArrayType |

The definition of each shape is outlined in Annex A (informative) Shape Definitions & Dimensions.

5.13.1.7.5 Gull-wing

Table 8 shows an array of dimensions that are applicable to this terminal.

Table 8 - Gull-wing Dimensions

| Schema Labels | Symbol | Schema Types | Data Requirements |
|--|--------|--|--|
| Length | L | JEP30-D10:DimensionalValueSetType | Mandatory – but can be derived by other dimensions provided |
| PackageEdge-to-end-of-Terminal | L1 | JEP30-D10:ValueSetType | Mandatory – but can be derived by other dimensions provided |
| Width | b | JEP30-D10:UnspecifiedDimensionalValueSetType | Mandatory |
| ModifiedCorner | | ImpactedCornerType | Optional |
| Contour | | ReferenceTerminalContourType | Optional |
| TerminalHeight | A3 | JEP30-D10:UnspecifiedDimensionalValueSetType | Optional |
| TerminalThickness | c | JEP30-D10:UnspecifiedValueSetType | Mandatory |
| RadiusTop | R1 | JEP30-D10:UnspecifiedValueSetType | Optional |
| RadiusBottom | R2 | JEP30-D10:UnspecifiedValueSetType | Optional |
| SeatingAngle | Φ1 | xs:integer | Optional |
| RiseAngle | Φ2 | xs:integer | Optional |
| PlatingThickness | p | JEP30-D10:UnspecifiedValueSetType | Mandatory |
| One of the following two elements is applicable to this Terminal type | | | |
| ToeUp | | JEP30-D10:EmptyType | If omitted, and seating angle is specified, then the default is ToeDown . |
| ToeDown | | JEP30-D10:EmptyType | |
| One of the following elements is applicable if there is a Shoulder or Neck on this Gull-wing | | | |
| Shoulder | | ConfigurableShoulderType | Optional |
| Neck | | NeckType | Optional |
| If the Gull-wing has a Dambar, then the following element is Optional | | | |
| Dambar | | DambarType | Optional |

Shoulder and Kinked are described in sections 4.15.1.5.15 Shoulder and 4.15.1.5.16 Kinked below, since they are applicable to other terminals.

5.13.1.7.5.1 Modified Corner

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalShape/ModifiedCorner |
| diagram | |
| type | ModifiedCornerType, ImpactedTerminalType, TerminalCenterType, Impact-to-TerminalGroupType, Apply-to-all-TerminalsType, SymmetryType, ImpactedCornerType, CornerType. |

5.13.1.7.5.1.1 Impacted Terminal

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalShape/ModifiedCorner/ImpactedTerminal |
| diagram | |
| type | ImpactedTerminalType, TerminalCenterType, JEP30-D10:PointXYType, JEP30-D10:MinIntegerOfOneType. |

5.13.1.7.5.1.2 Impact – to – Terminal Group

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalShape/ModifiedCorner/Impact-to-TerminalGroup |
| diagram | |
| type | Impact-to-TerminalGroupType , Apply-to-all-TerminalsType , SymmetryType , SymmetryRotationType , ReflectionType , |

5.13.1.7.5.1.2.1 Apply – to – all - Terminals

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalShape/ModifiedCorner/ShapeImpactedCorner |
| diagram | |
| type | Apply-to-all-TerminalsType , CornerImpact-to-StandardArrayType , JEP30-D10:EmptyType . |

5.13.1.7.5.1.2.2 Rotation

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalShape/ModifiedCorner/Impact-to-TerminalGroup/Symmetry/Rotation |
| diagram | |
| type | SymmetryRotationType, SymmetryRotationAxisType, SymmetryRotationCenterType, JEP30-D10:EmptyType. |

5.13.1.7.5.1.2.3 Reflection

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalShape/ModifiedCorner/Impact-to-TerminalGroup/Symmetry/Reflection |
| diagram | |
| type | ReflectionType, ReflectionAxisType, JEP30-D10:EmptyType, ReflectionInversionCenterType. |

5.13.1.7.5.1.3 Shape Impacted Corner

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalShape/ModifiedCorner/ShapeImpactedCorner |
| diagram | <pre> classDiagram class ShapeImpactedCorner { type ImpactedCornerType } class ImpactedCornerType { NE type CornerType SE type CornerType SW type CornerType NW type CornerType } class CornerType { ChamferedCorner type ChamferedCornerType RectangularInCorner type RectangularInCornerType ConvexCorner type ConvexCornerType ConcaveCorner type ConcaveCornerType CornerArc type CornerArcType } ShapeImpactedCorner -- ImpactedCornerType ImpactedCornerType -- CornerType ImpactedCornerType -- NE ImpactedCornerType -- SE ImpactedCornerType -- SW ImpactedCornerType -- NW </pre> |
| type | ImpactedCornerType, CornerType, ChamferedCornerType, RectangularInCornerType, ConvexCornerType, ConcaveCornerType, CornerArcType. |

5.13.1.7.5.1.4 Corner Type

| | |
|---------|--|
| path | <p>PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalShape/ModifiedCorner/ShapelImpactedCorner/NE</p> <p>PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalShape/ModifiedCorner/ShapelImpactedCorner/SE</p> <p>PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalShape/ModifiedCorner/ShapelImpactedCorner/SW</p> <p>PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalShape/ModifiedCorner/ShapelImpactedCorner/NW</p> |
| diagram | <pre> classDiagram class CornerType { type CornerType } class ChamferedCorner { type ChamferedCornerType } class RectangularInCorner { type RectangularInCornerType } class ConvexCorner { type ConvexCornerType } class ConcaveCorner { type ConcaveCornerType } class CornerArcType { type CornerArcType } CornerType "1" -- "*" ChamferedCorner CornerType "1" -- "*" RectangularInCorner CornerType "1" -- "*" ConvexCorner CornerType "1" -- "*" ConcaveCorner CornerType "1" -- "*" CornerArcType class ChamferedCornerType { DX JEP30-D10:UnspecifiedValueSetType DY JEP30-D10:UnspecifiedValueSetType } class RectangularInCornerType { DX JEP30-D10:UnspecifiedValueSetType DY JEP30-D10:UnspecifiedValueSetType } class ConvexCornerType { Radius JEP30-D10:UnspecifiedValueSetType ConvexCornerCenter JEP30-D10:PointXYType } class ConcaveCornerType { Radius JEP30-D10:UnspecifiedValueSetType ConcaveCornerCenter JEP30-D10:PointXYType } class CornerArcType { Radius1 JEP30-D10:UnspecifiedValueSetType Radius2 JEP30-D10:UnspecifiedValueSetType Neck JEP30-D10:UnspecifiedValueSetType CornerArcCenter JEP30-D10:PointXYType } </pre> |
| type | <p>ImpactedCornerType, CornerType, ChamferedCornerType, RectangularInCornerType, ConvexCornerType, ConcaveCornerType, CornerArcType.</p> |

5.13.1.7.5.2 Reference Terminal Contour

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalShape/ModifiedCorner |
| diagram | |
| type | ModifiedCornerType , ImpactedTerminalType , TerminalCenterType , Impact-to-TerminalGroupType , Apply-to-all-TerminalsType , SymmetryType , ImpactedCornerType , CornerType . |

5.13.1.7.5.2.1 Contour Tolerance Type

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalShape/Contour/X-Tolerance PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalShape/Contour/Y-Tolerance |
| diagram | |
| type | ContourToleranceType , JEP30-D10:SplitType , JEP30-D10:ToleranceUOMType . |

5.13.1.7.5.3 Neck

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalShape/Neck |
| diagram | |
| type | NeckType , JEP30-D10:UnspecifiedDimensionalValueSetType , NeckTransitionType , NeckTaperTransitionType , RightAngledNeckTransitionType , ShoulderConfigurationType |

5.13.1.7.5.3.1 Neck Transition

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalShape/Neck/NeckTransition |
| diagram | |
| type | ShoulderTransitionType , ShoulderTaperTransitionType , RightAngledShoulderTransitionType . |

5.13.1.7.5.3.1.1 Tapered

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalShape/Neck/NeckTransition/Tapered |
| diagram | |
| type | NeckTaperTransitionType, JEP30-D10:UnspecifiedValueSetType |

5.13.1.7.5.3.1.2 Right Angled

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalShape/Neck/NeckTransition/RightAngled |
| diagram | |
| type | RightAngledNeckTransitionType, JEP30-D10:UnspecifiedValueSetType |

5.13.1.7.5.3.2 Configuration

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalShape/Neck/Configuration |
| diagram | |
| type | NeckConfigurationType, JEP30-D10:EmptyType, NeckOffsetType, JEP30-D10:UnspecifiedValueSetType |

5.13.1.7.5.4 Dambar

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalShape/Dambar |
| diagram | |
| type | DambarType, DambarProtrusionType, |

The definition of each shape is outlined in Annex A (informative) Shape Definitions & Dimensions

5.13.1.7.6 Post (Stud) Terminal

Table 9 shows an array of dimensions that are applicable to this terminal.

Table 9 - Post (Stud) Terminal Dimensions

| Schema Labels | Butt | Flatten Post Connection | Schema Types | Data Requirements |
|--------------------------------|------|-------------------------|--|-------------------|
| Length | | Y | JEP30-D10:DimensionalValueSetType | Mandatory |
| PackageEdge-to-end-of-Terminal | Y | | JEP30-D10:ValueSetType | Mandatory |
| PackageEdge-to-TerminalKnee | Y | | JEP30-D10:DimensionalValueSetType | Optional |
| Width | Y | Y | JEP30-D10:UnspecifiedDimensionalValueSetType | Mandatory |
| TerminalHeight | Y | Y | JEP30-D10:UnspecifiedDimensionalValueSetType | Mandatory |
| TerminalThickness | Y | Opt | JEP30-D10:UnspecifiedValueSetType | Mandatory |
| RadiusTop | Y | | JEP30-D10:UnspecifiedValueSetType | Optional |
| RiseAngle | Y | | xs:integer | Optional |
| TerminalPostWidth | | Y | JEP30-D10:UnspecifiedValueSetType | Optional |
| TerminalPostThickness | | Y | JEP30-D10:UnspecifiedValueSetType | Optional |
| TerminalFootThickness | | Y | JEP30-D10:UnspecifiedValueSetType | Optional |
| PlatingThickness | Y | Y | JEP30-D10:UnspecifiedValueSetType | Mandatory |

The definition of each shape is outlined in Annex A (informative) Shape Definitions & Dimensions.

5.13.1.7.7 J-bend

Table 10 shows an array of dimensions that are applicable to this terminal.

Table 10 - J-bend Dimensions

| Schema Labels | Symbol | Schema Types | Data Requirements |
|--|--------|--|---|
| Length | L | JEP30-D10:DimensionalValueSetType | Mandatory – but can be derived by other dimensions provided |
| PackageEdge-to-end-of-Terminal | L1 | JEP30-D10:ValueSetType | Mandatory – but can be derived by other dimensions provided |
| Width | b | JEP30-D10:UnspecifiedDimensionalValueSetType | Mandatory |
| TerminalHeight | A3 | JEP30-D10:UnspecifiedDimensionalValueSetType | Optional |
| TerminalBendHeight | | JEP30-D10:UnspecifiedValueSetType | Optional |
| TerminalThickness | c | JEP30-D10:UnspecifiedValueSetType | Mandatory |
| RadiusTop | R1 | JEP30-D10:UnspecifiedValueSetType | Optional |
| RadiusBottom | R2 | JEP30-D10:UnspecifiedValueSetType | Optional |
| RiseAngle | Φ2 | xs:integer | Optional |
| PlatingThickness | p | JEP30-D10:UnspecifiedValueSetType | Mandatory |
| If the Terminal has a Shoulder, then the following elements is applicable. | | | |
| Shoulder | | ConfigurableShoulderType | Optional |

The definition of each shape is outlined in Annex A (informative) Shape Definitions & Dimensions.

5.13.1.7.8 L-bend

Table 11 shows an array of dimensions that are applicable to this terminal.

Table 11 - L-bend Dimensions

| Schema Labels | Inward | Outward | Side Inward | Side Outward | LC-bend | Schema Types |
|--------------------------------|--------|---------|----------------|-----------------|---------|--|
| Length | Y | Y | Y | Y | Y | JEP30-D10:DimensionalValueSetType |
| PackageEdge-to-End-of-Terminal | | Y | Opt | Y | Y | JEP30-D10:ValueSetType |
| Width | Y | Y | Y | Y | Y | JEP30-D10:UnspecifiedDimensionalValueSetType |
| Terminal Height | Y | Y | Y | Y | Y | JEP30-D10:UnspecifiedDimensionalValueSetType |
| TerminalThickness | Y | Y | Opt | Opt | Y | JEP30-D10:UnspecifiedValueSetType |
| ExposedTerminalThickness | Y | Y | Y | Y | | JEP30-D10:UnspecifiedValueSetType |
| RadiusBottom | Y | Y | Y | Y | | JEP30-D10:UnspecifiedValueSetType |
| RadiusInsideBottom | | | | | Y | JEP30-D10:UnspecifiedValueSetType |
| RadiusOutsideBottom | | | | | Y | JEP30-D10:UnspecifiedValueSetType |
| SeatingAngle | Y | Y | Y | Y | Y | xs:integer |
| PlatingThickness | Y | Y | Y | Y | Y | JEP30-D10:UnspecifiedValueSetType |

The definition of each shape is outlined in Annex A (informative) Shape Definitions & Dimensions.

5.13.1.7.9 Column Types

Table 12 shows an array of dimensions that are applicable to this terminal:

Table 12 - Column Type Dimensions

| Schema Labels | Column Types | Applicable Shapes to capture | | | Schema Types |
|---------------------|--------------|------------------------------|--------------------|-----------|---|
| | | Terminal Shape | Soldermask Opening | Pad Shape | |
| Rectangle | All | | NSMD | SMD | ReferenceRectangleType |
| Rounded Rectangle | All | | NSMD | SMD, NSMD | ReferenceRoundedRectangleType |
| Modified Rectangle | All | | NSMD | SMD | ReferenceModifiedRectangleType |
| Circle | All | SMD, NSMD | SMD, NSMD | SMD, NSMD | ReferenceCircleType |
| Double-D | All | | NSMD | SMD | ReferenceDouble-DType |
| Contour | All | | NSMD | SMD | ReferenceTerminalContourType |
| Terminal Height | All | SMD, NSMD | n/a | n/a | JEP30-D10: UnspecifiedDimensionalValueSetType |
| UpperSide-of-Column | Column | Y | | | JEP30-D10:EmptyType |
| LowerSide-of-Column | Column | Y | | | JEP30-D10:EmptyType |

All Column variations have only a circular terminal shape, even if mounted on the package via a SMD or a NSMD pad.

When *Column* has an no extended terminal code, then the Terminal shape dimensions should be classified for either the *LowerSide-of-Column* and/or the *UpperSide-of-Column*. If both ends of the are different (i.e., Column is tapered), then two entries are required to capture both shape dimensions. The *LowerSide-of-Column* terminal shape is mandatory, whereas the *UpperSide-of-Column* terminal shape dimensions are optional. If the column is not tapered, then both of these flags are set for the same dimension captured.

When the *Column* has an extended terminal code to denote *Ribbon Wrap*, then the element *RibbonThickness*, *RibbonHeight*, *RibbonPitch* are applicable. These elements have an *UnspecifiedValueSetType* and are optional. The *Number-of-Turns* in the helix can also be captured in addition to the *Direction-of-Helix* as in *LeftSpiral* or *RightSpiral*.

When the *Column* has an extended terminal code to denote *Microspring*, then the element *SpringConstant*, *WireDiameter* and *SpringLengthFreeLoad* are applicable. The *WireDiameter* element has an *UnspecifiedValueSetType* and is optional. The *SpringLengthFreeLoad* element has an *DimensionalValueSetType* and is mandatory.

When the *Column* has an extended terminal code to denote *Copper Coated Solder Column*, then the element *CopperThickness* is applicable. This has an *UnspecifiedValueSetType* and is optional. An additional entry is required to capture the *Core* shape dimensions.

The definition of each shape is outlined in Annex A (informative) Shape Definitions & Dimensions.

5.13.1.7.10 Surface-terminal

Table 13 shows an array of dimensions that are applicable to this terminal.

Table 13 - Surface-terminal Dimensions

| Schema Labels | Surface-terminal | Castellated | Hole | With-opening | Open-Ring | Schema Types |
|--|------------------|-------------|------|--------------|-----------|--|
| One of the following 2D shapes is applicable to this Terminal type | | | | | | |
| Rectangle | Y | Y | Y | Y | Y | ReferenceRectangleType |
| RoundedRectangle | Y | | Y | Y | | ReferenceRoundedRectangleType |
| ModifiedRectangle | Y | Y | Y | Y | Y | ReferenceModifiedRectangleType |
| Circle | Y | | Y | Y | | ReferenceCircleType |
| D-Shape | Y | Y | | | | ReferenceD-ShapeType |
| Double-D | Y | | Y | Y | | ReferenceDouble-DType |
| MickeyMouseEars | | | Y | | | ReferenceTerminalContourType |
| RegularPolygon | Y | | Y | Y | | RegularPolygonType |
| Segment | Y | | | | | SegmentType |
| Para-truncatedCircle | Y | | Y | Y | | |
| Contour | Y | Y | Y | Y | Y | ReferenceTerminalContourType |
| The elements below are mandatory if applicable | | | | | | |
| ExposedTerminalThickness | Y | Y | Y | Y | Y | JEP30-D10:UnspecifiedValueSetType |
| Terminal Height | Y | Y | | | Y | JEP30-D10:UnspecifiedDimensionalValueSetType |
| TerminalVoid-Array | | | Y | Y | | TerminalVoid-ArrayType |
| The elements below are optional if applicable | | | | | | |
| CircularCastellation | | Y | | | Y | CircularCastellationType |
| TriangularCastellation | | Y | | | Y | TriangularCastellationType |
| ChamferedRectangleCastellation | | Y | | | Y | ChamferedRectangleCastellationType |
| RoundedRectangleCastellation | | Y | | | Y | RoundedRectangleCastellationType |

The definition of each shape is outlined in Annex A (informative) Shape Definitions & Dimensions.

5.13.1.7.11 Pin

Table 14. shows an array of dimensions that are applicable to this terminal.

Table 14 - Pin Dimensions

| Schema Labels | Pin | Kinked | Shoulder | Press-in Solderable | Swage Fastening Pin | Press-in Non-Solderable | Schema Types |
|-------------------------|-----|--------|----------|---------------------|---------------------|-------------------------|--|
| TerminalInsertionHeight | Y | Y | Y | Y | | Y | JEP30-D10:UnspecifiedValueSetType |
| Diameter | Y | Y | Y | Y | | Y | JEP30-D10: DimensionalValueSetType |
| Width | Y | Y | Y | Y | | Y | JEP30-D10:UnspecifiedDimensionalValueSetType |
| TerminalThickness | Y | Y | Y | Y | | Y | JEP30-D10:UnspecifiedValueSetType |
| TerminalHeight | Y | Y | Y | Y | | Y | JEP30-D10:UnspecifiedDimensionalValueSetType |
| PlatingThickness | Y | Y | Y | Y | | | JEP30-D10:UnspecifiedValueSetType |
| Contour | Y | Y | Y | Y | | Y | ReferenceTerminalContourType |
| Kinked | | Y | | | | | KinkedType |
| Shoulder | | | Y | Y | | Y | ConfigurableShoulderType |
| SwageFasteningPin | | | | | Y | | SwageFasteningPinType |
| TerminalEndShape | Y | Y | Y | Y | | Y | TerminalEndShapeType |

The definition of each shape is outlined in Annex A (informative) Shape Definitions & Dimensions.

5.13.1.7.11.1 Swage Fastening Pin

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalShape/SwageFasteningPin |
| diagram | |
| type | SwageFasteningPinType, JEP30-D10:DimensionalValueSetType, SwageFasteningPinTailStyleType, |

5.13.1.7.11.1.1 Tail Style

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalShape/SwageFasteningPin/TailStyle |
| diagram | |
| type | SwageFasteningPinTailStyleType, SwageFasteningPinSlottedTailStyleType, SwageFasteningPinTurretTailStyleType |

5.13.1.7.11.1.1 Slotted

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalShape/SwageFasteningPin/TailStyle/Slotted |
| diagram | |
| type | SwageFasteningPinSlottedTailStyleType , JEP30-D10:DimensionalValueSetType . |

5.13.1.7.11.1.2 Turret

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalShape/SwageFasteningPin/TailStyle/Turret |
| diagram | |
| type | SwageFasteningPinTurretTailStyleType , JEP30-D10:DimensionalValueSetType . |

5.13.1.7.12 Wraparound

Table 15 shows an array of dimensions that are applicable to this terminal.

Table 15 - Wraparound Dimensions

| Schema Labels | Wraparound | Castellated | Ring | Open-Ring | Nibble | Convex-E | Convex-S | Schema Types |
|--|------------|-------------|------|-----------|--------|----------|----------|--|
| One of the following 2D shapes is applicable to this Terminal type | | | | | | | | |
| Rectangle | Y | Y | Y | Y | Y | Y | Y | ReferenceRectangleType |
| ModifiedRectangle | Y | Y | Y | Y | Y | | Y | ReferenceModifiedRectangleType |
| Contour | Y | Y | Y | Y | Y | | Y | ReferenceTerminalContourType |
| The elements below are mandatory if applicable | | | | | | | | |
| ExposedTerminalThickness | Y | Y | Y | Y | Y | Y | Y | JEP30-D10:UnspecifiedValueSetType |
| Terminal Height | Y | Y | Y | Y | Y | Y | Y | JEP30-D10:UnspecifiedDimensionalValueSetType |
| The elements below are optional if applicable | | | | | | | | |
| CircularCastellation | | Y | Y | Y | | | | CircularCastellationType |
| TriangularCastellation | | Y | Y | Y | | | | TriangularCastellationType |
| ChamferedRectangleCastellation | | Y | Y | Y | | | | ChamferedRectangleCastellationType |
| RoundedRectangleCastellation | | Y | Y | Y | | | | RoundedRectangleCastellationType |

The definition of each shape is outlined in Annex A (informative) Shape Definitions & Dimensions.

5.13.1.7.13 S-bend Terminal

Table 16 shows an array of dimensions that are applicable to this terminal.

Table 16 - S-bend Dimensions

| Schema Labels | S-bend | Inward Flat | Inward Curved | Outward Flat | Outward Curved | Schema Types |
|--------------------------------|--------|----------------|------------------|-----------------|-------------------|--|
| Length | Y | Y | Y | Y | Y | JEP30-D10:DimensionalValueSetType |
| PackageEdge-to-End-of-Terminal | Y | | | Y | Y | JEP30-D10:ValueSetType |
| Width | Y | Y | Y | Y | Y | JEP30-D10:UnspecifiedDimensionalValueSetType |
| Terminal Height | Opt | Opt | Opt | Opt | Opt | JEP30-D10:UnspecifiedDimensionalValueSetType |
| TerminalBendHeight | | | Opt | | Opt | JEP30-D10:UnspecifiedValueSetType |
| TerminalThickness | Y | Y | Y | Y | Y | JEP30-D10:UnspecifiedValueSetType |
| RadiusTop | Y | Y | Y | Y | Y | JEP30-D10:UnspecifiedValueSetType |
| RadiusBottom | Y | Y | Y | Y | Y | JEP30-D10:UnspecifiedValueSetType |
| PlatingThickness | Y | Y | Y | Y | Y | JEP30-D10:UnspecifiedValueSetType |

The definition of each shape is outlined in Annex A (informative) Shape Definitions & Dimensions.

5.13.1.7.14 Through - Hole

Table 17 — Through - Hole Dimensions shows an array of dimensions that are applicable to this terminal.

Table 17 - Through - Hole Dimensions

| Schema Labels | Through - Hole | Kinked | Shoulder | Schema Types |
|-----------------------------|----------------|--------|----------|--|
| PackageEdge-to-TerminalKnee | Y | Y | Y | JEP30-D10: DimensionalValueSetType |
| Diameter | Y | Y | Y | JEP30-D10: DimensionalValueSetType |
| Width | Y | Y | Y | JEP30-D10:UnspecifiedDimensionalValueSetType |
| TerminalThickness | Y | Y | Y | JEP30-D10:UnspecifiedValueSetType |
| TerminalHeight | Y | Y | Y | JEP30-D10:UnspecifiedDimensionalValueSetType |
| TerminalInsertionHeight | Y | Y | Y | JEP30-D10:UnspecifiedValueSetType |
| Un-preppedTerminalLength | Y | Y | Y | JEP30-D10:UnspecifiedValueSetType |
| RadiusTop | Y | Y | Y | JEP30-D10:UnspecifiedValueSetType |
| RiseAngle | Y | Y | Y | xs:integer |
| PlatingThickness | Y | Y | Y | JEP30-D10:UnspecifiedValueSetType |
| TerminalEndShape | Y | Y | Y | TerminalEndShapeType |
| Contour | Y | Y | Y | ReferenceTerminalContourType |
| Kinked | | Y | | KinkedType |
| Shoulder | | | Y | ConfigurableShoulderType |

The definition of each shape is outlined in Annex A (informative) Shape Definitions & Dimensions

5.13.1.7.15 J-inverted

Table 18 shows an array of dimensions that are applicable to this terminal.

Table 18 - J-inverted Dimensions

| Schema Labels | Symbol | Schema Types | Data Requirements |
|--|--------|--|---|
| Length | L | JEP30-D10:DimensionalValueSetType | Mandatory – but can be derived by other dimensions provided |
| PackageEdge-to-end-of-Terminal | L1 | JEP30-D10:ValueSetType | Mandatory – but can be derived by other dimensions provided |
| PackageEdge-to-TerminalKnee | | JEP30-D10:DimensionalValueSetType | Optional |
| Width | b | JEP30-D10:UnspecifiedDimensionalValueSetType | Mandatory |
| TerminalHeight | A3 | JEP30-D10:UnspecifiedDimensionalValueSetType | Optional |
| TerminalBendHeight | | JEP30-D10:UnspecifiedValueSetType | Optional |
| TerminalThickness | c | JEP30-D10:UnspecifiedValueSetType | Mandatory |
| RadiusTop | R1 | JEP30-D10:UnspecifiedValueSetType | Optional |
| RadiusBottom | R2 | JEP30-D10:UnspecifiedValueSetType | Optional |
| RiseAngle | Φ2 | xs:integer | Optional |
| PlatingThickness | p | JEP30-D10:UnspecifiedValueSetType | Mandatory |
| If the Terminal has a Shoulder, then the following elements is applicable. | | | |
| Shoulder | | ConfigurableShoulderType | Optional |

The definition of each shape is outlined in Annex A (informative) Shape Definitions & Dimensions.

5.13.1.7.16 Terminal Wire

Table 19 shows an array of dimensions that are applicable to this terminal.

Table 19 - Terminal Wire Dimensions

| Schema Labels | Terminal Wire | Kinked (Bent) | SM-Wire | SM-Coined Wire | Wraparound SMT Mount | Coined Wraparound SMT Mount | Schema Types |
|--------------------------|---------------|---------------|---------|----------------|----------------------|-----------------------------|--|
| TerminalInsertionHeight | Y | Y | Y | Y | | | JEP30-D10:UnspecifiedValueSetType |
| Diameter | Y | Y | Y | Y | | | JEP30-D10: DimensionalValueSetType |
| WireGuage | Y | Y | Y | Y | Y | Y | WireGuageType |
| TerminalHeight | Y | Y | Y | Y | Y | Y | JEP30-D10:UnspecifiedDimensionalValueSetType |
| TerminalThickness | Y | Y | Y | Y | Y | Y | JEP30-D10:UnspecifiedValueSetType |
| PlatingThickness | Y | Y | Y | Y | | | JEP30-D10:UnspecifiedValueSetType |
| Kinked | | Y | | | | | KinkedType |
| Un-preppedTerminalLength | Y | | | | | | JEP30-D10:UnspecifiedValueSetType |
| Contour | Y | Y | Y | Y | Y | Y | ReferenceTerminalContourType |
| Rectangle | Y | Y | Y | Y | Y | Y | ReferenceRectangleType |
| WireEndShape | Y | Y | Y | Y | | | WireEndShapeType |
| TerminalEndShape | Y | Y | Y | Y | | | TerminalEndShapeType |

WireGuage is a numerical integer value between 10 and 40. The definition of each shape is outlined in Annex A (informative) Shape Definitions & Dimensions.

5.13.1.7.17 Screw

Table 20 shows an array of dimensions that are applicable to this terminal.

Table 20 - Screw Dimensions

| Schema Labels | Clearance Hole | Threaded Hole | Screw | Schema Types |
|--------------------------|----------------|---------------|-------|-----------------------------------|
| Rectangle | Y | | | ReferenceRectangleType |
| RoundedRectangle | Y | | | ReferenceRoundedRectangleType |
| ModifiedRectangle | Y | | | ReferenceModifiedRectangleType |
| Circle | Y | Y | | ReferenceCircleType |
| Double-D | Y | | | ReferenceDouble-DType |
| ScrewHeadType | | | Y | Xs: string |
| ScrewHeadHeight | | | Y | JEP30-D10:UnspecifiedValueSetType |
| ScrewThreadLength | | | Y | JEP30-D10:UnspecifiedValueSetType |
| ScrewThreadPitch | | Y | Y | JEP30-D10:UnspecifiedValueSetType |
| ScrewThreadDepth | | Y | Y | JEP30-D10:UnspecifiedValueSetType |
| ScrewThreadMajorDiameter | | Y | Y | JEP30-D10:UnspecifiedValueSetType |
| ScrewThreadMinorDiameter | | Y | Y | JEP30-D10:UnspecifiedValueSetType |
| ScrewThreadPitchDiameter | | Y | Y | JEP30-D10:UnspecifiedValueSetType |

The definition of each shape is outlined in Annex A (informative) Shape Definitions & Dimensions.

5.13.1.7.18 Shoulder

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalShape/Shoulder |
| diagram | |
| type | ConfigurableShoulderType, JEP30-D10:UnspecifiedDimensionalValueSetType, ShoulderTransitionType, ShoulderConfigurationType, JEP30-D10:EmptyType. |

5.13.1.7.18.1 Shoulder Transition

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalShape/Shoulder/ShoulderTransition |
| diagram | |
| type | ShoulderTransitionType, ShoulderTaperTransitionType, RightAngledShoulderTransitionType. |

5.13.1.7.18.1.1 Tapered

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalShape/Shoulder/ShoulderTransition/Tapered |
| diagram | |
| type | ShoulderTaperTransitionType, JEP30-D10:UnspecifiedValueSetType |

5.13.1.7.18.1.2 Right Angled

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalShape/Shoulder/ShoulderTransition/RightAngled |
| diagram | |
| type | RightAngledShoulderTransitionType, JEP30-D10:UnspecifiedValueSetType |

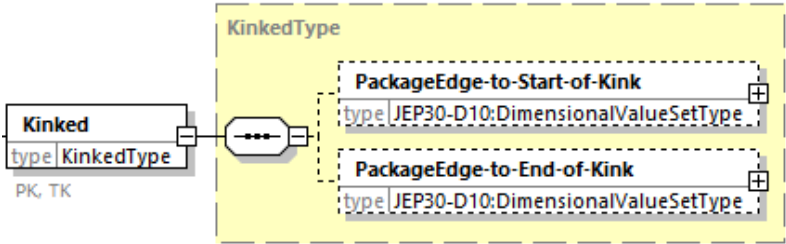
5.13.1.7.18.2 Configuration

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalShape/Shoulder/Configuration |
| diagram | |
| type | ShoulderConfigurationType, JEP30-D10:EmptyType, ShoulderOffsetType, JEP30-D10:UnspecifiedValueSetType |

5.13.1.7.18.3 TH - Shoulder Cutout

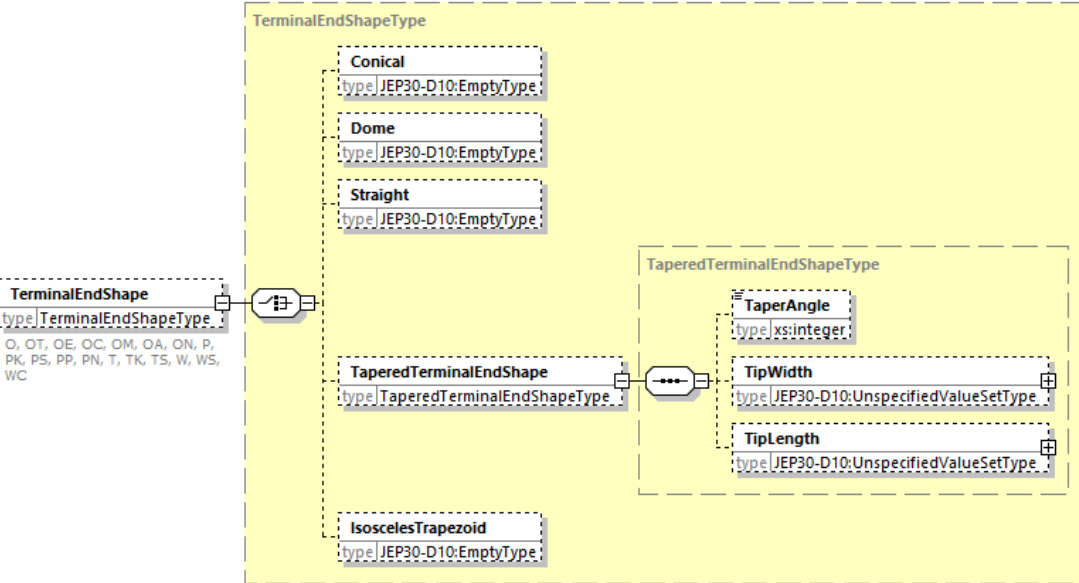
| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalShape/Shoulder/TH-ShoulderCutout |
| diagram | |
| type | TH-ShoulderCutoutType, CutoutConfigurationType, JEP30-D10:EmptyType, JEP30-D10:UnspecifiedValueSetType. |

5.13.1.7.19 Kinked

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalShape/Kinked |
| diagram |  |
| type | KinkedType, JEP30-D10:DimensionalValueSetType, |

The definition of each shape is outlined in Annex A (informative) Shape Definitions & Dimensions

5.13.1.7.20 Terminal End Shape

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalShape/TerminalEndShape |
| diagram |  |
| type | TerminalEndShapeType, TaperedTerminalEndShapeType, JEP30-D10:UnspecifiedValueSetType, JEP30-D10:EmptyType. |

5.13.1.7.21 Terminal Shape versus Dimensions

Table 21 — Terminal Shape versus Dimensions shows an array of dimensions that are captured below each shape listed above:

Table 21 - Terminal Shape versus Dimensions

| Terminal Shape | Dimension 1 | Dimension 2 | Dimension 3 | Diameter | Radius | Angle | No. of Sides | Inner/Outer | Shoulder | Impacted Corner |
|----------------------------------|-------------|-------------|-------------|----------|--------|-------|--------------|-------------|----------|-----------------|
| Rectangle | Y | Y | | | | Y | | | | |
| Rectangle with Shoulder | Y | Y | | | | Y | | | Y | |
| Rounded Rectangle | Y | Y | | | Y | Y | | | | |
| Modified Rectangle | Y | Y | | | | Y | | | | Y |
| Modified Rectangle with Shoulder | Y | Y | | | | Y | | | Y | Y |
| Circle | | | | Y | | | | | | |
| Circle with Shoulder | | | | Y | | | | | Y | |
| D-Shape | Y | Y | | | | Y | | | | |
| Double-D | Y | Y | | | | Y | | | | |
| Regular Polygon | | | | Y | | Y | Y | Y | | |
| Segment | Y | | | Y | | Y | | | | |
| Para-truncated Circle | Y | | | Y | | Y | | | | |
| Concave | Y | Y | | | Y | | | | | |
| Concave D-Shape | Y | Y | | | Y | | | | | |
| Isosceles Trapezoid | Y | Y | Y | | | Y | | | | |
| Contour | | | | | | Y | | | | |

The definition of each shape is outlined in Annex A (informative) Shape Dimensions.

5.13.1.7.22 Terminal Void - Array

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalShape/TerminalVoid-Array. |
| diagram | |
| type | TerminalVoid-ArrayType, TerminalVoidType, VoidlocationType, VoidStandardArrauType, VoidCircularArrayType, VoidStatusType, VoidRandomArrayType, TerminalVoidShapeType, RoundedRectangleType, CircleType, Double-DType, ContourShapeType. |

Some terminals such as Flat terminals and Surface terminals may have voids within the Terminal Contact Area, as shown in JESD30, "Illustrations of terminal shape" table. The example in Figure 4 is a Flat terminal with Hole, whereas the example in Figure 1 is a Surface terminal With-opening, plus a 2nd terminal called a Screw - Clearance Hole.

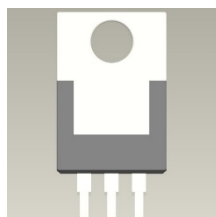


Figure 4 - Flat Terminal with Hole

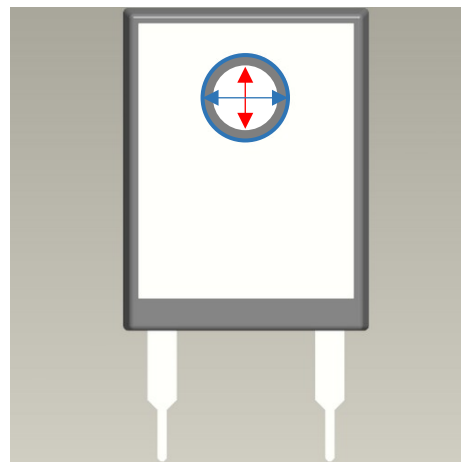


Figure 3 - Surface terminal With-Opening, plus Screw-Clearance Hole

5.13.1.6.22 Terminal Void – Array (cont'd)

When capturing the details of the Surface terminal With-opening, the dimensions associated with the larger diameter (as denoted by the horizontal arrow) is used to define the diameter of the circular opening within the Surface terminal. When the 2nd terminal group is defined to capture the definition of the Screw - Clearance Hole, the smaller diameter (as denoted by the vertical arrow) is used to define the diameter of the hole. The dimension then as calculated by the (larger diameter minus the smaller diameter) / 2 = the clearance between the edge of the hole and the metallization of the Surface terminal.

Since there may be more than one void within the Terminal shape shown in Figure 3, and since these voids may be different in shape (e.g., A Circle and a Double-D shape holes), the following sections outlines the array location of the voids and the different shapes that can be defined.



Figure 5 - Flat terminal with different shape Voids

The definition of each shape is outlined in Annex A (informative) Shape Dimensions.

5.13.1.7.22.1 Standard Array


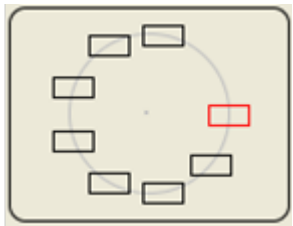
| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalShape/TerminalVoid-Array/TerminalVoid/TerminalVoidLocation/StandardArray. |
| diagram | |
| type | VoidStandardArrayType, PitchValueSetType, StandardType, MinMaxLimitsType, JEP30-D10:PointType. |

5.13.1.7.22.2 Circular Array

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalShape/TerminalVoid-Array/TerminalVoid/TerminalVoidLocation/CircularArray. |
| diagram | <p>The diagram illustrates the structure of the CircularArray class and its associated types. The CircularArray class is shown with a type of VoidCircularArrayType. It is connected to a dashed box labeled VoidCircularArrayType, which contains several attributes and a nested class.</p> <ul style="list-style-type: none">PitchRadius: type xs:doubleCenter: type PointTypeStartAngle: type xs:integerAngleToFill: type xs:integerAngleBetweenVoids: type xs:integerNumberOfVoids: type xs:integerRotation: type VoidRotationType <p>The VoidRotationType class is nested within the VoidCircularArrayType and contains two attributes:</p> <ul style="list-style-type: none">RotateWithCircle: type EmptyTypeRotationAngle: type xs:integer |
| type | VoidCircularArrayType, JEP30-D10:PointType, VoidRotationType. |

5.13.1.6.22.2 Circular Array (cont'd)

Table 22 - Void Circular Array Elements Definition

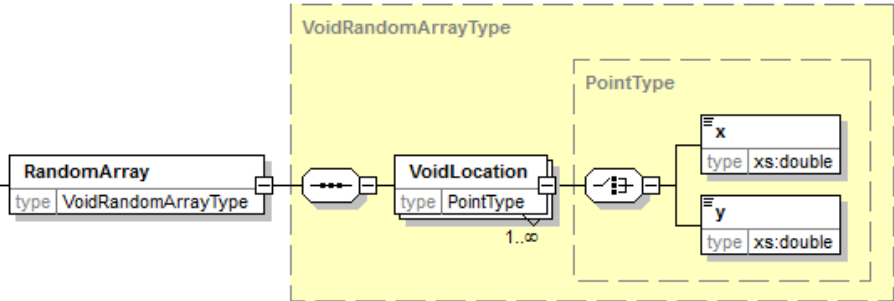
| Element | Explanation |
|---------------------|---|
| Pitch Radius | The radius of the circle of voids. |
| Center | The center of the circle of voids. |
| Start Angle | The angle at which to place the first void in the array. By default, the void is at the right-most point on the circle (i.e. the number 3 position on a clock face). This angle specifies a rotation clockwise around the circle from this point. |
| Angle to Fill | The angle through which the voids are distributed, starting from the first void and continuing clockwise, until the last void is reached. |
| Angle between voids | Instead of specifying Angle to fill, you can specify the angle between the centers of each void. |
| Number of voids | The number of voids to be arrayed around the circle. The first void is displayed in red. This number includes any deleted voids in the array. |
| Rotate with Circle |  <p>If the voids are rotated so that they are oriented perpendicular to the circle. This does not apply to circular or contour void shapes.</p> |
| Rotation Angle |  <p>If the voids are not oriented perpendicular to the circle, then the voids could be rotated around the center of the voids itself. In this image, each void has a "0" degree rotation with respect to the "3 O'clock position". This does not apply to circular or contour void shapes.</p> |

5.13.1.7.22.3 Void Status

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalShape/TerminalVoid-Array/TerminalVoid/TerminalVoidLocation/VoidStatus. |
| diagram | |
| type | VoidStatusType , JEP30-D10:PointType , JEP30-D10:MinIntegerOfOneType . |

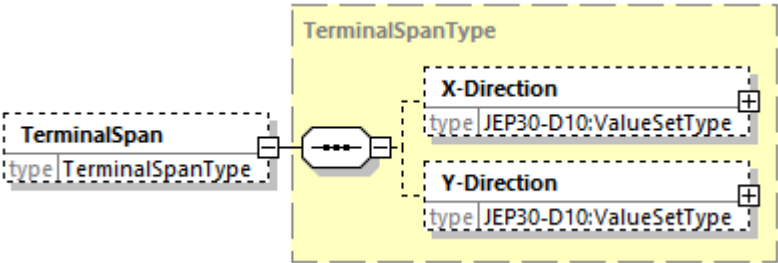
For [TerminalVoidLocation](#) defined via [StandardArray](#) or [CircularArray](#), some voids in the array may be deleted. This is covered in the [VoidStatus](#) branch.

5.13.1.7.22.4 Random Array

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalShape/TerminalVoid-Array/TerminalVoid/TerminalVoidLocation/RandomArray.. |
| diagram |  |
| type | VoidRandomArrayType, JEP30-D10:PointType. |

TerminalVoidLocation can also be defined via *RandomArray* when there is just 1 void, or when there is no logical structure to the location of the voids.

5.13.1.8 Terminal Span

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalSpan. |
| diagram |  |
| type | TerminalSpanType, JEP30-D10:ValueSetType. |

5.13.1.9 Terminal Spacing

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalSpacing. |
| diagram | |
| type | TerminalSpacingType , JEP30-D10:ValueSetType . |

[TerminalSpan](#) and [TerminalSpacing](#) values are critical in the calculation of the most efficient land patterns for several [Terminal](#) types. The requirements for the provision of the above data elements can be found in JESD30, under the “Terminal Span and Terminal Spacing” section which includes a table titled “Terminal Shapes requiring Terminal Span and Terminal Spacing dimensions”.

5.13.1.10 Terminal Group to Body Relationship

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalGroupToBodyRelationship |
| diagram | |
| type | TerminalGroupToBodyRelationshipType , TerminalGroupToBody-X-OffsetType , TerminalGroupToBody-Y-OffsetType , TerminalGroupToBody-Z-OffsetType . |

For non-symmetrical Terminals around the package center, dimensions may be referenced from the Terminal Contact Area to the Package body area. This data can be captured in the [TerminalGroupToBodyRelationship](#) section.

5.13.1.10.1 Terminal Group to Body X-Offset



5.13.1.9.1 Terminal Group to Body X-Offset (cont'd)

| | |
|------------------------------------|---|
| <p>diagram part 2 of 2</p> | |
| <p>type</p> | <p>TerminalGroupToBody-X-OffsetType, JEP30-D10:ValueSetType.</p> |

Table 23 — Terminal Group to Package Body X-Offset lists out all the various dimensions that can be selected from any major point of the package body to any point of the terminal contact area in X direction.

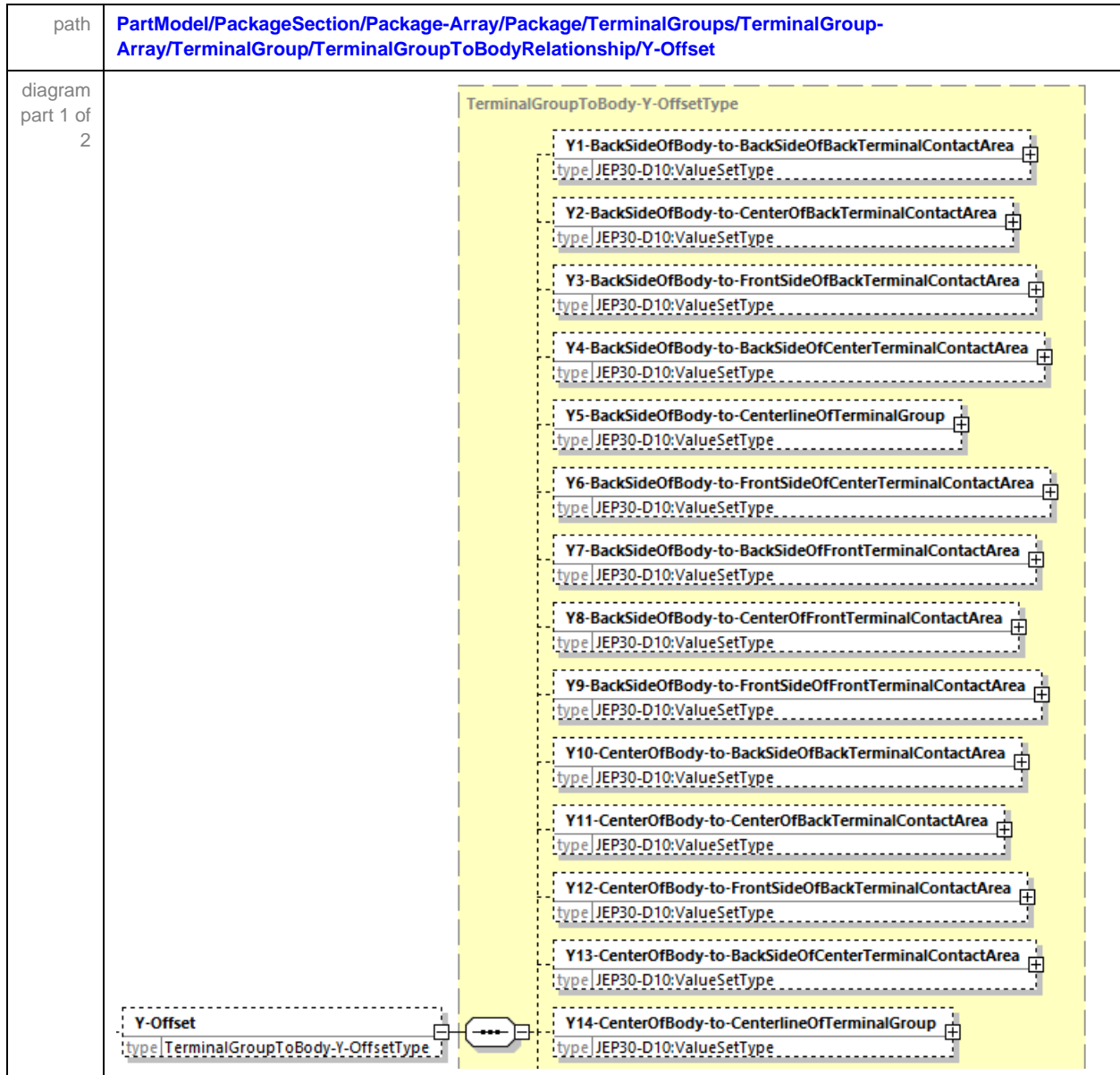
Center of Center Terminal Contact Area is the same as Centerline of Terminal Group, however the use of the "...Centerline of Terminal Group" is more encompassing since it also includes the centerline of a row or column of an even number of terminals, i.e., midway between terminal 2 and terminal 3 of 4 terminals in a row or column.

5.13.1.9.1 Terminal Group to Body X-Offset (cont'd)

Table 23 - Terminal Group to Package Body X-Offset

| Symbol | Description |
|--------|--|
| X1 | Left side of body to Left Side of Left Terminal Contact Area |
| X2 | Left side of body to Center of Left Terminal Contact Area |
| X3 | Left side of body to Right side of Left Terminal Contact Area |
| X4 | Left side of body to Left Side of Center Terminal Contact Area |
| X5 | Left side of body to Centerline of Terminal Group |
| X6 | Left side of body to Right side of Center Terminal Contact Area |
| X7 | Left side of body to Left side of Right Terminal Contact Area |
| X8 | Left side of body to Center of Right Terminal Contact Area |
| X9 | Left side of body to Right side of Right Terminal Contact Area |
| X10 | Center of body to Left side of Left Terminal Contact Area |
| X11 | Center of body to Center of Left Terminal Contact Area |
| X12 | Center of body to Right side of Left Terminal Contact Area |
| X13 | Center of body to Left side of Center Terminal Contact Area |
| X14 | Center of body to Centerline of Terminal Group |
| X15 | Center of body to Right side of Center Terminal Contact Area |
| X16 | Center of body to Left side of Right Terminal Contact Area |
| X17 | Center of body to Center of Right Terminal Contact Area |
| X18 | Center of body to Right side of Right Terminal Contact Area |
| X19 | Right side of body to Left side of Left Terminal Contact Area |
| X20 | Right side of body to Center of Left Terminal Contact Area |
| X21 | Right side of body to Right side of Left Terminal Contact Area |
| X22 | Right side of body to Left side of Center Terminal Contact Area |
| X23 | Right side of body to Centerline of Terminal Group |
| X24 | Right side of body to Right side of Center Terminal Contact Area |
| X25 | Right side of body to Left side of Right Terminal Contact Area |
| X26 | Right side of body to Center of Right Terminal Contact Area |
| X27 | Right side of body to Right side of Right Terminal Contact Area |

5.13.1.10.2 Terminal Group to Body Y-Offset



5.13.1.9.2 Terminal Group to Body Y-Offset (cont'd)

| | |
|---------------------------|---|
| diagram part 2 of 2 | <div><div><div>Y-Offset</div><div>type TerminalGroupToBody-Y-OffsetType</div></div><div><div>Y14-CenterOfBody-to-CenterlineOfTerminalGroup</div><div>type JEP30-D10:ValueSetType</div></div><div><div>Y15-CenterOfBody-to-FrontSideOfCenterTerminalContactArea</div><div>type JEP30-D10:ValueSetType</div></div><div><div>Y16-CenterOfBody-to-BackSideOfFrontTerminalContactArea</div><div>type JEP30-D10:ValueSetType</div></div><div><div>Y17-CenterOfBody-to-CenterOfFrontTerminalContactArea</div><div>type JEP30-D10:ValueSetType</div></div><div><div>Y18-CenterOfBody-to-FrontSideOfFrontTerminalContactArea</div><div>type JEP30-D10:ValueSetType</div></div><div><div>Y19-FrontSideOfBody-to-BackSideOfBackTerminalContactArea</div><div>type JEP30-D10:ValueSetType</div></div><div><div>Y20-FrontSideOfBody-to-CenterOfBackTerminalContactArea</div><div>type JEP30-D10:ValueSetType</div></div><div><div>Y21-FrontSideOfBody-to-FrontSideOfBackTerminalContactArea</div><div>type JEP30-D10:ValueSetType</div></div><div><div>Y22-FrontSideOfBody-to-BackSideOfCenterTerminalContactArea</div><div>type JEP30-D10:ValueSetType</div></div><div><div>Y23-FrontSideOfBody-to-CenterlineOfTerminalGroup</div><div>type JEP30-D10:ValueSetType</div></div><div><div>Y24-FrontSideOfBody-to-FrontSideOfCenterTerminalContactArea</div><div>type JEP30-D10:ValueSetType</div></div><div><div>Y25-FrontSideOfBody-to-BackSideOfFrontTerminalContactArea</div><div>type JEP30-D10:ValueSetType</div></div><div><div>Y26-FrontSideOfBody-to-CenterOfFrontTerminalContactArea</div><div>type JEP30-D10:ValueSetType</div></div><div><div>Y27-FrontSideOfBody-to-FrontSideOfFrontTerminalContactArea</div><div>type JEP30-D10:ValueSetType</div></div></div> |
| type | TerminalGroupToBody-Y-OffsetType, JEP30-D10:ValueSetType. |

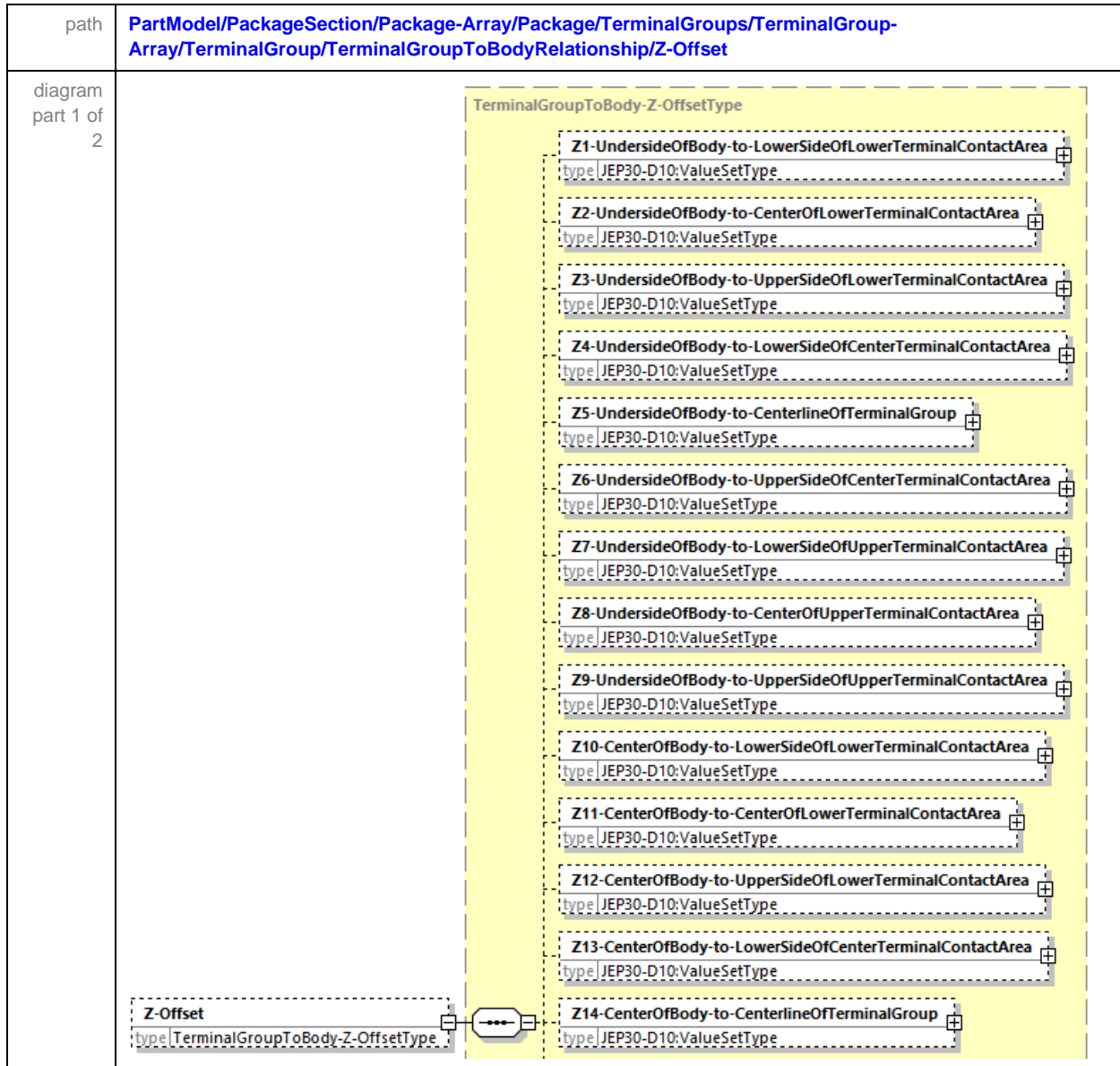
Table 24 — Terminal Group to Package Body Y-Offset lists out all the various dimensions that can be selected from any major point of the package body to any point of the terminal contact area in Y direction.

5.13.1.9.2 Terminal Group to Body Y-Offset (cont'd)

Table 24 - Terminal Group to Package Body Y-Offset

| Symbol | Description |
|--------|--|
| Y1 | Back side of body to Back side of Back Terminal Contact Area |
| Y2 | Back side of body to Center of Back Terminal Contact Area |
| Y3 | Back side of body to Front side of Back Terminal Contact Area |
| Y4 | Back side of body to Back side of Center Terminal Contact Area |
| Y5 | Back side of body to Centerline of Terminal Group |
| Y6 | Back side of body to Front side of Center Terminal Contact Area |
| Y7 | Back side of body to Back side of Front Terminal Contact Area |
| Y8 | Back side of body to Center of Front Terminal Contact Area |
| Y9 | Back side of body to Front side of Front Terminal Contact Area |
| Y10 | Center of body to Back side of Back Terminal Contact Area |
| Y11 | Center of body to Center of Back Terminal Contact Area |
| Y12 | Center of body to Front side of Back Terminal Contact Area |
| Y13 | Center of body to Back side of Center Terminal Contact Area |
| Y14 | Center of body to Centerline of Terminal Group |
| Y15 | Center of body to Front side of Center Terminal Contact Area |
| Y16 | Center of body to Back side of Front Terminal Contact Area |
| Y17 | Center of body to Center of Front Terminal Contact Area |
| Y18 | Center of body to Front side of Front Terminal Contact Area |
| Y19 | Front side of body to Back side of Back Terminal Contact Area |
| Y20 | Front side of body to Center of Back Terminal Contact Area |
| Y21 | Front side of body to Front side of Back Terminal Contact Area |
| Y22 | Front side of body to Back side of Center Terminal Contact Area |
| Y23 | Front side of body to Centerline of Terminal Group |
| Y24 | Front side of body to Front side of Center Terminal Contact Area |
| Y25 | Front side of body to Back side of Front Terminal Contact Area |
| Y26 | Front side of body to Center of Front Terminal Contact Area |
| Y27 | Front side of body to Front side of Front Terminal Contact Area |

5.13.1.10.3 Terminal Group to Body Z-Offset



5.13.1.9.3 Terminal Group to Body Z-Offset (cont'd)

| | |
|---------------------------|--|
| diagram part 2 of 2 | <div><div><div>Z-Offset</div><div>type TerminalGroupToBody-Z-OffsetType</div></div><div><div>Z14-CenterOfBody-to-CenterlineOfTerminalGroup</div><div>type JEP30-D10:ValueSetType</div></div><div><div>Z15-CenterOfBody-to-UpperSideOfCenterTerminalContactArea</div><div>type JEP30-D10:ValueSetType</div></div><div><div>Z16-CenterOfBody-to-LowerSideOfUpperTerminalContactArea</div><div>type JEP30-D10:ValueSetType</div></div><div><div>Z17-CenterOfBody-to-CenterOfUpperTerminalContactArea</div><div>type JEP30-D10:ValueSetType</div></div><div><div>Z18-CenterOfBody-to-UpperSideOfUpperTerminalContactArea</div><div>type JEP30-D10:ValueSetType</div></div><div><div>Z19-TopsideOfBody-to-LowerSideOfLowerTerminalContactArea</div><div>type JEP30-D10:ValueSetType</div></div><div><div>Z20-TopsideOfBody-to-CenterOfLowerTerminalContactArea</div><div>type JEP30-D10:ValueSetType</div></div><div><div>Z21-TopsideOfBody-to-UpperSideOfLowerTerminalContactArea</div><div>type JEP30-D10:ValueSetType</div></div><div><div>Z22-TopsideOfBody-to-LowerSideOfCenterTerminalContactArea</div><div>type JEP30-D10:ValueSetType</div></div><div><div>Z23-TopsideOfBody-to-CenterlineOfTerminalGroup</div><div>type JEP30-D10:ValueSetType</div></div><div><div>Z24-TopsideOfBody-to-UpperSideOfCenterTerminalContactArea</div><div>type JEP30-D10:ValueSetType</div></div><div><div>Z25-TopsideOfBody-to-LowerSideOfUpperTerminalContactArea</div><div>type JEP30-D10:ValueSetType</div></div><div><div>Z26-TopsideOfBody-to-CenterOfUpperTerminalContactArea</div><div>type JEP30-D10:ValueSetType</div></div><div><div>Z27-TopsideOfBody-to-UpperSideOfUpperTerminalContactArea</div><div>type JEP30-D10:ValueSetType</div></div></div> |
| type | TerminalGroupToBody-Z-OffsetType, JEP30-D10:ValueSetType. |

Table 25 — Terminal Group to Package Body Z-Offset lists out all the various dimensions that can be selected from any major point of the package body to any point of the terminal contact area in Z direction.

5.13.1.9.3 Terminal Group to Body Z-Offset (cont'd)

Table 25 - Terminal Group to Package Body Z-Offset

| Symbol | Description |
|--------|--|
| Z1 | Under side of Body to Lower side of Lower Terminal Contact Area |
| Z2 | Under side of Body to Center of Lower Terminal Contact Area |
| Z3 | Under side of Body to Upper side of Lower Terminal Contact Area |
| Z4 | Under side of Body to Lower side of Center Terminal Contact Area |
| Z5 | Under side of Body to Centerline of Terminal Group |
| Z6 | Under side of Body to Upper side of Center Terminal Contact Area |
| Z7 | Under side of Body to Lower side of Upper Terminal Contact Area |
| Z8 | Under side of Body to Center of Upper Terminal Contact Area |
| Z9 | Under side of Body to Upper side of Upper Terminal Contact Area |
| Z10 | Center of Body to Lower side of Lower Terminal Contact Area |
| Z11 | Center of Body to Center of Lower Terminal Contact Area |
| Z12 | Center of Body to Upper side of Lower Terminal Contact Area |
| Z13 | Center of Body to Lower side of Center Terminal Contact Area |
| Z14 | Center of Body to Centerline of Terminal Group |
| Z15 | Center of Body to Upper side of Center Terminal Contact Area |
| Z16 | Center of Body to Lower side of Upper Terminal Contact Area |
| Z17 | Center of Body to Center of Upper Terminal Contact Area |
| Z18 | Center of Body to Upper side of Upper Terminal Contact Area |
| Z19 | Upper side of Body to Lower side of Lower Terminal Contact Area |
| Z20 | Upper side of Body to Center of Lower Terminal Contact Area |
| Z21 | Upper side of Body to Upper side of Lower Terminal Contact Area |
| Z22 | Upper side of Body to Lower side of Center Terminal Contact Area |
| Z23 | Upper side of Body to Centerline of Terminal Group |
| Z24 | Upper side of Body to Upper side of Center Terminal Contact Area |
| Z25 | Upper side of Body to Lower side of Upper Terminal Contact Area |
| Z26 | Upper side of Body to Center of Upper Terminal Contact Area |
| Z27 | Upper side of Body to Upper side of Upper Terminal Contact Area |

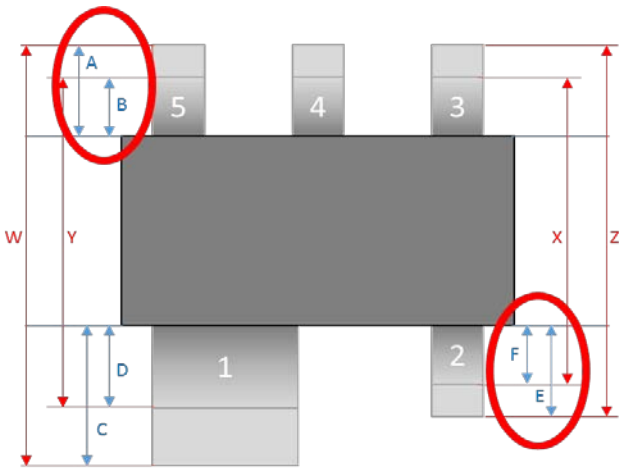
5.13.2 Terminal Group to Terminal Group Relationship Array

| | |
|------------------------|---|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroupToTerminalGroupRelationships-Array |
| diagram part 1 of 2 | |
| diagram part 2 of 2 | |
| type | TerminalGroupToTerminalGroupRelationships-ArrayType, TerminalGroupToTerminalGroupRelationship, TerminalSpanXYZType, TerminalSpacingXYZType. |

This section is very similar to Terminal Span and Spacing but in this case, it is capturing the Terminal Span and Spacing between Terminal Contact Area that are not symmetrical around the Package-body shape, or between terminal groups whose terminal types are different.

Figure 6 - Terminal Group to Terminal Group Span and Spacing

Assuming that the only dimensions provided for the Part in Figure 6, were the dimensions (w, x, y, and z), but not dimensions (a, b, c, d, e or f), then the only way to establish the location of Terminal 1 contact area is to define terminals 2, 3, 4 and 5 first, and then leveraging off the terminal location for terminal 5, using dimension w and y, the dimension w therefore becomes the Terminal Span between terminal 1 and terminal 5. Dimension y becomes the Terminal Spacing between terminal 1 and terminal 5. Terminal 1 must be located in different terminal groups to terminals 2-5.



5.13.2 Terminal Group to Terminal Group Relationship Array (cont'd)

Terminal Span and Spacing values are more critical than Terminal length in the calculation of the most efficient land patterns for several Terminal Shapes. The requirements for the provision of the above data elements can be found in JESD30, under the “Terminal Span and Terminal Spacing” section which includes a table titled “Terminal Shapes requiring Terminal Span and Terminal Spacing dimensions”.

5.13.3 Terminal Detail Array

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalDetail-Array |
| diagram | <pre> classDiagram class TerminalDetailArray { type TerminalDetailArrayType } class TerminalDetail { type TerminalDetailType } class TerminalNumberPattern { type TerminalNumberPatternType } TerminalDetailArray --> TerminalDetail TerminalDetailArray --> TerminalNumberPattern </pre> |
| type | TerminalDetail-ArrayType , TerminalDetailType , TerminalNumberPatternType . |

5.13.3.1 Terminal Detail

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalDetail-Array/ TerminalDetail |
| diagram | <p>The diagram illustrates the structure of the TerminalDetailType. It is a complex type containing the following elements:</p> <ul style="list-style-type: none">ID: type xs:stringTerminalNumber: type xs:stringTerminalCenter: type TerminalCenterTypeTerminalLocationID: type xs:stringPatternGroupID: type xs:stringTerminalGroupID: type xs:stringRowTerminalIndex: type JEP30-D10:MinIntegerOfOneType, minIncl/maxIncl 1ColumnTerminalIndex: type JEP30-D10:MinIntegerOfOneType, minIncl/maxIncl 1PolarTerminalIndex: type JEP30-D10:MinIntegerOfOneType, minIncl/maxIncl 1TerminalStatus: type TerminalStatusTypeIgnoreForLandPattern: type JEP30-D10:EmptyType <p>The TerminalDetail element is shown as a complex type derived from TerminalDetailType with a cardinality of 0..∞.</p> |
| type | TerminalDetailType, TerminalCenterType, JEP30-D10:MinIntegerOfOneType, TerminalStatusType, JEP30-D10:EmptyType. |

This section identifies the relationship between the electrical *TerminalNumber* and the physical terminal location on the package. The physical terminal location can be defined by either the identification of the *TerminalGroup*, *RowTerminalIndex* and the *ColumnTerminalIndex* within that specific *TerminalGroup*, or by a set of x,y coordinates. If the *TerminalLocation* is a *CircularArray*, then the physical terminal location can be defined by either the identification of the *TerminalGroup*, *PolarTerminalIndex* within that specific *TerminalGroup*, or by polar coordinates.

5.13.3.1 Terminal Detail (cont'd)

The row and column terminal index are simply a grid index of the terminal locations, within a specific [TerminalGroup](#) and starts off with the number 1 in the lower left position. The [RowTerminalIndex](#) increments from left-to-right, whereas the [ColumnTerminalIndex](#) increments from back-to-front. The [TerminalGroupID](#), [RowTerminalIndex](#) and the [ColumnTerminalIndex](#) combination provide a unique ID for every terminal on the package, irrespective of the status that is assigned to the specific terminal.

The [IgnoreForLandPattern](#) is when a specific terminal does not contribute to the generation of a land pattern.

5.13.3.1.1 Terminal Center

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalDetail-Array/TerminalDetail/TerminalCenter |
| diagram | <pre> classDiagram class TerminalCenter { type TerminalCenterType } class TerminalCenterType { x xs:decimal y xs:decimal PitchRadius xs:decimal Center JEP30-D10:PointXYType Angle xs:integer } class JEP30-D10:PointXYType { x xs:decimal y xs:decimal } TerminalCenter --> TerminalCenterType TerminalCenterType --> x1[x] TerminalCenterType --> y1[y] TerminalCenterType --> PitchRadius TerminalCenterType --> Center TerminalCenterType --> Angle Center --> x2[x] Center --> y2[y] </pre> |
| type | TerminalCenterType , JEP30-D10:PointType |

For [StandardArray](#) and for [RandomArray](#), xy coordinates can be used to capture the center of the terminals, whereas for [CircularArray](#), polar coordinates are a more convenient representation to capture the terminal centers.

5.13.3.1.1 Terminal Center (cont'd)

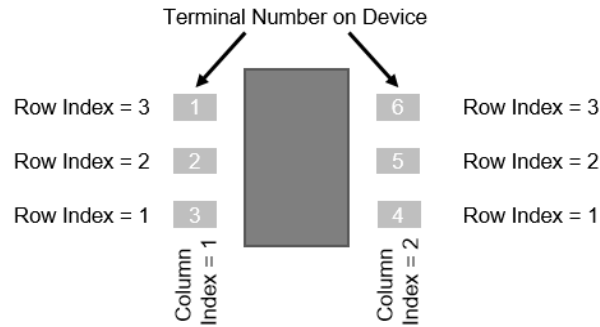


Figure 7 - SOIC

```

<TerminalDetail-Array>
  <TerminalDetail>
    <TerminalNumber>1</TerminalNumber>
    <TerminalGroupID>1</TerminalGroupID>
    <RowTerminalIndex>3</RowTerminalIndex>
    <ColumnTerminalIndex>1</ColumnTerminalIndex>
  </TerminalDetail>
  <TerminalDetail>
    <TerminalNumber>2</TerminalNumber>
    <TerminalGroupID>1</TerminalGroupID>
    <RowTerminalIndex>2</RowTerminalIndex>
    <ColumnTerminalIndex>1</ColumnTerminalIndex>
  </TerminalDetail>
  <TerminalDetail>
    <TerminalNumber>3</TerminalNumber>
    <TerminalGroupID>1</TerminalGroupID>
    <RowTerminalIndex>1</RowTerminalIndex>
    <ColumnTerminalIndex>1</ColumnTerminalIndex>
  </TerminalDetail>
  <TerminalDetail>
    <TerminalNumber>4</TerminalNumber>
    <TerminalGroupID>1</TerminalGroupID>
    <RowTerminalIndex>1</RowTerminalIndex>
    <ColumnTerminalIndex>2</ColumnTerminalIndex>
  </TerminalDetail>
  <TerminalDetail>
    <TerminalNumber>5</TerminalNumber>
    <TerminalGroupID>1</TerminalGroupID>
    <RowTerminalIndex>2</RowTerminalIndex>
    <ColumnTerminalIndex>2</ColumnTerminalIndex>
  </TerminalDetail>
  <TerminalDetail>
    <TerminalNumber>6</TerminalNumber>
    <TerminalGroupID>1</TerminalGroupID>
    <RowTerminalIndex>3</RowTerminalIndex>
    <ColumnTerminalIndex>2</ColumnTerminalIndex>
  </TerminalDetail>
</TerminalDetail-Array>

```

5.13.3.1.2 Terminal Status

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalDetail-Array/TerminalDetail/TerminalStatus |
| diagram | <p>The diagram illustrates the XSD structure for TerminalStatusType. It is a choice type with four possible values: Missing, Deleted, Excluded, and Associated. Each of these values is of type <code>JEP30-D10:EmptyType</code>. The Associated value is further detailed by the AssociatedTerminalStatusType, which contains a TerminalCenter (type <code>TerminalCenterType</code>) and a choice of several attributes: ReferenceTerminalLocationID (type <code>xs:string</code>), ReferencePatternGroupID (type <code>xs:string</code>), ReferenceTerminalGroupID (type <code>xs:string</code>), RowTerminalIndex (type <code>JEP30-D10:MinIntegerOfOneType</code>, minIncl=1, maxIncl=1), ColumnTerminalIndex (type <code>JEP30-D10:MinIntegerOfOneType</code>, minIncl=1, maxIncl=1), and PolarTerminalIndex (type <code>JEP30-D10:MinIntegerOfOneType</code>, minIncl=1, maxIncl=1).</p> |
| type | TerminalStatusType , JEP30-D10:EmptyType , AssociatedTerminalStatusType , TerminalCenterType , JEP30-D10:MinIntegerOfOneType . |

This section identifies the terminal status of each terminal. The status of *Missing*, *Deleted* and *Excluded* are defined in section called “Terminal-count suffixes” of JESD30.

The *TerminalDetail* and associated *TerminalStatus* will now look like this in the xml file.

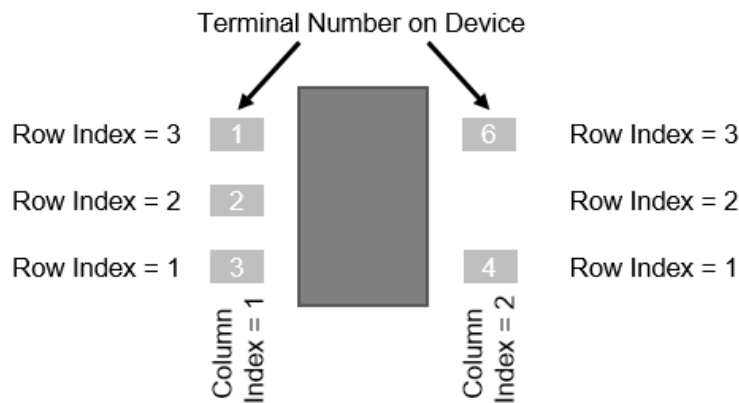


Figure 8 - SOIC with Terminal 5 Missing

5.13.3.1.2 Terminal Status (cont'd)

```
<TerminalDetail>
  <TerminalNumber>5</TerminalNumber>
  <TerminalGroupID>1</TerminalGroupID>
  <RowTerminalIndex>2</RowTerminalIndex>
  <ColumnTerminalIndex>2</ColumnTerminalIndex>
  <TerminalStatus>
    <Missing/>
  </TerminalStatus>
</TerminalDetail>
```

Note how *TerminalNumber* 5 is included in the xml file, and that the terminal at location of *RowTerminalIndex* = 3, and *ColumnTerminalIndex* = 2, that the *TerminalNumber* is 6.

When the Terminal is deleted, its index within the *TerminalGroup*, must still be defined. However, notice how the *TerminalNumber* element is not populated for this deleted position of *RowTerminalIndex* = 2, and *ColumnTerminalIndex* = 2. *TerminalNumber* 5 is now included in the xml file at location of *RowTerminalIndex* = 3, and *ColumnTerminalIndex* = 2.

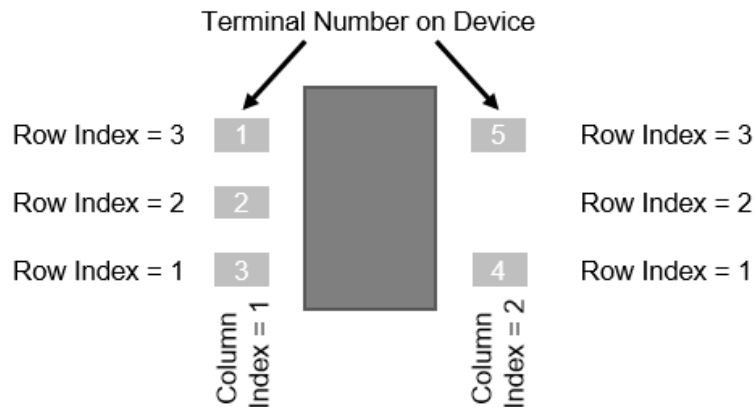


Figure 9 - SOIC with Terminal 5 Deleted

5.13.3.1.2 Terminal Status (cont'd)

```
<TerminalDetail-Array>
  <TerminalDetail>
    <TerminalNumber>1</TerminalNumber>
    <TerminalGroupID>1</TerminalGroupID>
    <RowTerminalIndex>3</RowTerminalIndex>
    <ColumnTerminalIndex>1</ColumnTerminalIndex>
  </TerminalDetail>
  <TerminalDetail>
    <TerminalNumber>2</TerminalNumber>
    <TerminalGroupID>1</TerminalGroupID>
    <RowTerminalIndex>2</RowTerminalIndex>
    <ColumnTerminalIndex>1</ColumnTerminalIndex>
  </TerminalDetail>
  <TerminalDetail>
    <TerminalNumber>3</TerminalNumber>
    <TerminalGroupID>1</TerminalGroupID>
    <RowTerminalIndex>1</RowTerminalIndex>
    <ColumnTerminalIndex>1</ColumnTerminalIndex>
  </TerminalDetail>
  <TerminalDetail>
    <TerminalNumber>4</TerminalNumber>
    <TerminalGroupID>1</TerminalGroupID>
    <RowTerminalIndex>1</RowTerminalIndex>
    <ColumnTerminalIndex>2</ColumnTerminalIndex>
  </TerminalDetail>
  <TerminalDetail>
    <TerminalGroupID>1</TerminalGroupID>
    <RowTerminalIndex>2</RowTerminalIndex>
    <ColumnTerminalIndex>2</ColumnTerminalIndex>
    <TerminalStatus>
      <Deleted/>
    </TerminalStatus>
  </TerminalDetail>
  <TerminalDetail>
    <TerminalNumber>5</TerminalNumber>
    <TerminalGroupID>1</TerminalGroupID>
    <RowTerminalIndex>3</RowTerminalIndex>
    <ColumnTerminalIndex>2</ColumnTerminalIndex>
  </TerminalDetail>
</TerminalDetail-Array>
```

5.13.3.1.2 Terminal Status (cont'd)

TerminalStatus of *Associated* occurs when you have parts with physically connected terminals and where there is a special requirement to be implemented in the land pattern for the proper performance of the device. The device may be constructed in accordance with either option A or option B as shown in Figure 7.

This device has two *TerminalGroups*. Lets assume that *TerminalGroup* 1 represents a 2 rows of 4 columns, and that *TerminalGroup* 2 represent the actual shape of terminal 1 on the device.

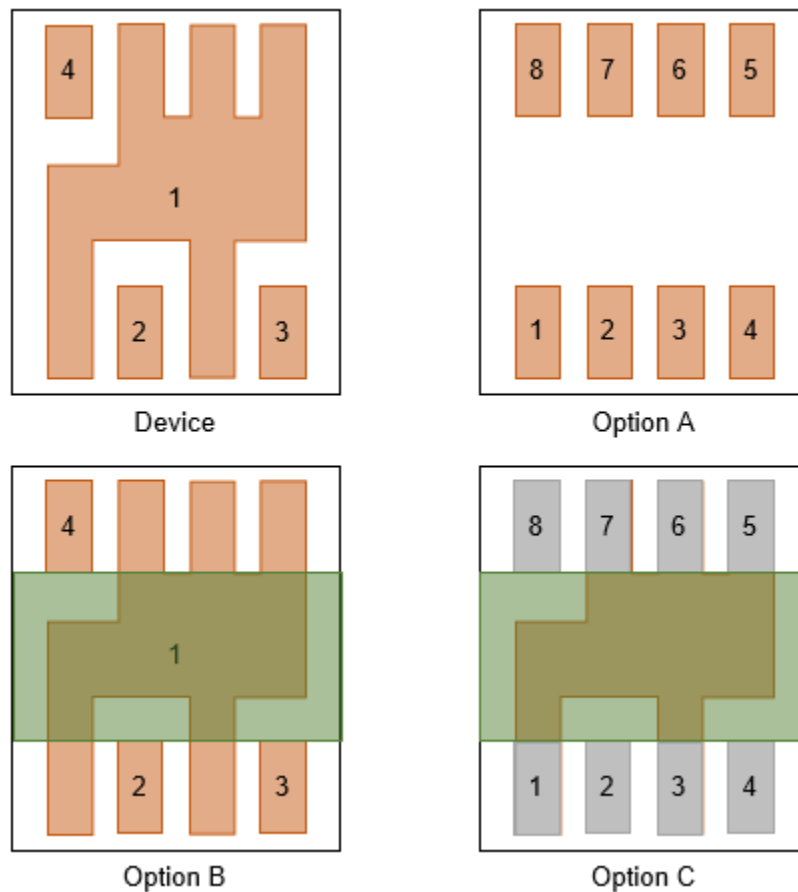


Figure 10 – Part with Special Terminal Pattern

5.13.3.1.2 Terminal Status (cont'd)

If option A land pattern is desired for the device, then the xml configuration should be as follows:

```
<TerminalDetail-Array>
  <TerminalDetail>
    <TerminalNumber>1</TerminalNumber>
    <TerminalGroupID>1</TerminalGroupID>
    <RowTerminalIndex>1</RowTerminalIndex>
    <ColumnTerminalIndex>1</ColumnTerminalIndex>
    <TerminalStatus>
      <Associated>
        <ReferenceTerminalGroupID>2</ReferenceTerminalGroupID>
      </Associated>
    </TerminalStatus>
  </TerminalDetail>
  <TerminalDetail>
    <TerminalNumber>2</TerminalNumber>
    <TerminalGroupID>1</TerminalGroupID>
    <RowTerminalIndex>1</RowTerminalIndex>
    <ColumnTerminalIndex>2</ColumnTerminalIndex>
  </TerminalDetail>
  <TerminalDetail>
    <TerminalNumber>3</TerminalNumber>
    <TerminalGroupID>1</TerminalGroupID>
    <RowTerminalIndex>1</RowTerminalIndex>
    <ColumnTerminalIndex>3</ColumnTerminalIndex>
    <TerminalStatus>
      <Associated>
        <ReferenceTerminalGroupID>2</ReferenceTerminalGroupID>
      </Associated>
    </TerminalStatus>
  </TerminalDetail>
  <TerminalDetail>
    <TerminalNumber>4</TerminalNumber>
    <TerminalGroupID>1</TerminalGroupID>
    <RowTerminalIndex>1</RowTerminalIndex>
    <ColumnTerminalIndex>4</ColumnTerminalIndex>
  </TerminalDetail>
  <TerminalDetail>
    <TerminalNumber>5</TerminalNumber>
    <TerminalGroupID>1</TerminalGroupID>
    <RowTerminalIndex>2</RowTerminalIndex>
    <ColumnTerminalIndex>4</ColumnTerminalIndex>
    <TerminalStatus>
      <Associated>
        <ReferenceTerminalGroupID>2</ReferenceTerminalGroupID>
      </Associated>
    </TerminalStatus>
  </TerminalDetail>
  <TerminalDetail>
    <TerminalNumber>6</TerminalNumber>
    <TerminalGroupID>1</TerminalGroupID>
    <RowTerminalIndex>2</RowTerminalIndex>
    <ColumnTerminalIndex>3</ColumnTerminalIndex>
    <TerminalStatus>
      <Associated>
        <ReferenceTerminalGroupID>2</ReferenceTerminalGroupID>
      </Associated>
    </TerminalStatus>
  </TerminalDetail>
</TerminalDetail-Array>
```


5.13.3.1.2 Terminal Status (cont'd)

```

<TerminalDetail>
  <TerminalNumber>7</TerminalNumber>
  <TerminalGroupID>1</TerminalGroupID>
  <RowTerminalIndex>2</RowTerminalIndex>
  <ColumnTerminalIndex>2</ColumnTerminalIndex>
  <TerminalStatus>
    <Associated>
      <ReferenceTerminalGroupID>2</ReferenceTerminalGroupID>
    </Associated>
  </TerminalStatus>
</TerminalDetail>
<TerminalDetail>
  <TerminalNumber>8</TerminalNumber>
  <TerminalGroupID>1</TerminalGroupID>
  <RowTerminalIndex>2</RowTerminalIndex>
  <ColumnTerminalIndex>1</ColumnTerminalIndex>
</TerminalDetail>
<TerminalDetail>
  <TerminalGroupID>2</TerminalGroupID>
  <RowTerminalIndex>1</RowTerminalIndex>
  <ColumnTerminalIndex>1</ColumnTerminalIndex>
  <IgnoreForLandPattern/>
</TerminalDetail>
</TerminalDetail-Array>

```

In this scenario, the following are enforced:-

1. Terminal group 2 is set to *IgnoreForLandPattern*,
2. Positions as shown by the number sequence in option A of 1, 3, 5, 6 and 7 are associated with terminal Group 2.
3. Since there is only 1 terminal in TerminalGroup 2, then there is no requirement to reference the *RowTerminalIndex* and the *ColumnTerminalIndex*. If the *RowTerminalIndex* and the *ColumnTerminalIndex* are omitted, then 1 is assumed.

5.13.3.1.2 Terminal Status (cont'd)

If option B land pattern is desired for the device, then the xml configuration should be as follows.

```
<TerminalDetail-Array>
  <TerminalDetail>
    <TerminalNumber>1</TerminalNumber>
    <TerminalGroupID>1</TerminalGroupID>
    <RowTerminalIndex>1</RowTerminalIndex>
    <ColumnTerminalIndex>1</ColumnTerminalIndex>
    <TerminalStatus>
      <Associated>
        <ReferenceTerminalGroupID>2</ReferenceTerminalGroupID>
      </Associated>
    </TerminalStatus>
    <IgnoreForLandPattern/>
  </TerminalDetail>
  <TerminalDetail>
    <TerminalNumber>2</TerminalNumber>
    <TerminalGroupID>1</TerminalGroupID>
    <RowTerminalIndex>1</RowTerminalIndex>
    <ColumnTerminalIndex>2</ColumnTerminalIndex>
  </TerminalDetail>
  <TerminalDetail>
    <TerminalNumber>1</TerminalNumber>
    <TerminalGroupID>1</TerminalGroupID>
    <RowTerminalIndex>1</RowTerminalIndex>
    <ColumnTerminalIndex>3</ColumnTerminalIndex>
    <TerminalStatus>
      <Associated>
        <ReferenceTerminalGroupID>2</ReferenceTerminalGroupID>
      </Associated>
    </TerminalStatus>
    <IgnoreForLandPattern/>
  </TerminalDetail>
  <TerminalDetail>
    <TerminalNumber>3</TerminalNumber>
    <TerminalGroupID>1</TerminalGroupID>
    <RowTerminalIndex>1</RowTerminalIndex>
    <ColumnTerminalIndex>4</ColumnTerminalIndex>
  </TerminalDetail>
  <TerminalDetail>
    <TerminalNumber>1</TerminalNumber>
    <TerminalGroupID>1</TerminalGroupID>
    <RowTerminalIndex>2</RowTerminalIndex>
    <ColumnTerminalIndex>4</ColumnTerminalIndex>
    <TerminalStatus>
      <Associated>
        <ReferenceTerminalGroupID>2</ReferenceTerminalGroupID>
      </Associated>
    </TerminalStatus>
    <IgnoreForLandPattern/>
  </TerminalDetail>
  <TerminalDetail>
    <TerminalNumber>1</TerminalNumber>
    <TerminalGroupID>1</TerminalGroupID>
    <RowTerminalIndex>2</RowTerminalIndex>
    <ColumnTerminalIndex>3</ColumnTerminalIndex>
    <TerminalStatus>
      <Associated>
        <ReferenceTerminalGroupID>2</ReferenceTerminalGroupID>
      </Associated>
    </TerminalStatus>
  </TerminalDetail>

```

5.13.3.1.2 Terminal Status (cont'd)

```

        </TerminalStatus>
        <IgnoreForLandPattern/>
    </TerminalDetail>
    <TerminalDetail>
        <TerminalNumber>1</TerminalNumber>
        <TerminalGroupID>1</TerminalGroupID>
        <RowTerminalIndex>2</RowTerminalIndex>
        <ColumnTerminalIndex>2</ColumnTerminalIndex>
        <TerminalStatus>
            <Associated>
                <ReferenceTerminalGroupID>2</ReferenceTerminalGroupID>
            </Associated>
        </TerminalStatus>
        <IgnoreForLandPattern/>
    </TerminalDetail>
    <TerminalDetail>
        <TerminalNumber>4</TerminalNumber>
        <TerminalGroupID>1</TerminalGroupID>
        <RowTerminalIndex>2</RowTerminalIndex>
        <ColumnTerminalIndex>1</ColumnTerminalIndex>
    </TerminalDetail>
    <TerminalDetail>
        <TerminalNumber>1</TerminalNumber>
        <TerminalGroupID>2</TerminalGroupID>
        <RowTerminalIndex>1</RowTerminalIndex>
        <ColumnTerminalIndex>1</ColumnTerminalIndex>
    </TerminalDetail>
</TerminalDetail-Array>

```

In this scenario, the following are enforced:-

1. Terminal group 2 is not set to *IgnoreForLandPattern* and therefore drives the shape of the land pattern on the printed board.
2. Positions as shown by the number sequence in option A of 1, 3, 5, 6 and 7 are associated with terminal Group 2, and each of these are set to *IgnoreForLandPattern*.

5.13.3.1.2 Terminal Status (cont'd)

If option C land pattern is desired for the device, then the xml configuration should be as follows.

```
<TerminalDetail-Array>
  <TerminalDetail>
    <TerminalNumber>1</TerminalNumber>
    <TerminalGroupID>1</TerminalGroupID>
    <RowTerminalIndex>1</RowTerminalIndex>
    <ColumnTerminalIndex>1</ColumnTerminalIndex>
    <TerminalStatus>
      <Associated>
        <ReferenceTerminalGroupID>2</ReferenceTerminalGroupID>
      </Associated>
    </TerminalStatus>
    <IgnoreForLandPattern/>
  </TerminalDetail>
  <TerminalDetail>
    <TerminalNumber>2</TerminalNumber>
    <TerminalGroupID>1</TerminalGroupID>
    <RowTerminalIndex>1</RowTerminalIndex>
    <ColumnTerminalIndex>2</ColumnTerminalIndex>
  </TerminalDetail>
  <TerminalDetail>
    <TerminalNumber>3</TerminalNumber>
    <TerminalGroupID>1</TerminalGroupID>
    <RowTerminalIndex>1</RowTerminalIndex>
    <ColumnTerminalIndex>3</ColumnTerminalIndex>
    <TerminalStatus>
      <Associated>
        <ReferenceTerminalGroupID>2</ReferenceTerminalGroupID>
      </Associated>
    </TerminalStatus>
    <IgnoreForLandPattern/>
  </TerminalDetail>
  <TerminalDetail>
    <TerminalNumber>4</TerminalNumber>
    <TerminalGroupID>1</TerminalGroupID>
    <RowTerminalIndex>1</RowTerminalIndex>
    <ColumnTerminalIndex>4</ColumnTerminalIndex>
  </TerminalDetail>
  <TerminalDetail>
    <TerminalNumber>5</TerminalNumber>
    <TerminalGroupID>1</TerminalGroupID>
    <RowTerminalIndex>2</RowTerminalIndex>
    <ColumnTerminalIndex>4</ColumnTerminalIndex>
    <TerminalStatus>
      <Associated>
        <ReferenceTerminalGroupID>2</ReferenceTerminalGroupID>
      </Associated>
    </TerminalStatus>
    <IgnoreForLandPattern/>
  </TerminalDetail>
  <TerminalDetail>
    <TerminalNumber>6</TerminalNumber>
    <TerminalGroupID>1</TerminalGroupID>
    <RowTerminalIndex>2</RowTerminalIndex>
    <ColumnTerminalIndex>3</ColumnTerminalIndex>
    <TerminalStatus>
      <Associated>
        <ReferenceTerminalGroupID>2</ReferenceTerminalGroupID>
      </Associated>
    </TerminalStatus>
  </TerminalDetail>
</TerminalDetail-Array>
```

5.13.3.1.2 Terminal Status (cont'd)

```

        </TerminalStatus>
        <IgnoreForLandPattern/>
    </TerminalDetail>
    <TerminalDetail>
        <TerminalNumber>7</TerminalNumber>
        <TerminalGroupID>1</TerminalGroupID>
        <RowTerminalIndex>2</RowTerminalIndex>
        <ColumnTerminalIndex>2</ColumnTerminalIndex>
        <TerminalStatus>
            <Associated>
                <ReferenceTerminalGroupID>2</ReferenceTerminalGroupID>
            </Associated>
        </TerminalStatus>
        <IgnoreForLandPattern/>
    </TerminalDetail>
    <TerminalDetail>
        <TerminalNumber>8</TerminalNumber>
        <TerminalGroupID>1</TerminalGroupID>
        <RowTerminalIndex>2</RowTerminalIndex>
        <ColumnTerminalIndex>1</ColumnTerminalIndex>
    </TerminalDetail>
    <TerminalDetail>
        <TerminalNumber>1</TerminalNumber>
        <TerminalGroupID>2</TerminalGroupID>
        <RowTerminalIndex>1</RowTerminalIndex>
        <ColumnTerminalIndex>1</ColumnTerminalIndex>
    </TerminalDetail>
</TerminalDetail-Array>

```

In this scenario, the following are enforced:-

1. Terminal group 2 is not set to *IgnoreForLandPattern* and therefore drives the shape of the land pattern on the printed board.
2. Positions as shown by the number sequence in option A of 1, 3, 5, 6 and 7 are associated with terminal Group 2, and each of these are set to *IgnoreForLandPattern*.

5.13.3.2 Terminal Number Pattern

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalDetail-Array/TerminalNumberPattern |
| diagram | <p>The diagram illustrates the structure of the TerminalNumberPatternType. It is a sequence of elements: FirstTerminalLocation (type FirstTerminalLocationType), Prefix (type xs:string), and Suffix (type xs:string). This is followed by a choice between Sequential (type SequentialTerminalNumberOrderingType) and Grid (type GridTerminalNumberOrderingType). Finally, there is a sequence of TerminalDetailExceptions (type TerminalDetailExceptionsType). The entire structure is enclosed in a dashed box labeled TerminalNumberPatternType. The diagram also shows the TerminalNumberPattern element (type TerminalNumberPatternType) with a cardinality of 0..∞.</p> |
| type | TerminalNumberPatternType, FirstTerminalLocationType, SequentialTerminalNumberOrderingType, GridTerminalNumberOrderingType, TerminalDetailExceptionsType. |

If the terminal numbering on a Part has logical structure to its pattern, then this can be captured in this [TerminalNumberPattern](#) section. The section covers the capturing of the [FirstTerminalLocation](#) which is the start of the pattern numbering via the [Sequential](#) or [Grid](#) branches. Any terminals that are either deleted, missing or associated within the pattern are then covered via the [TerminalDetailExceptions](#).

5.13.3.2.1 First Terminal Location

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalDetail-Array/TerminalNumberPattern/FirstTerminalLocation |
| diagram | <p>The diagram illustrates the structure of the <code>FirstTerminalLocationType</code>. It is a sequence of the following elements:</p> <ul style="list-style-type: none"> <code>TerminalPatternID</code> (type: <code>xs:string</code>) <code>PatternGroupID</code> (type: <code>xs:string</code>) <code>TerminalGroupID</code> (type: <code>xs:string</code>) <code>TerminalCenter</code> (type: <code>TerminalCenterType</code>) A choice of three index types, each with <code>minIncl/maxIncl</code> of 1: <ul style="list-style-type: none"> <code>RowTerminalIndex</code> (type: <code>JEP30-D10:MinIntegerOfOneType</code>) <code>ColumnTerminalIndex</code> (type: <code>JEP30-D10:MinIntegerOfOneType</code>) <code>PolarTerminalIndex</code> (type: <code>JEP30-D10:MinIntegerOfOneType</code>) <code>LocationRelative-to-PackageCenter</code> (type: <code>LocationRelative-to-PackageCenterType</code>), which contains a choice of: <ul style="list-style-type: none"> <code>Code</code> (type: <code>LocationRelative-to-PackageCenterCodeType</code>) <code>Description</code> (type: <code>LocationRelative-to-PackageCenterDescriptionType</code>) |
| type | FirstTerminalLocationType , TerminalCenterType , JEP30-D10:MinIntegerOfOneType , LocationRelative-to-PackageCenterType , LocationRelative-to-PackageCenterCodeType , LocationRelative-to-PackageCenterDescriptionType . |

The enumerated values for the [LocationRelative-to-PackageCenter](#) (Code and Description) are defined in Table 26.

5.13.3.2.1 First Terminal Location (cont'd)

Table 26 - Location Relative to Package Center

| Code | Description |
|------|--------------|
| SW | Southwest |
| SE | Southeast |
| NE | Northeast |
| NW | Northwest |
| BL | Back-Left |
| BC | Back-Center |
| BR | Back-Right |
| FL | Front-Left |
| FC | Front-Center |
| FR | Front-Right |
| LB | Left-Bottom |
| LC | Left-Center |
| LT | Left-Top |
| RB | Right-Bottom |
| RC | Right-Center |
| RT | Right-Top |
| L | Left |
| B | Back |
| R | Right |
| F | Front |

Reference the “Single Position” single terminal illustrations in JESD30, Annex A for graphical representations of some of the above descriptions.

5.13.3.2.2 Sequential

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalDetail-Array/TerminalNumberPattern/Sequential |
| diagram | |
| type | SequentialTerminalNumberOrderingType , TerminalNumberOrderingCodeType , TerminalNumberOrderingDescriptionType , NumericalSequenceType , AlphabeticalSequenceType . |

The enumerated values for the *Sequential* (*Code* and *Description*) are defined in Table 27.

Table 27 - Terminal Number Sequential Pattern

| Code | Description |
|------|--------------------|
| CW | Clockwise |
| CCW | Counter-Clockwise |
| L2R | Left-to-Right |
| B2F | Back-to-Front |
| R2L | Right-to-Left |
| F2B | Front-to-Back |
| S-H | Snake Horizontal |
| S-V | Snake Vertical |
| ZZ-H | Zig-Zag Horizontal |
| ZZ-V | Zig-Zag Vertical |










5.13.3.2.2 Sequential (cont'd)

Typically, the default sequence is a running number starting with the digit 1 and incrementing by 1. If this is the case, then the [NumericalSequence](#) or the [AlphabeticalSequence](#) do not have to be populated. However, in some cases, the Numerical or alphabetical sequence can override the default, in which case the sequence of terminal numbers is applied in the same order as the sequence pattern defined by the Code or Description as outlined in Table 27.

Table 28 - Terminal Number Patterns

| Pattern | Terminal Numbering Pattern | Sequential Code | Start Position | Towards Point 2 | Towards Point 3 | Towards Point 4 | Towards Point 5 | Last Point |
|---------|---|-----------------|----------------|-----------------|-----------------|-----------------|-----------------|-------------------------|
| 1 |  | CW | User Defined | Start + 1 CW | Start + 2 CW | Start + 3 CW | Start + 4 CW | Start - 1 CW |
| 2 |  | CCW | User Defined | Start + 1 CCW | Start + 2 CCW | Start + 3 CCW | Start + 4 CCW | Start - 1 CCW |
| 3 |  | ZZ-V | BL | TL | BL+1 | TL+1 | BL+2 | TR |
| 4 |  | ZZ-V | BR | TR | BR-1 | TR-1 | BR-2 | TL |
| 5 |  | ZZ-V | TR | BR | TR-1 | BR-1 | TR-2 | BL |
| 6 |  | ZZ-V | TL | BL | TL+1 | BL+1 | TL+2 | BR |
| 7 |  | ZZ-H | TL | TR | T-1L | T-1R | T-2L | BR |
| 8 |  | ZZ-H | BL | BR | B+1L | B+1R | B+2L | TR |
| 9 |  | ZZ-H | BR | BL | B+1R | B+1L | B+2R | TL |
| 10 |  | ZZ-H | TR | TL | T-1R | T-1L | T-2R | BL |
| 11 |  | S-V | TL | BL | BL+1 | TL+1 | TL+2 | IF(Col # = odd, BR, TR) |
| 12 |  | S-V | BL | TL | TL+1 | BL+1 | BL+2 | IF(Col # = odd, TR, BR) |
| 13 |  | S-V | TR | BR | BR-1 | TR-1 | TR-2 | IF(Col # = odd, BL, TL) |
| 14 |  | S-V | BR | TR | TR-1 | BR-1 | BR-2 | IF(Col # = odd, TL, BL) |

5.13.3.2.2 Sequential (cont'd)

| Pattern | Terminal Numbering Pattern | Sequential Code | Start Position | Towards Point 2 | Towards Point 3 | Towards Point 4 | Towards Point 5 | Last Point |
|---------|---|-----------------|----------------|-----------------|-----------------|-----------------|-----------------|-------------------------|
| 15 |  | S-H | TL | TR | T-1R | T-1L | T-2L | IF(Col # = odd, BR, BL) |
| 16 |  | S-H | BL | BR | B+1R | B+1L | B+2L | IF(Col # = odd, TR, TL) |
| 17 |  | S-H | TR | TL | R-1L | T-1R | T-2R | IF(Col # = odd, BL, BR) |
| 18 |  | S-H | BR | BL | B+1L | B+1R | B+2R | IF(Col # = odd, TL, TR) |
| 19 |  | L2R | L | L+1 | L+2 | L+3 | L+4 | R |
| 20 |  | R2L | R | R-1 | R-2 | R-3 | R-4 | L |
| 21 |  | B2F | B | B+1 | B+2 | B+3 | B+4 | T |
| 22 |  | F2B | T | T-1 | T-2 | T-3 | T-4 | B |
| 23 |  | Assign | User Defined | N/A | N/A | N/A | N/A | N/A |

By combining the start position of the terminal number via the [LocationRelative-to-PackageCenter](#) with the [Sequential](#) Pattern, we can achieve the patterns outlined in Table 28 — Terminal Number Patterns for terminal numbering.

For positions of BL, TL, BR, TR, the following is also valid for terminals positioned outside of the package body, namely.

1. The position of BL (Bottom Left) corresponds to
 - (a) For terminals with a position of Left, Right or Left-to-Right of the package body, then Left-bottom (LB),
 - (b) For terminals with a position of Back, Front or Back-to-Front of the package body, then Back-left (BL),
 - (c) For terminals with a position of Diagonal of the package body, then Southwest (SW),
2. The position of TL (Top Left) corresponds to
 - (a) For terminals with a position of Left, Right or Left-to-Right of the package body, then Left-top (LT),
 - (b) For terminals with a position of Back, Front or Back-to-Front of the package body, then Front-left (FL),

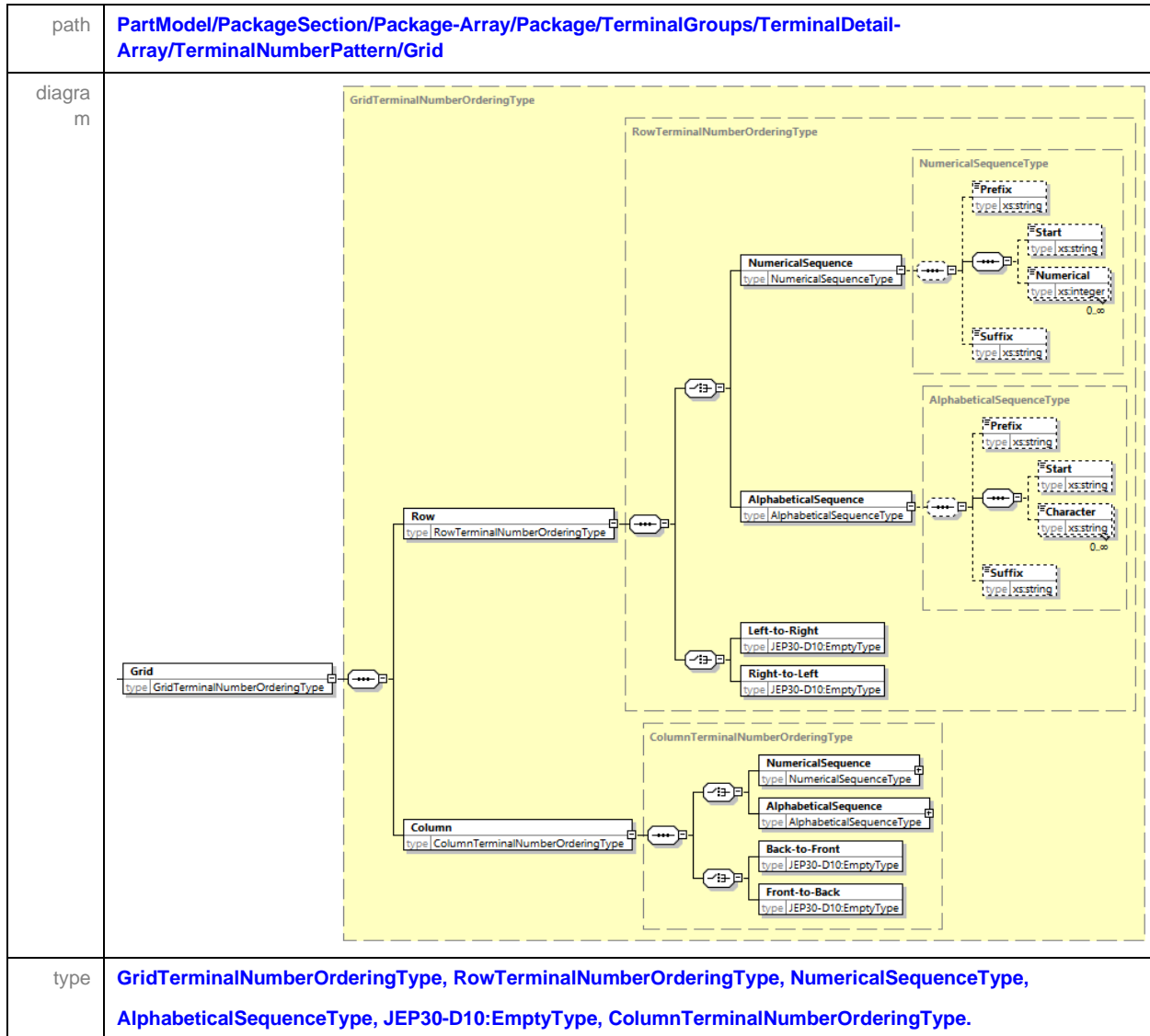
5.13.3.2.2 Sequential (cont'd)

- (c) For terminals with a position of Diagonal of the package body, then Northwest (NW),
- 3. The position of BR (Bottom Right) corresponds to
 - (a) For terminals with a position of Left, Right or Left-to-Right of the package body, then Right-bottom (RB),
 - (b) For terminals with a position of Back, Front or Back-to-Front of the package body, then Back-right (BR),
 - (c) For terminals with a position of Diagonal of the package body, then Southeast (SE),
- 4. The position of TR (Top Right) corresponds to
 - (a) For terminals with a position of Left, Right or Left-to-Right of the package body, then Right-top (RT),
 - (b) For terminals with a position of Back, Front or Back-to-Front of the package body, then Front-right (FR),
 - (c) For terminals with a position of Diagonal of the package body, then Northeast (NE).



Figure 11 – Terminal Numbering Pattern Samples

5.13.3.2.3 Grid



This section identifies the terminal numbering of each terminal in a grid array by providing both a *Row* and *Column* identification of the terminal location on the *Grid*. Each Row and Column can have either a *NumericalSequence* or an *AlphabeticalSequence*. The *NumericalSequence* is a running number beginning with the digit 1 and incrementing by 1, however that sequence can be over-ridden by the provision of a sequence of numbers defined in the *Numerical* element.

Similarly, the *AlphabeticalSequence* is a running number sequence of the letters of the English alphabet, however in the majority of cases, the characters I, O, Q, S, X, and Z are sometimes not used because it is hard to distinguish these either legibly or from their numerical equivalency from the legend. As such the Character element under the *AlphabeticalSequence* is an unbounded element that contains the used set of characters in an ordered sequence.

The *Prefix* is an optional integer that can be placed before the character to define the terminal numbering of a multi grid package, as shown in Figure 12 – A Multi grid Package.

5.13.3.2.3 Grid (cont'd)

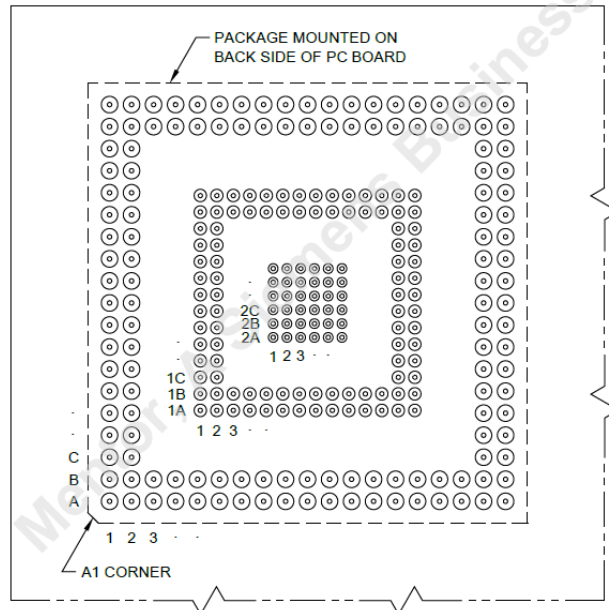
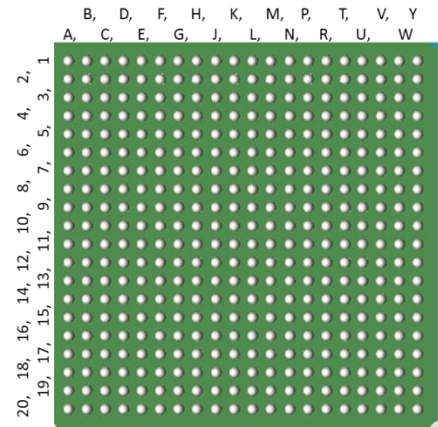


Figure 12 – A Multi grid Package

In Figure 13, the columns have a *NumericalSequence* from *Front-to-Back*. The rows have an *AlphabeticalSequence* going from *Left-to-Right*, with a specific set of alphabetical set of *Characters*. The xml structure below represents the detail terminal numbering in this example.

Figure 13 – Grid Array with Terminal Numbering



```
<TerminalDetail-Array>
  <TerminalNumberPattern>
    <FirstTerminalLocation>
      <TerminalGroupID> Terminal Group ID 1 </TerminalGroupID>
      <LocationRelative-to-PackageCenter>
        <Code>SW</Code>
      </LocationRelative-to-PackageCenter>
    </FirstTerminalLocation>
  </TerminalNumberPattern>
  <Grid>
    <Row>
      <AlphabeticalSequence>
        <Character>A</Character>
        <Character>B</Character>
        <Character>C</Character>
        <Character>D</Character>
        <Character>E</Character>
        <Character>F</Character>
        <Character>G</Character>
        <Character>H</Character>
        <Character>J</Character>
        <Character>K</Character>
        <Character>L</Character>
      </AlphabeticalSequence>
    </Row>
  </Grid>
</TerminalDetail-Array>
```

5.13.3.2.3 Grid (cont'd)

```

<Character>M</Character>
<Character>N</Character>
<Character>P</Character>
<Character>R</Character>
<Character>T</Character>
<Character>U</Character>
<Character>V</Character>
<Character>W</Character>
<Character>Y</Character>
</AlphabeticalSequence>
<Left-to-Right/>
</Row>
<Column>
<NumericalSequence></NumericalSequence>
<Front-to-Back/>
</Column>
</Grid>
</TerminalNumberPattern>
</TerminalDetail-Array>

```

Table 29 – Grid Terminal Numbering Pattern


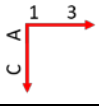
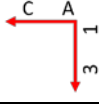
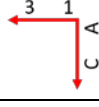

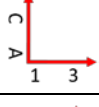

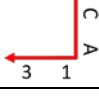
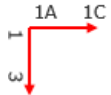
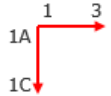

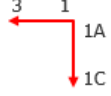
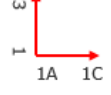

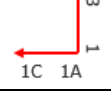

| Grid | Terminal Grid Pattern | First Terminal Location | Numerical Sequence | Alphabetical Sequence |
|------|---|-------------------------|--------------------|-----------------------|
| 1 |  | NW | Front-to-Back | Left-to-Right |
| 2 |  | NW | Left-to-Right | Front-to-Back |
| 3 |  | NE | Front-to-Back | Right-to-Left |
| 4 |  | NE | Right-to-Left | Front-to-Back |
| 5 |  | SW | Back-to-Front | Left-to-Right |
| 6 |  | SW | Left-to-Right | Back-to-Front |
| 7 |  | SE | Back-to-Front | Right-to-Left |
| 8 |  | SE | Right-to-Left | Back-to-Front |

Table 29 - Grid Terminal Number Patterns (cont'd).

| Grid | Terminal Grid Pattern | First Terminal Location | Numerical Sequence | Alphabetical Sequence |
|------|---|-------------------------|--------------------|-----------------------|
| 9 |  | NW | Front-to-Back | Left-to-Right |
| 10 |  | NW | Left-to-Right | Front-to-Back |
| 11 |  | NE | Front-to-Back | Right-to-Left |
| 12 |  | NE | Right-to-Left | Front-to-Back |
| 13 |  | SW | Back-to-Front | Left-to-Right |
| 14 |  | SW | Left-to-Right | Back-to-Front |
| 15 |  | SE | Back-to-Front | Right-to-Left |
| 15 |  | SE | Right-to-Left | Back-to-Front |

5.13.3.2.4 Terminal Detail Exceptions

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalDetail-Array/TerminalNumberPattern/TerminalDetailExceptions |
| diagram | |
| type | TerminalDetailExceptionsType, TerminalCenterType, TerminalIndexType, TerminalNumberType, TerminalStatusType, JEP30-D10:EmptyType, AssociatedTerminalStatusType. |

5.13.3.2.4.1 Terminal Index

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalDetail-Array/TerminalNumberPattern/TerminalDetailExceptions/TerminalIndex |
| diagram | <p>The diagram illustrates the structure of the TerminalIndex element, which is of type TerminalIndexType. It is composed of three main parts:</p> <ul style="list-style-type: none">TerminalLocationID (type <code>xs:string</code>): A single required element.Group ID Choice: A choice between PatternGroupID and TerminalGroupID, both of type <code>xs:string</code>. This choice is required and has a cardinality of <code>1..∞</code>.Index Type Choice: A choice between RowTerminalIndex, ColumnTerminalIndex, and PolarTerminalIndex. Each of these is required and has a cardinality of <code>1..∞</code>.<ul style="list-style-type: none">RowTerminalIndex: Contains a choice between FromRowTerminalIndex and ToRowTerminalIndex, both of type <code>JEP30-D10:MinIntegerOfOneType</code> and cardinality <code>1</code>.ColumnTerminalIndex: Contains a choice between FromColumnTerminalIndex and ToColumnTerminalIndex, both of type <code>JEP30-D10:MinIntegerOfOneType</code> and cardinality <code>1</code>.PolarTerminalIndex: Contains a choice between FromPolarTerminalIndex and ToPolarTerminalIndex, both of type <code>JEP30-D10:MinIntegerOfOneType</code> and cardinality <code>1</code>. |
| type | TerminalIndexType, JEP30-D10:MinIntegerOfOneType. |

5.13.3.2.4.2 Terminal Number

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalDetail-Array/TerminalNumberPattern/TerminalDetailExceptions/TerminalNumberType |
| diagram | <pre>graph LR TN[TerminalNumber type TerminalNumberType] -- 1..∞ --> C1(()) C1 -- 1..∞ --> N[Number type xs:string] C1 -- 1..∞ --> R[Row type xs:string] C1 -- 1..∞ --> Col[Column type xs:string] N -- 1..∞ --> FN[FromNumber type xs:string] N -- 1..∞ --> TNr[ToNumber type xs:string] R -- 1..∞ --> FR[FromRow type xs:string] R -- 1..∞ --> TR[ToRow type xs:string] Col -- 1..∞ --> FC[FromColumn type xs:string] Col -- 1..∞ --> TC[ToColumn type xs:string]</pre> |
| type | TerminalNumberType |

If a pattern of terminals are missing from the device, then these missing patterns can be captured under the [TerminalDetailExceptions](#) section as shown below for the Peripheral Grid Array shown in Figure 12.

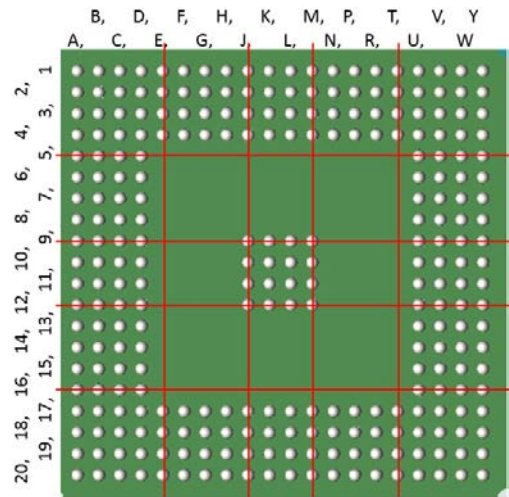


Figure 14 – Periphery Grid Array with Inner Array Matrix

5.13.3.2.4.2 Terminal Number (cont'd)

```
<TerminalDetailExceptions>
  <TerminalNumber>
    <FromRow>5</FromRow>
    <ToRow>16</ToRow>
    <FromColumn>E</FromColumn>
    <ToColumn>H</ToColumn>
  </TerminalNumber>
  <TerminalStatus>
    <Missing></Missing>
  </TerminalStatus>
</TerminalDetailExceptions>
<TerminalDetailExceptions>
  <TerminalNumber>
    <FromRow>5</FromRow>
    <ToRow>16</ToRow>
    <FromColumn>N</FromColumn>
    <ToColumn>T</ToColumn>
  </TerminalNumber>
  <TerminalStatus>
    <Missing></Missing>
  </TerminalStatus>
</TerminalDetailExceptions>
<TerminalDetailExceptions>
  <TerminalNumber>
    <FromRow>5</FromRow>
    <ToRow>8</ToRow>
    <FromColumn>J</FromColumn>
    <ToColumn>M</ToColumn>
  </TerminalNumber>
  <TerminalStatus>
    <Missing></Missing>
  </TerminalStatus>
</TerminalDetailExceptions>
<TerminalDetailExceptions>
  <TerminalNumber>
    <FromRow>13</FromRow>
    <ToRow>16</ToRow>
    <FromColumn>J</FromColumn>
    <ToColumn>M</ToColumn>
  </TerminalNumber>
  <TerminalStatus>
    <Missing></Missing>
  </TerminalStatus>
</TerminalDetailExceptions>
```

5.13.4 Via Array

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/Via-Array |
| diagram | |
| type | Via-ArrayType, ViaLocationType, ViaStandardArrayType, ViaCircularArrayType, ViaRandomArrayType, ViaDetailType. |

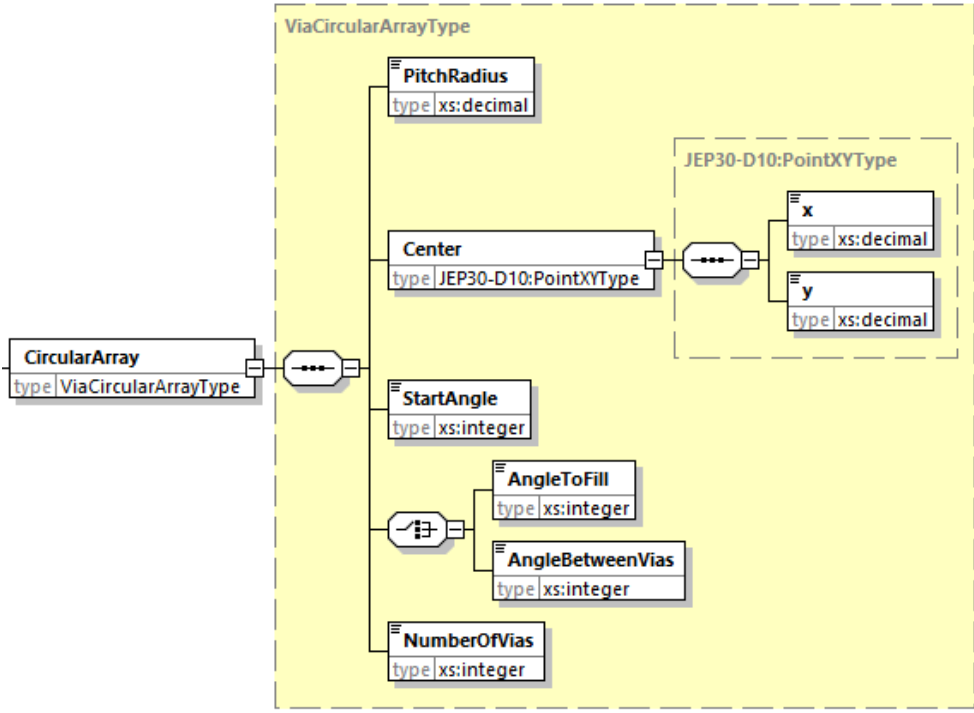
Grounding Vias are sometimes required to support the operational functionality of the Part in a product design. This [Via-Array](#) section enables this detail to be captured. The location and pattern of these Vias can be captured via any combination of a [StandardArray](#), [CircularArray](#) or [RandomArray](#) as shown in the diagram..

The dimensions and technology of the vias are not defined since this is determined by the Product on to which the Part is assembled.

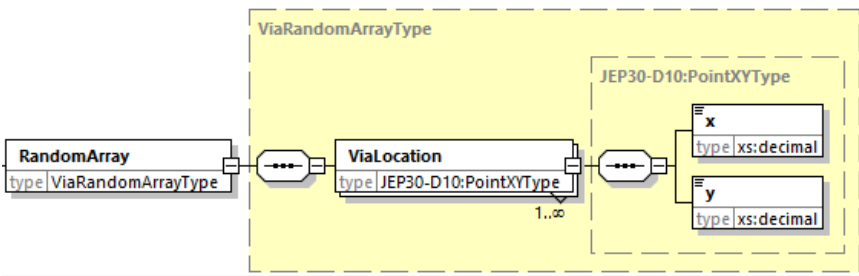
5.13.4.1 Standard Array

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/Via-Array/ViaLocation/StandardArray |
| diagram | <p>The diagram illustrates the structure of the StandardArray type. It is a class with a single attribute type of type ViaStandardArrayType. The ViaStandardArrayType is a complex type containing several attributes: ny (type xs:integer), nx (type xs:integer), dx (type JEP30-D10:ValueType), dy (type JEP30-D10:ValueType), and LowerLeftViaLocation (type JEP30-D10:PointXYType). The JEP30-D10:ValueType is a complex type with a single attribute Nominal (type xs:decimal). The JEP30-D10:PointXYType is a complex type with two attributes: x (type xs:decimal) and y (type xs:decimal).</p> |
| type | ViaStandardArrayType, JEP30-D10:ValueType, JEP30-D10:PointXYType. |

5.13.4.2 Circular Array

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/Via-Array/ViaLocation/CircularArray |
| diagram |  <p>The diagram illustrates the structure of the CircularArray. It starts with a box labeled 'CircularArray' with 'type ViaCircularArrayType'. This box is connected to a dashed-line container labeled 'ViaCircularArrayType'. Inside this container, there are several elements: 'PitchRadius' (type xs:decimal), 'Center' (type JEP30-D10:PointXYType), 'StartAngle' (type xs:integer), 'AngleToFill' (type xs:integer), 'AngleBetweenVias' (type xs:integer), and 'NumberOfVias' (type xs:integer). The 'Center' element is further connected to a dashed-line container labeled 'JEP30-D10:PointXYType', which contains two elements: 'x' (type xs:decimal) and 'y' (type xs:decimal). The 'AngleToFill' and 'AngleBetweenVias' elements are connected to a single box.</p> |
| type | ViaCircularArrayType, JEP30-D10:PointXYType. |

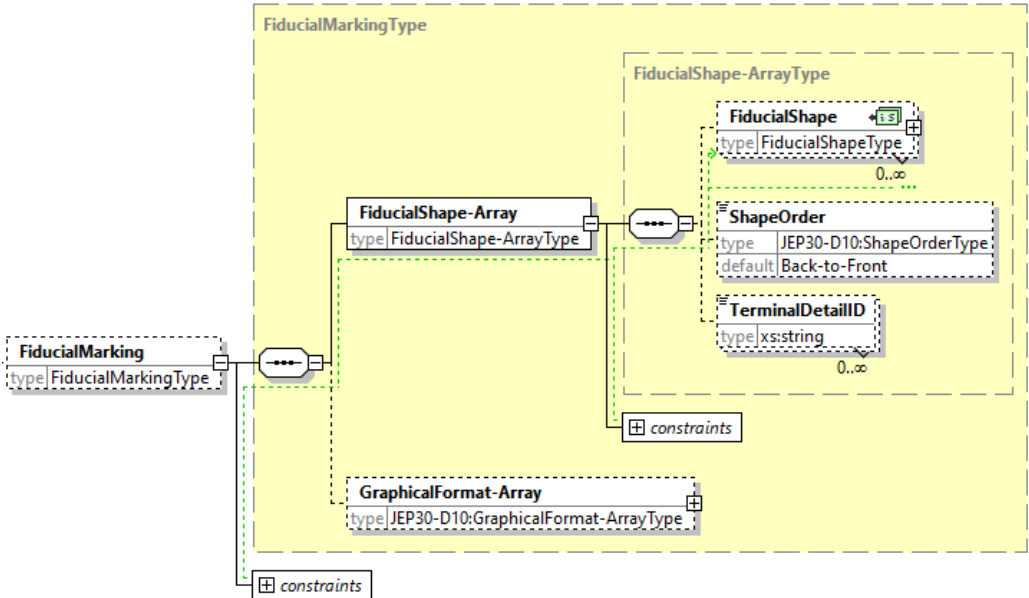
5.13.4.3 Random Array

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/Via-Array/ViaLocation/RandomArray |
| diagram |  <p>The diagram illustrates the structure of the RandomArray. It starts with a box labeled 'RandomArray' with 'type ViaRandomArrayType'. This box is connected to a dashed-line container labeled 'ViaRandomArrayType'. Inside this container, there is a box labeled 'ViaLocation' with 'type JEP30-D10:PointXYType'. The 'ViaLocation' box is connected to a dashed-line container labeled 'JEP30-D10:PointXYType', which contains two elements: 'x' (type xs:decimal) and 'y' (type xs:decimal). The 'ViaLocation' box has a multiplicity of '1..∞'.</p> |
| type | ViaRandomArrayType, JEP30-D10:PointXYType. |

5.13.4.4 Via Detail

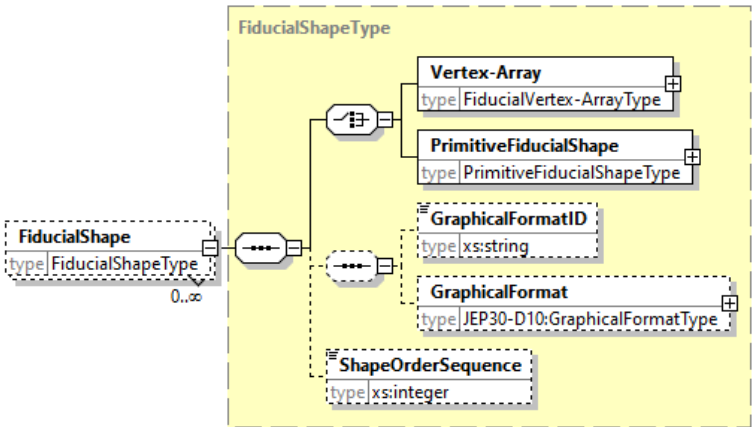
| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/Via-Array/ViaDetail |
| diagram | <p>The diagram illustrates the XSD structure for the ViaDetailType. It is a complex type containing several elements:</p> <ul style="list-style-type: none"> ViaDetail (type: ViaDetailType): An optional sequence of elements (0..∞). ViaCenter (type: ViaCenterType): An optional sequence of elements (0..1). TerminalGroupID (type: xs:string): A single element. RowTerminalIndex (type: JEP30-D10:MinIntegerOfOneType): A single element with minIncl/maxIncl of 1. ColumnTerminalIndex (type: JEP30-D10:MinIntegerOfOneType): A single element with minIncl/maxIncl of 1. PolarTerminalIndex (type: JEP30-D10:MinIntegerOfOneType): A single element with minIncl/maxIncl of 1. ViaStatus (type: ViaStatusType): A single element. <p>The ViaCenterType contains the following elements:</p> <ul style="list-style-type: none"> x (type: xs:decimal): A single element. y (type: xs:decimal): A single element. PitchRadius (type: xs:decimal): A single element. Center (type: JEP30-D10:PointXYType): A single element. Angle (type: xs:integer): A single element. <p>The ViaStatusType contains the following elements:</p> <ul style="list-style-type: none"> Capped (type: JEP30-D10:EmptyType): A single element. Filled (type: JEP30-D10:EmptyType): A single element. PlatedShut (type: JEP30-D10:EmptyType): A single element. Deleted (type: JEP30-D10:EmptyType): A single element. |
| type | ViaDetailType , ViaCenterType , JEP30-D10:PointXYType , JEP30-D10:MinIntegerOfOneType , ViaStatusType , JEP30-D10:EmptyType . |

5.14 Fiducial Marking

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/FiducialMarking |
| diagram |  <p>The diagram illustrates the structure of the FiducialMarking type. It is a container for a FiducialShape-Array (type FiducialShape-ArrayType). The FiducialShape-Array is a sequence of FiducialShape elements (type FiducialShapeType). Each FiducialShape contains a ShapeOrder (type JEP30-D10:ShapeOrderType, default Back-to-Front) and a TerminalDetailID (type xs:string). The FiducialShape-Array also includes a GraphicalFormat-Array (type JEP30-D10:GraphicalFormat-ArrayType) and a constraints block. The FiducialShape also includes a constraints block.</p> |
| type | FiducialMarkingType , FiducialShape-ArrayType , FiducialShapeType , JEP30-D10:ShapeOrderType , JEP30-D10:GraphicalFormat-ArrayType |

A fiducial marker or fiducial is an object placed in the field of view of an imaging system that appears in the image produced, for use as a point of reference or a measure. It may be either something placed into or on the imaging subject, or a mark or set of marks in the reticle of an optical instrument.

5.14.1 Fiducial Shape

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/FiducialMarking/FiducialShape-Array/FiducialShape |
| diagram |  <p>The diagram illustrates the structure of the FiducialShape type. It is a container for a Vertex-Array (type FiducialVertex-ArrayType), a PrimitiveFiducialShape (type PrimitiveFiducialShapeType), a GraphicalFormatID (type xs:string), a GraphicalFormat (type JEP30-D10:GraphicalFormatType), and a ShapeOrderSequence (type xs:integer). The FiducialShape also includes a constraints block.</p> |
| type | FiducialShapeType , FiducialVertex-ArrayType , PrimitiveFiducialShapeType , JEP30-D10:GraphicalFormat-ArrayType |

5.14.1 Fiducial Shape (cont'd)

The Shape can be created from a choice of Vertices, or primitive shapes.

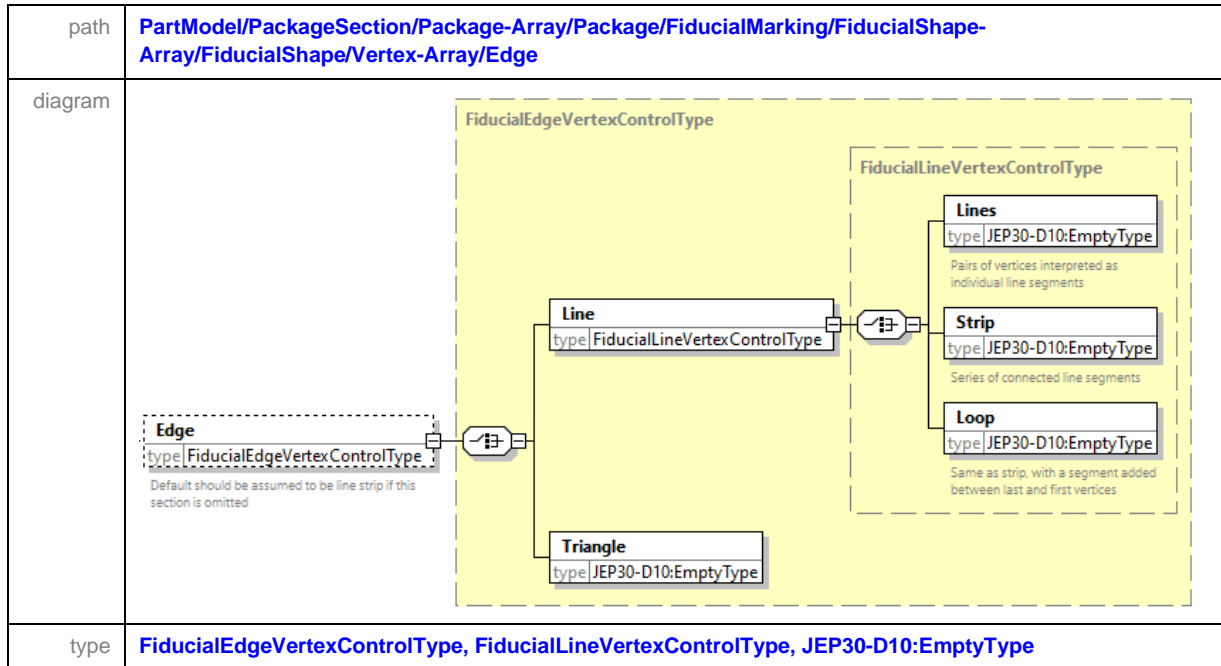
GraphicalFormatID serves as a reference ID for the *GraphicalFormat*, that is defined under the *GraphicalFormat-Array/GraphicalFormats* where a set of graphical formats can be defined that can be standardized across multiple *Shape* entries. The addition of the *GraphicalFormat* under *Shape* enables unique modification of a references Graphical Format for applying to this specific shape instance.

The *ShapeOrderSequence* can be used to define the order of shapes for a complex fiducial that may consist of various shapes. It works in tandem with the *ShapeOrder* that is defined under *FiducialMarking/FiducialShape-Array*.

5.14.1.1 Vertex - Array

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/FiducialMarking/FiducialShape-Array/FiducialShape/Vertex-Array |
| diagram | <pre> classDiagram class VertexArray { type: FiducialVertex-ArrayType } class Vertex { type: FiducialVertexType } class Edge { type: FiducialEdgeVertexControlType } class FiducialVertexType { x: xs:decimal y: xs:decimal FeatureControl-IDs: JEP30-D10:FeatureControl-IDsType } VertexArray --> Vertex : 1..∞ VertexArray --> Edge Vertex --> FiducialVertexType </pre> <p>Default should be assumed to be line strip if this section is omitted</p> |
| type | FiducialVertex-ArrayType , FiducialVertexType , JEP30-D10:FeatureControl-IDsType , FiducialEdgeVertexControlType |

5.14.1.1.1 Edge



The default is assumed to be *Line/Strip* if the *Edge* element is omitted. The *Edge* structure governs how each of the vertices in the unbounded *Vertex* element is to be processed.

Line/Lines represent pairs of vertices interpreted as individual line segments. Vertices 1 and 2 make one line segment whereas vertices 3 and 4 makes the 2nd line segment. Vertices 2 and 3 are not connected with a line segment.

Line/Strip represent a series of connected line segments. Vertices 1 and 2 make one line segment whereas vertices 2 and 3 makes the 2nd line segment, and vertices 3 and 4 makes the 3rd line segment.

Lines/Loop is the same as strip, with a segment added between last and first vertices. From the previous example, vertices 4 and 1 makes the final line segment, creation a closed loop.

Triangle represent each triple set of vertices to be interpreted as an individual triangle. Vertices 1, 2, and 3 make one triangle whereas vertices 4, 5, and 6 makes the 2nd triangle.

5.14.1.2 Primitive Fiducial Shape

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/FiducialMarking/FiducialShape-Array/FiducialShape/PrimitiveFiducialShape |
| diagram | |
| type | PrimitiveFiducialShapeType , ReferenceRectangleType , ReferenceCrosshairType , JEP30-D10:DimensionalValueSetType , ReferenceRegularPolygonType , ReferenceCircleType , JEP30-D10:PointXYType |

5.14.2 Graphical Format - Array

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/FiducialMarking/GraphicalFormat-Array |
| diagram | |
| type | JEP30-D10:GraphicalFormat-ArrayType , GraphicalFormatsType , GraphicalFormatType . |

5.14.2.1 Graphical Format

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/FiducialMarking/FiducialShape-Array/FiducialShape/GraphicalFormat |
| diagram | |
| type | GraphicalFormatType, StrokeWidthType, StrokeOpacityType, StrokeLineCapType, StrokeDash-ArrayType, ColorType, FillOpacityType |

5.14.2.1.1 Color Type

| | |
|---------|---|
| path | <div>PartModel/PackageSection/Package-Array/Package/FiducialMarking/FiducialShape-Array/FiducialShape/GraphicalFormat/StrokeColor</div> <div>PartModel/PackageSection/Package-Array/Package/FiducialMarking/FiducialShape-Array/FiducialShape/GraphicalFormat/FillColor</div> |
| diagram | |
| type | DefaultColorType, DefaultColorNameType, DefaultColorHexType, DefaultColor-RType, DefaultColor-GType, DefaultColor-BType |

[ColorNameType](#) as defined in the [StrokeColor/Name](#) element list the names of the Scalable Vector Graphics (SVG) Colors as defined by the Scalable Vector Graphics (SVG) Specification.

[ColorHexType](#) as defined in the [StrokeColor/Hex](#) element uses a regular expression pattern to recognize the six-digit hexadecimal representation of the Scalable Vector Graphics (SVG) Colors.

[Color-RType](#), [Color-GType](#), [Color-BType](#) allows the construction of all the colors from the combination of the red, green, and blue colors in the RGB Color Space. The red, green, and blue use 8 bits each, which have integer values from 0 to 255.

5.15 Geometric Dimensioning and Tolerancing

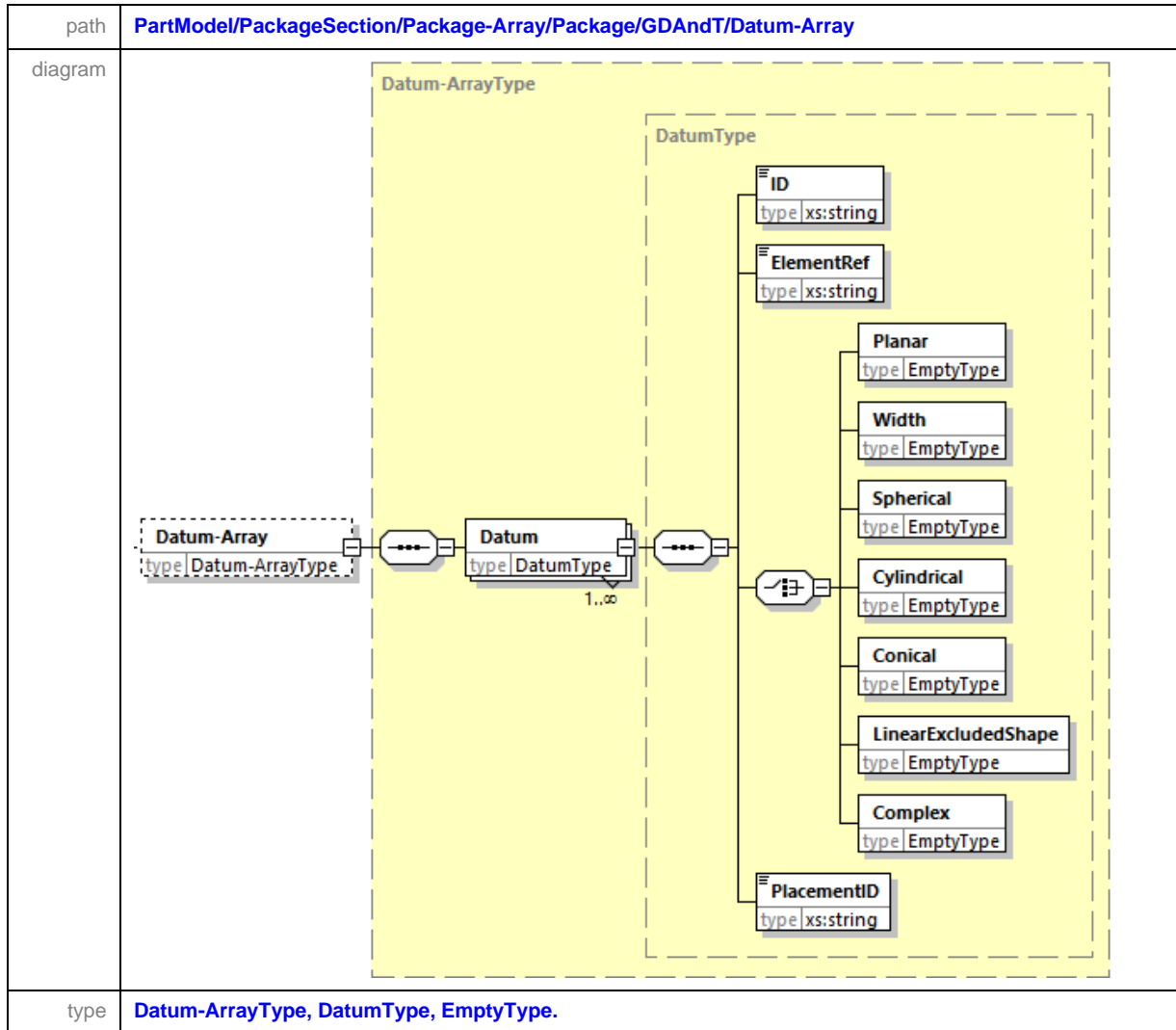
| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/GDAndT, PartModel/PackageSection/Package-Array/Package/GDandTDatum-to-ElementMap |
| diagram | |
| type | JEP30-D10:GDAndTType, Datum-ArrayType, FeatureControl-ArrayType, FeatureControlPlacement-ArrayType |

When applying GD&T, the first consideration is to establish a datum reference frame based on the function of the part in the assembly with its mating parts. After the datum reference plane is established, the form of the primary datum features is controlled, followed by the orientation and / or location of the secondary and tertiary datum features. After the datum features are related relative to each other, the remaining features are controlled for orientation and location relative to the datum reference framework.

As a result of increased decentralization of design and manufacturing, it is even more important that the design more precisely states the functional requirements. To accomplish this, it is becoming increasingly more important that the use of geometric and dimensioning replace the former limit dimensioning for form, orientation, location, and profile of part features. This section follows the ASME Y14.5-2009 Dimensioning and Tolerancing which establishes uniform practices for stating and interpreting dimensioning, Tolerancing and related requirements for use on engineering drawings and in related documents.

Refer to this ASME Y14.5-2009 Dimensioning and Tolerancing standard for a detailed explanation of the terminology used in association with this section.

5.15.1 Datum Array

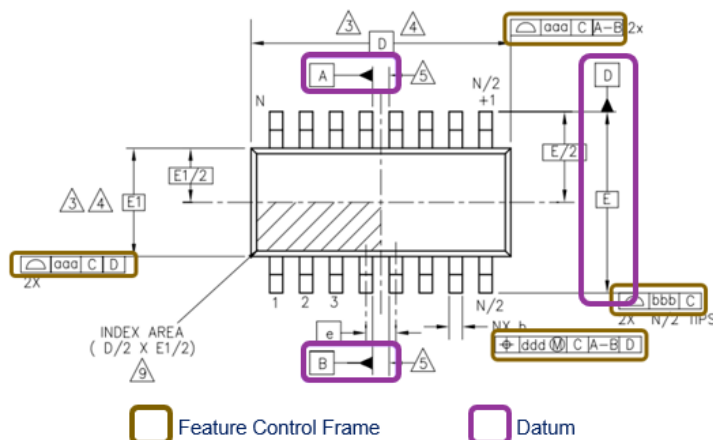


A datum feature is selected based on its functional relationship to the tolerance feature and the requirements of the design.

Figure 15 – Example showing Datum and Feature Control Frames

5.15.1 Datum Array (cont'd)

The diagram below from the ASME Y14.5-2009 Dimensioning and Tolerancing standard for a detailed explanation of the terminology used in association with this section.



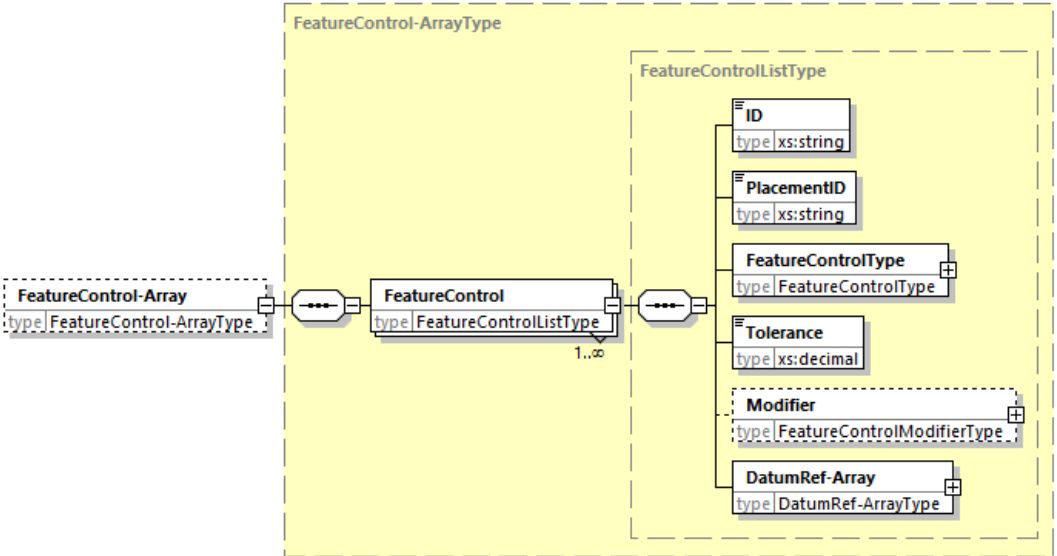
5.15.1 Datum Array (cont'd)

| FEATURE TYPE | ON THE DRAWING | DATUM FEATURE | DATUM AND DATUM FEATURE SIMULATOR | DATUM AND CONSTRAINING DEGREES OF FREEDOM |
|------------------------------|----------------|---------------|-----------------------------------|---|
| PLANAR (a) | | | | |
| WIDTH (b) | | | | |
| SPHERICAL (c) | | | | |
| CYLINDRICAL (d) | | | | |
| CONICAL (e) | | | | |
| LINEAR EXTRUDED SHAPE (f) | | | | |
| COMPLEX (g) | | | | |

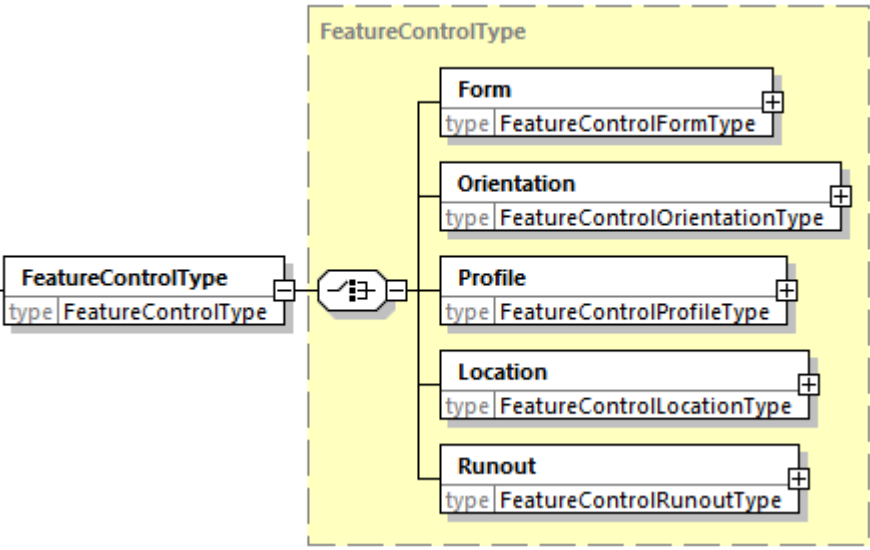
Source: ASME Y14-5 2009

Figure 16 – Constrained Degrees of Freedom for Primary Datum Features

5.15.2 Feature Control Array

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/GDAndT/FeatureControl-Array |
| diagram |  <p>The diagram illustrates the structure of the FeatureControl-ArrayType. It is composed of a sequence of FeatureControl elements. Each FeatureControl element is of type FeatureControlListType and has a cardinality of 1..∞. The FeatureControlListType is further detailed as containing the following elements:</p> <ul style="list-style-type: none"> ID: type xs:string PlacementID: type xs:string FeatureControlType: type FeatureControlType (indicated with a '+' sign for optional) Tolerance: type xs:decimal Modifier: type FeatureControlModifierType (indicated with a '+' sign for optional) DatumRef-Array: type DatumRef-ArrayType (indicated with a '+' sign for optional) |
| type | FeatureControl-ArrayType , FeatureControlListType , FeatureControlType , FeatureControlModifierType , DatumRef-ArrayType . |

5.15.2.1 Feature Control Type

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/GDAndT/FeatureControl-Array/FeatureControl/FeatureControlType |
| diagram |  <p>The diagram illustrates the structure of the FeatureControlType. It is composed of a sequence of FeatureControlType elements. Each FeatureControlType element is of type FeatureControlType and has a cardinality of 1..∞. The FeatureControlType is further detailed as containing the following elements:</p> <ul style="list-style-type: none"> Form: type FeatureControlFormType (indicated with a '+' sign for optional) Orientation: type FeatureControlOrientationType (indicated with a '+' sign for optional) Profile: type FeatureControlProfileType (indicated with a '+' sign for optional) Location: type FeatureControlLocationType (indicated with a '+' sign for optional) Runout: type FeatureControlRunoutType (indicated with a '+' sign for optional) |
| type | FeatureControlTypeType , FeatureControlFormType , FeatureControlOrientationType , FeatureControlProfileType , FeatureControlLocationType , FeatureControlRunoutType . |

5.15.2.1.1 Form

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/GDAndT/FeatureControl-Array/FeatureControl/FeatureControlType/Form |
| diagram | |
| type | FeatureControlFormType , EmptyType . |

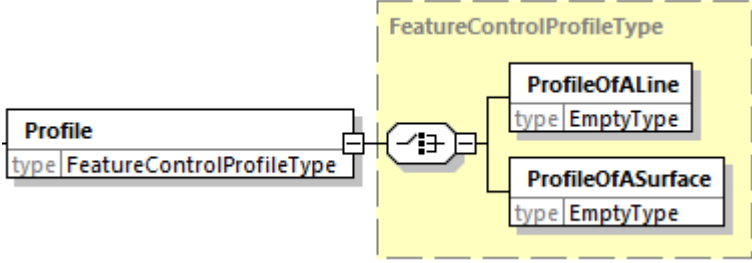
Form tolerances control straightness, flatness, circularity, and cylindricity. See section on “Tolerance of Form” in the ASME Y14-5 2009 for more details.

5.15.2.1.2 Orientation

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/GDAndT/FeatureControl-Array/FeatureControl/FeatureControlType/Orientation |
| diagram | |
| type | FeatureControlOrientationType , EmptyType . |

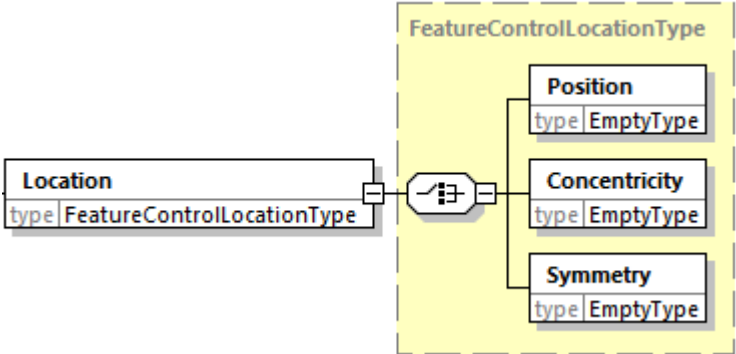
An orientation tolerance controls parallel, perpendicular, and all other angular relationships. See section on “Tolerance of Orientation” in the ASME Y14-5 2009 for more details.

5.15.2.1.3 Profile

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/GDAndT/FeatureControl-Array/FeatureControl/FeatureControlType/Profile |
| diagram |  <p>The diagram shows a box labeled 'Profile' with 'type FeatureControlProfileType' below it. This box is connected to a dashed yellow box labeled 'FeatureControlProfileType'. Inside this dashed box, there is a feature control symbol (a circle with a vertical line and a horizontal line) connected to two sub-features: 'ProfileOfALine' and 'ProfileOfASurface'. Both sub-features have 'type EmptyType' below them.</p> |
| type | FeatureControlProfileType , EmptyType . |

A profile is an outline of a surface, a shape made up of one or more features, or a two-dimensional element of one or more features. Profile tolerances are used to define a tolerance zone to control form or combinations of size, form, orientation, and location of a feature(s) relative to a true profile. See section on “Tolerance of Profile” in the ASME Y14-5 2009 for more details.

5.15.2.1.4 Location

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/GDAndT/FeatureControl-Array/FeatureControl/FeatureControlType/Location |
| diagram |  <p>The diagram shows a box labeled 'Location' with 'type FeatureControlLocationType' below it. This box is connected to a dashed yellow box labeled 'FeatureControlLocationType'. Inside this dashed box, there is a feature control symbol (a circle with a vertical line and a horizontal line) connected to three sub-features: 'Position', 'Concentricity', and 'Symmetry'. Each sub-feature has 'type EmptyType' below it.</p> |
| type | FeatureControlLocationType , EmptyType . |

Included in the principles of tolerances of location. are position, concentricity, and symmetry used to control the following relationships:

1. Center distance between features of size such as holes, slots, bosses, and tabs.
2. Location of features of size (such as in the previous bullet) as a group, from datum features, such as plane and cylindrical surfaces.
3. Coaxiality of features of size, and
4. Concentricity or symmetry of features of size—center distances of correspondingly located feature elements equally disposed about a datum axis or plane

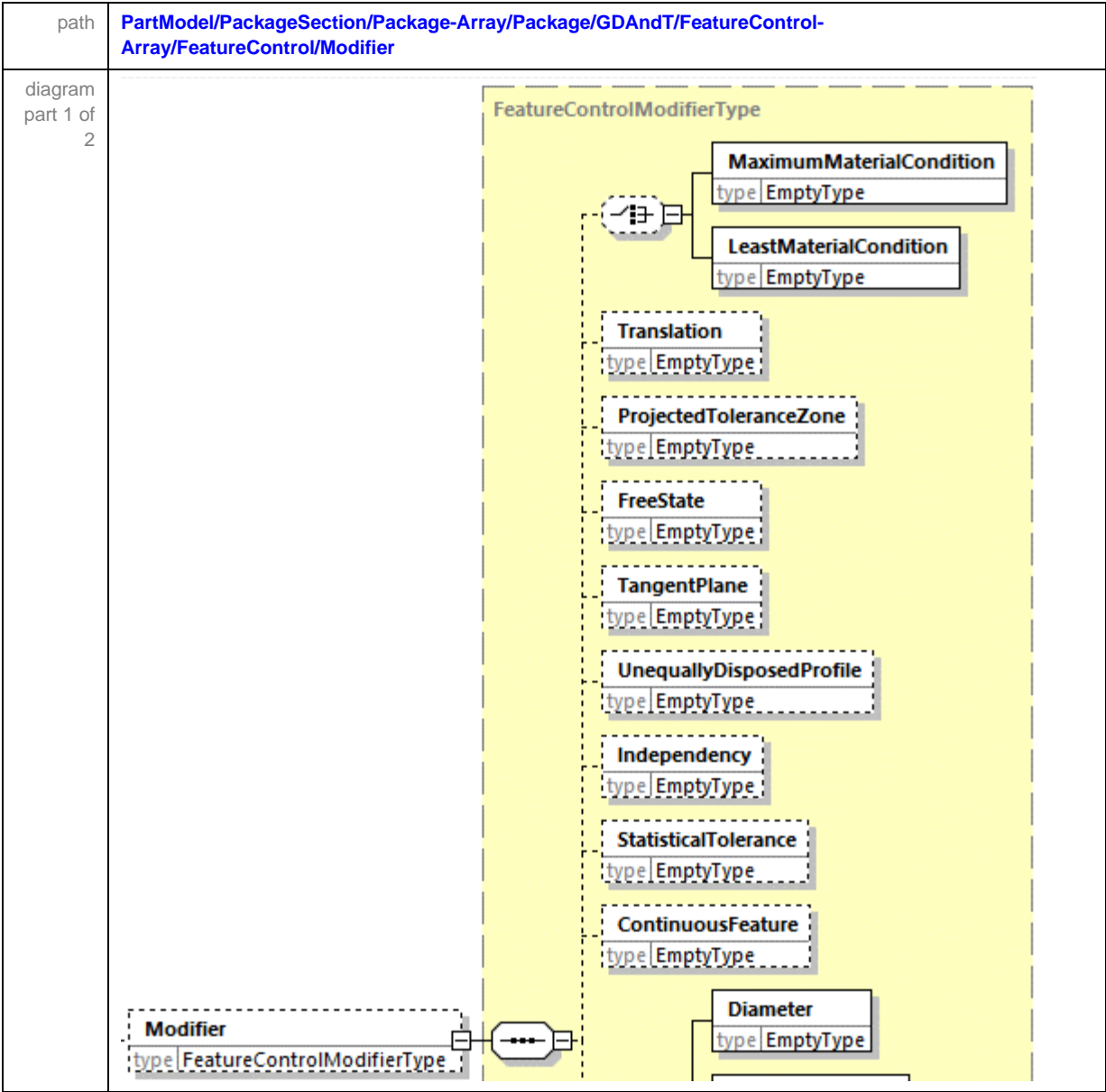
See section on “Tolerance of Location” in the ASME Y14-5 2009 for more details.

5.15.2.1.5 Runout

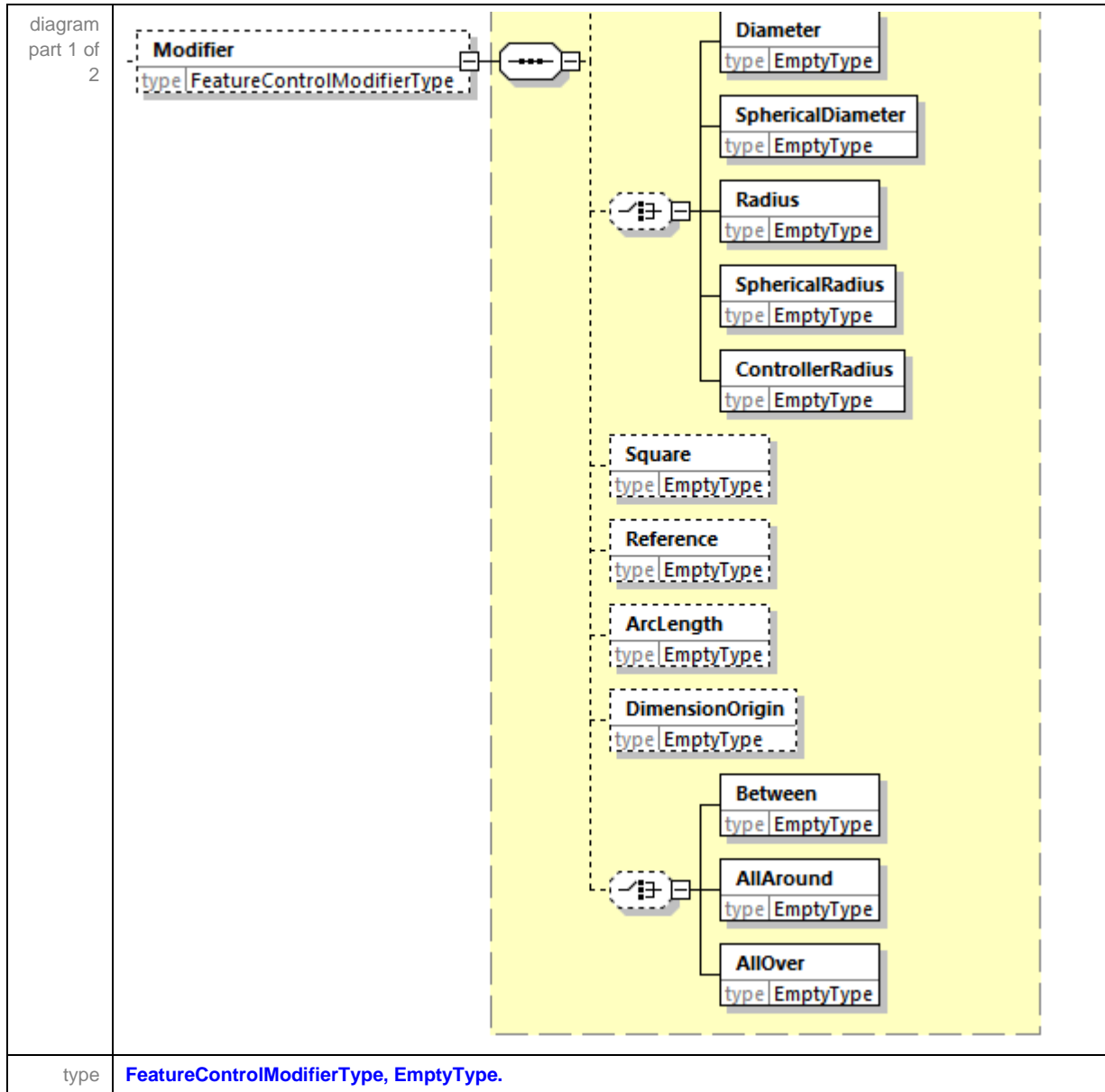
| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/GDAndT/FeatureControl-Array/FeatureControl/FeatureControlType/Runout |
| diagram | <pre>classDiagram class Runout { type FeatureControlRunoutType } class FeatureControlRunoutType { Runout TotalRunout } Runout "1" -- "*" FeatureControlRunoutType class RunoutSub { type EmptyType } class TotalRunoutSub { type EmptyType } FeatureControlRunoutType < -- RunoutSub FeatureControlRunoutType < -- TotalRunoutSub</pre> |
| type | FeatureControlRunoutType, EmptyType. |

Runout is a tolerance used to control the functional relationship of one or more features to a datum axis established from a datum feature specified at RMB. The types of features controlled by runout tolerances include those surfaces constructed around a datum axis and those constructed at right angles to a datum axis. See section on “Tolerance of Runout” in the ASME Y14-5 2009 for more details.

5.15.3 Modifier

























5.15.3 Modifier (cont'd)



Modifiers establishes additional symbols for specifying geometric characteristics and other dimensional requirements on engineering drawings. The application of Modifiers such as Most Material Condition or Least Material Condition may be applied to geometric tolerance values on features of size. See section on “Applicability of Modifiers on Geometric Tolerance Values and Datum Feature References” in the ASME Y14-5 2009 for more details.

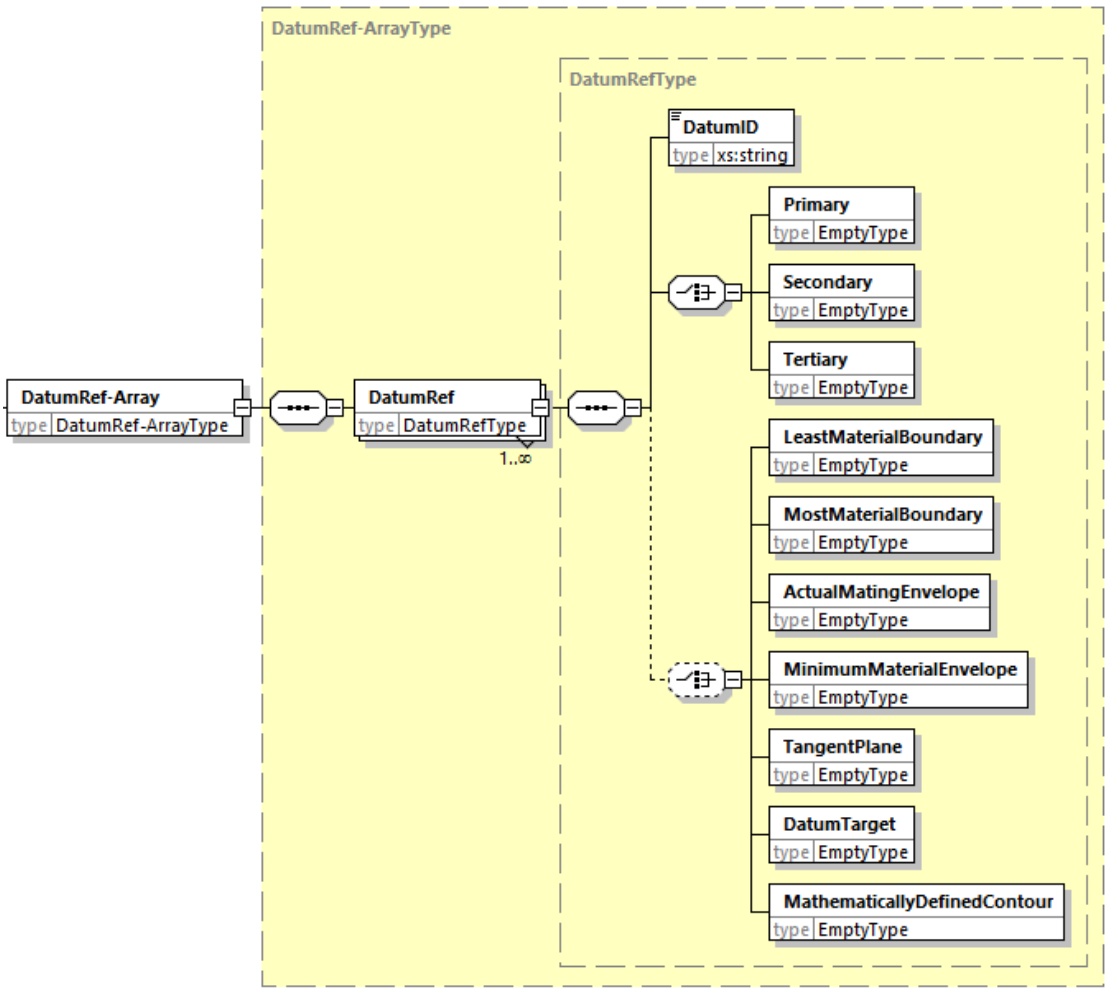
5.15.3 Modifier (cont'd)

| TERM | SYMBOL | SEE: |
|---|---|--------|
| AT MAXIMUM MATERIAL CONDITION (When applied to a tolerance value) AT MAXIMUM MATERIAL BOUNDARY (When applied to a datum reference) |  | 3.3.5 |
| AT LEAST MATERIAL CONDITION (When applied to a tolerance value) AT LEAST MATERIAL BOUNDARY (When applied to a datum reference) |  | 3.3.5 |
| TRANSLATION |  | 3.3.26 |
| PROJECTED TOLERANCE ZONE |  | 3.3.6 |
| FREE STATE |  | 3.3.20 |
| TANGENT PLANE |  | 3.3.21 |
| UNEQUALLY DISPOSED PROFILE |  | 3.3.22 |
| INDEPENDENCY |  | 3.3.24 |
| STATISTICAL TOLERANCE |  | 3.3.10 |
| CONTINUOUS FEATURE |  | 3.3.23 |
| DIAMETER |  | 3.3.7 |
| SPHERICAL DIAMETER |  | 3.3.7 |
| RADIUS |  | 3.3.7 |
| SPHERICAL RADIUS |  | 3.3.7 |
| CONTROLLED RADIUS |  | 3.3.7 |
| SQUARE |  | 3.3.16 |
| REFERENCE |  | 3.3.8 |
| ARC LENGTH |  | 3.3.9 |
| DIMENSION ORIGIN |  | 3.3.17 |
| BETWEEN |  | 3.3.11 |
| ALL AROUND |  | 3.3.19 |
| ALL OVER |  | 3.3.25 |

Source: ASME Y14-5 2009

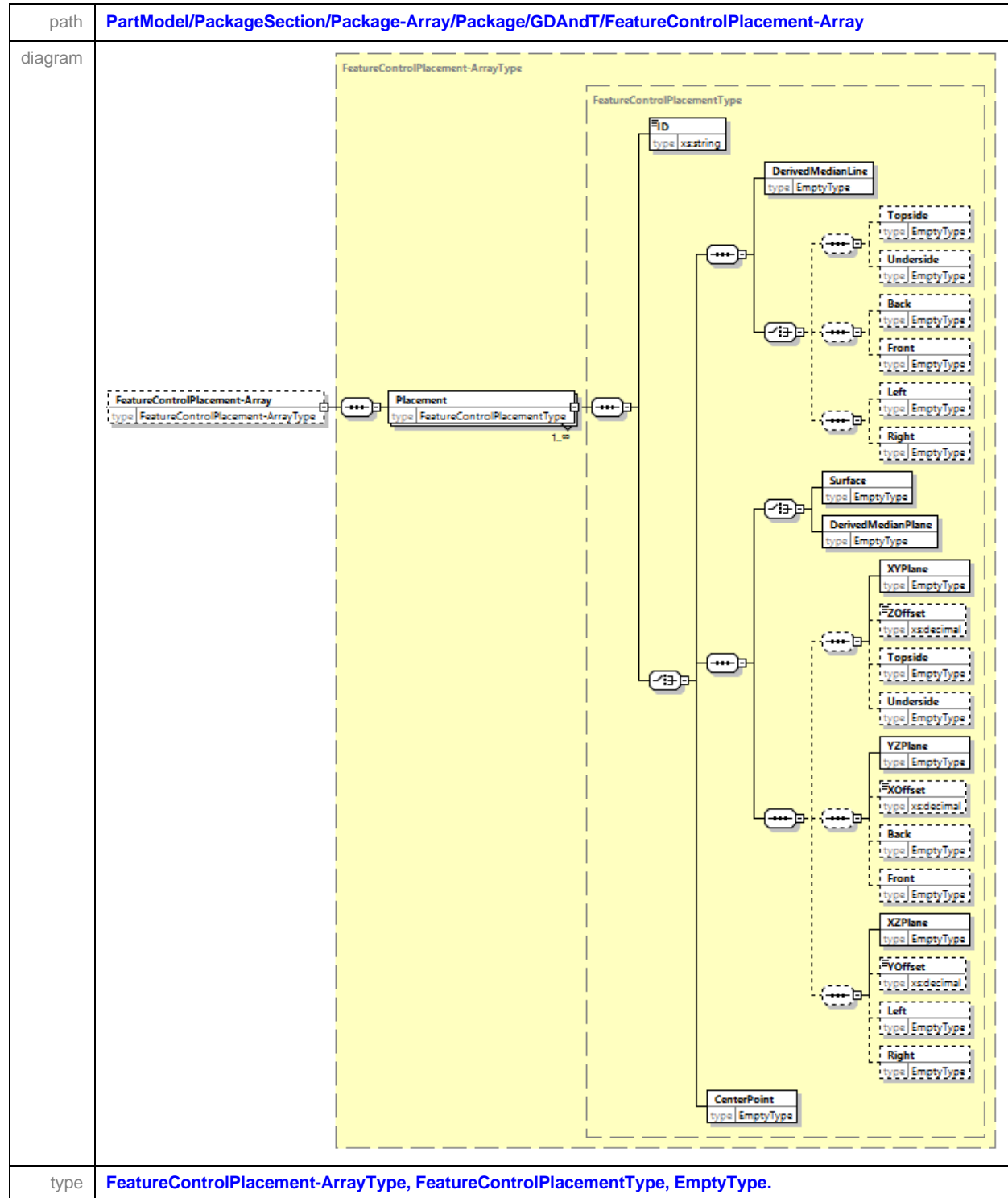
Figure 17 – Modifying Symbols

5.15.4 Datum Reference Array

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/GDAndT/FeatureControl-Array/FeatureControl/DatumRef-Array |
| diagram |  <p>The diagram illustrates the structure of a Datum Reference Array. It shows a DatumRef-Array element containing a sequence of DatumRef elements. Each DatumRef element contains a DatumRefType element. The DatumRefType element is further detailed with a DatumID (type xs:string) and a choice of various datum types: Primary, Secondary, Tertiary, LeastMaterialBoundary, MostMaterialBoundary, ActualMatingEnvelope, MinimumMaterialEnvelope, TangentPlane, DatumTarget, and MathematicallyDefinedContour. All these datum types have a type of EmptyType. The DatumRef element has a cardinality of 1..∞, and the DatumRef-Array element has a cardinality of 1..∞.</p> |
| type | DatumRef-ArrayType, DatumRefType. |

A datum reference frame is three mutually perpendicular intersecting datum planes. See section on “Datum Reference Frames” in the ASME Y14-5 2009 for more details.

5.15.5 Feature Control Placement Array



A feature control frame is related to a considered feature by one of the following methods

1. Locating the frame below or attached to a leader-directed note or dimension pertaining to the feature.

5.15.5 Feature Control Placement Array (cont'd)

2. Attaching a leader from the frame pointing to the feature.
3. Attaching a side, corner, or an end of the frame to an extension line from the feature, provided it is a plane surface,
4. Attaching a side, corner, or an end of the frame to an extension of the dimension line pertaining to a feature of size, and
5. Placing in a note, chart, or the general tolerance block.

See section on “Datum Reference Frames” in the ASME Y14-5 2009 for more details.

5.15.6 GD&T XML Example

Figure 14 – Sample for Representation in an XML Structure represents an SOIC device with a Feature Control Frame highlighted that is represented in the xml example below.

```
<GDAndT>
  <FeatureControl-Array>
    <FeatureControl>
      <ID>Feature Control ID 1</ID>
      <PlacementID>Place ID X</PlacementID>
      <FeatureControlType>
        <Location>
          <Position/>
        </Location>
      </FeatureControlType>
      <Tolerance>0.25</Tolerance>
      <Modifier>
        <MaximumMaterialBoundary/>
      </Modifier>
      <DatumRef-Array>
        <DatumRef>
          <DatumID>C</DatumID>
          <Primary/>
        </DatumRef>
        <DatumRef>
          <DatumID>A</DatumID>
          <Secondary/>
        </DatumRef>
        <DatumRef>
          <DatumID>B</DatumID>
          <Secondary/>
        </DatumRef>
        <DatumRef>
          <DatumID>D</DatumID>
          <Tertiary/>
        </DatumRef>
      </DatumRef-Array>
    </FeatureControl>
  </FeatureControl-Array>
  ...
  <FeatureControlPlacement-Array>
    <Placement>
      <ID> Place ID X</ID>
      <DerivedMedianPlane/>
    </Placement>
  </FeatureControlPlacement-Array>
</GDAndT>
```

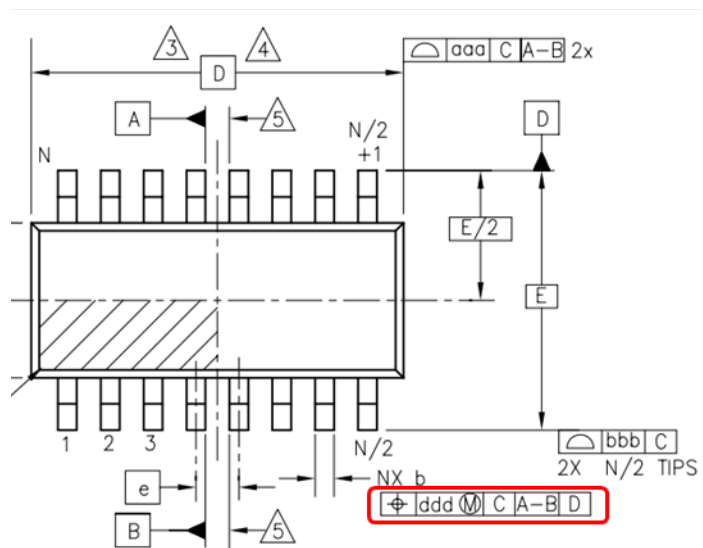


Figure 18 – Sample for Representation in an XML Structure

5.15.6 GD&T XML Example (cont'd)

```

    <XZPlane/>
    <Left/>
    <Right/>
  </Placement>
</FeatureControlPlacement-Array>
</GDAndT>
...
<FeatureControl-ID>
  <FeatureControlID>456</FeatureControlID>
  <FeatureControlPlacementID>BBB</FeatureControlPlacementID>
</FeatureControl-ID>

```

5.15.7 GD and T Datum-to-Element Map

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/GDandTDatum-to-ElementMap-Array/GDandTDatum-to-ElementMap |
| diagram | |
| type | GDandTDatum-to-ElementMapType, JEP30-D10:EmptyType |

5.16 Recommended Footprint - Array

| | | |
|---------|---|--|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array | |
| diagram | | |
| type | RecommendedFootprint-ArrayType, RecommendedFootprintType, JEP30-D10:EmptyType, InterconnectTechnology-ArrayType, RecommendedPadOrHole-ArrayType, ThermalRelief-ArrayType, AssemblyOutlineLayerType, ConductiveArea-ArrayType, PlacementOutlineType, Keep-inLayer-ArrayType, KeepoutRegion-ArrayType, SoldermaskLayer-ArrayType, PasteMaskLayer-ArrayType. | |

The *FootprintName* and *FootprintExtendedName* is described in JESD30, “Descriptive Designation System for Electronic-device Packages and Footprints”.

5.16.1 Interconnect Technology - Array

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/InterconnectTechnology-Array |
| diagram | <p>The diagram illustrates the structure of the <code>InterconnectTechnology-Array</code> element. It is a sequence of elements: <code>InterconnectTechnology-Array</code> (type <code>InterconnectTechnology-ArrayType</code>), <code>InterconnectTechnology</code> (type <code>InterconnectTechnologyType</code>, 1..∞), and a choice of <code>ID</code> (type <code>xs:string</code>), <code>FiniteSoldering</code> (type <code>JEP30-D10:EmptyType</code>), <code>InfiniteSoldering</code> (type <code>JEP30-D10:EmptyType</code>), <code>Pressfit</code> (type <code>JEP30-D10:EmptyType</code>), <code>AssemblyTechnologyID</code> (type <code>xs:string</code>), and <code>OtherTechnologyClass</code> (type <code>xs:string</code>).</p> |
| type | InterconnectTechnology-ArrayType , InterconnectTechnologyType , JEP30-D10:EmptyType |

5.16.2 Recommended Pad Or Hole Shape - Array

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/RecommendedPadOrHoleShape-Array |
| diagram | <p>The diagram illustrates the structure of the <code>RecommendedPadOrHoleShape-Array</code> element. It is a sequence of elements: <code>RecommendedPadOrHoleShape-Array</code> (type <code>RecommendedPadOrHole-ArrayType</code>), <code>RecommendedPadOrHoleShape</code> (type <code>RecommendedPadOrHoleType</code>, 0..∞), and <code>PadGroupToPadGroupRelationship</code> (type <code>PadGroupToPadGroupRelationshipType</code>, 0..∞).</p> |
| type | RecommendedPadOrHole-ArrayType , RecommendedPadOrHoleType , PadGroupToPadGroupRelationshipType |

5.16.2.1 Recommended Pad or Hole Shape

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/RecommendedPadOrHoleShape-Array/RecommendedPadOrHoleShape |
| diagram | |
| type | RecommendedPadOrHoleShapeType , RecommendedSurfaceMountPadShapeType , RecommendedThroughHoleType , RecommendedPadOrHoleLocationType , RecommendedPadOrHolePatternGroupType , RecommendedLandPatternSpanType , RecommendedLandPatternSpacingType . |

Although not recommended for most terminal types, the component manufacturer may provide their recommended land pattern or hole mounting requirement for the part. This is typically not good business practice as the component manufacturer can not necessarily control or know in advance all the possible use cases of the part, resulting in a one size fits all user environment. This seldom leads to a high yield impacting footprint. The exception to this rule is for the following terminal types since final hole sizes is critical to be properly matched to the dimensions of the pin or pressfit terminal. Recommended Pad or Hole Shape sizes are applicable to the following terminal types:

- Pressfit – All variations
- Column – Microspring solid core and air-core
- Pin – Press-in Solderable Terminal
- Pin – Swage Fastening Pin
- Pin – Press-in Non-Solderable

Although not recommended for most terminal types, the component manufacturer may provide

5.16.2.1.1 Surface Mount

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/RecommendedPadOrHoleShape-Array/RecommendedPadOrHoleShape/SurfaceMount |
| diagram | <p>The diagram illustrates the class hierarchy for surface mount pad shapes. A central class, SurfaceMount, has a 'type' attribute of RecommendedSurfaceMountPadShapeType. This attribute points to a dashed box containing a collection of subclasses, each with its own 'type' attribute:</p> <ul style="list-style-type: none">Rectangle: type RectangleValueTypeRoundedRectangle: type RoundedRectangleValueTypeHalfRoundedRectangle: type HalfRoundedRectangleValueTypeModifiedRectangle: type ModifiedRectangleValueTypeCircle: type CircleValueTypeD-Shape: type D-ShapeValueTypeRoundedRectangleD-Shape: type D-ShapeRoundedRectangleValueTypeDouble-D: type Double-DValueTypeContour: type ContourShapeValueTypePadVoid-Array: type PadVoid-ArrayTypeThermalReliefID: type xs:string |
| type | RecommendedSurfaceMountPadShapeType, RectangleValueType, RoundedRectangleValueType, HalfRoundedRectangleValueType, ModifiedRectangleValueType, CircleValueType, D-ShapeValueType, D-ShapeRoundedRectangleValueType, Double-DValueType, ContourShapeValueType, PadVoid-ArrayType. |

The definition of each shape is outlined in Annex A (informative) Shape Definitions & Dimensions. Recommended land pattern only require the nominal defined and not any of the tolerances nor the min and max values.

5.16.2.1.1.1 Pad Void - Array

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/RecommendedPadOrHoleShape-Array/RecommendedPadOrHoleShape/SurfaceMount/PadVoid-Array |
| diagram | |
| type | PadVoid-ArrayType , PadVoidType , PadVoidShapeType , PadVoidLocationType . |

5.16.2.1.1.1.1 Pad Void Shape

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/RecommendedPadOrHoleShape-Array/RecommendedPadOrHoleShape/SurfaceMount/PadVoid-Array/PadVoid/PadVoidLocation/VoidStatus |
| diagram | |
| type | PadVoidShapeType , RectangleValueType , RoundedRectangleValueType , ModifiedRectangleValueType , CircleValueType , Double-DValueType , ContourShapeValueType . |

The definition of each shape is outlined in Annex A (informative) Shape Definitions & Dimensions.

5.16.2.1.1.2 Pad Void Location

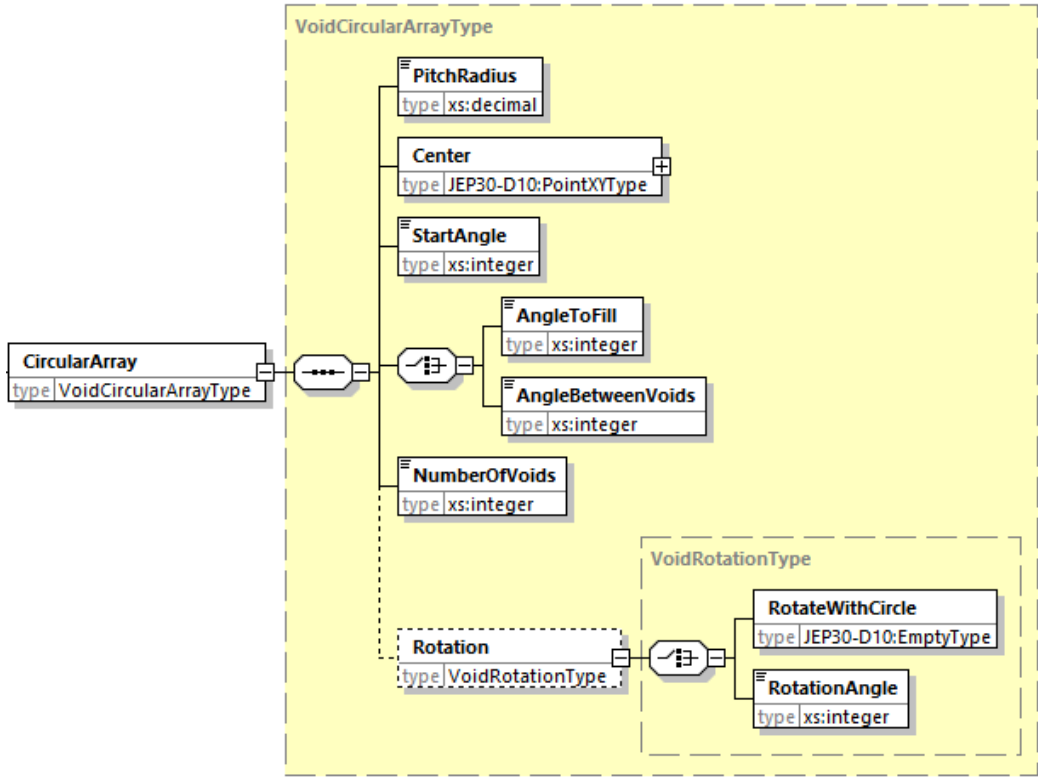
| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/RecommendedPadOrHoleShape-Array/RecommendedPadOrHoleShape/SurfaceMount/PadVoid-Array/PadVoid/PadVoidLocation |
| diagram | |
| type | PadVoidLocationType, VoidStandardArrayType, VoidCircularArrayType, DeletedStatusType, VoidRandomArrayType, JEP30-D10:EmptyType, JEP30-D10:PointXYType, JEP30-D10:MinIntegerOfOneType. |

For *PadVoidLocation* defined via *StandardArray* or *CircularArray*, some voids in the array may be deleted. This is covered in the *VoidStatus* branch. Alternatively, *PadVoidLocation* can also be defined via *RandomArray* when there is just 1 Pad-or-Hole, or when there is no logical structure to the location of the Pads-or-Holes.

5.16.2.1.1.1.2.1 Standard Array

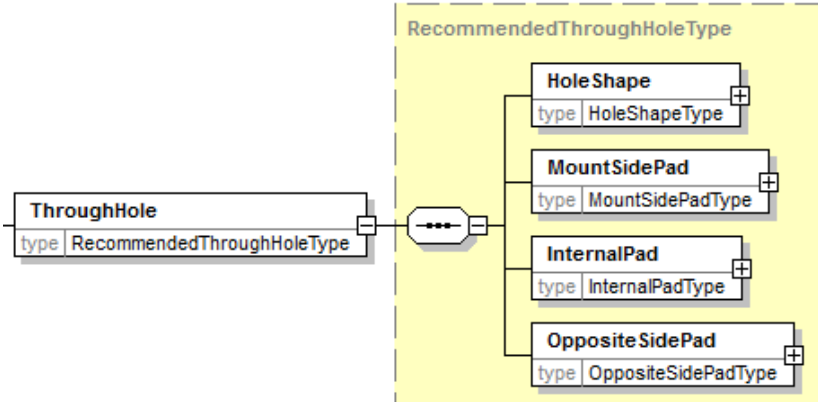
| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/RecommendedPadOrHoleShape-Array/RecommendedPadOrHoleShape/SurfaceMount/PadVoid-Array/PadVoid/PadVoidLocation/StandardArray |
| diagram | <p>The diagram illustrates the structure of the StandardArray class. It is a class with a type attribute named VoidStandardArrayType. The VoidStandardArrayType is a complex type represented by a dashed box. It contains several attributes: ny (type xs:integer), nx (type xs:integer), dx (type JEP30-D10:PitchValueSetType), dy (type JEP30-D10:PitchValueSetType), Angle (type xs:integer), and VoidGroupLowerLeftVoidCenter (type JEP30-D10:PointXYType). The StandardArray class is connected to the VoidStandardArrayType box by a line with a small square at the end, indicating a composition or aggregation relationship.</p> |
| type | VoidStandardArrayType, JEP30-D10:PitchValueSetType, JEP30-D10:PointXYType. |

5.16.2.1.1.1.2.2 Circular Array

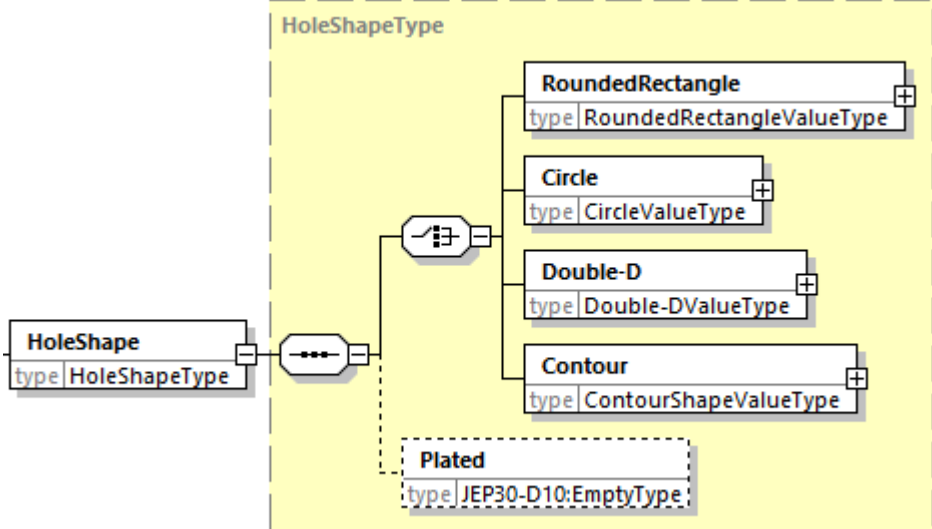
| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/RecommendedPadOrHoleShape-Array/RecommendedPadOrHoleShape/SurfaceMount/PadVoid-Array/PadVoid/PadVoidLocation/CircularArray |
| diagram |  <p>The diagram illustrates the structure of the CircularArray data type. It is defined as a sequence of VoidCircularArrayType elements, indicated by a dashed box and a sequence connector. The VoidCircularArrayType is a complex type containing several attributes: PitchRadius (type <code>xs:decimal</code>), Center (type <code>JEP30-D10:PointXYType</code>), StartAngle (type <code>xs:integer</code>), AngleToFill (type <code>xs:integer</code>), AngleBetweenVoids (type <code>xs:integer</code>), and NumberOfVoids (type <code>xs:integer</code>). Additionally, it contains a Rotation attribute of type VoidRotationType. The VoidRotationType is a complex type with two attributes: RotateWithCircle (type <code>JEP30-D10:EmptyType</code>) and RotationAngle (type <code>xs:integer</code>).</p> |
| type | JEP30-D10:PointXYType, VoidRotationType. |

See Table 22 — Void Circular Array Elements Definition for the definition of the *CircularArray* data elements.

5.16.2.1.2 Through Hole

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/RecommendedPadOrHoleShape-Array/RecommendedPadOrHoleShape/ThroughHole |
| diagram |  <p>The diagram shows a 'ThroughHole' block with a 'type' field 'RecommendedThroughHoleType'. It is connected to a dashed box labeled 'RecommendedThroughHoleType'. Inside this box, there are four sub-blocks: 'Hole Shape' (type: HoleShapeType), 'MountSidePad' (type: MountSidePadType), 'InternalPad' (type: InternalPadType), and 'Opposite SidePad' (type: OppositeSidePadType). Each sub-block has a '+' icon in its top right corner.</p> |
| type | RecommendedThroughHoleType , HoleShapeType , MountSidePadType , InternalPadType , OppositeSidePadType . |

5.16.2.1.2.1 Hole Shape

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/RecommendedPadOrHoleShape-Array/RecommendedPadOrHoleShape/ThroughHole/HoleShape |
| diagram |  <p>The diagram shows a 'HoleShape' block with a 'type' field 'HoleShapeType'. It is connected to a dashed box labeled 'HoleShapeType'. Inside this box, there are four sub-blocks: 'RoundedRectangle' (type: RoundedRectangleValueType), 'Circle' (type: CircleValueType), 'Double-D' (type: Double-DValueType), and 'Contour' (type: ContourShapeValueType). Each sub-block has a '+' icon in its top right corner. Below these sub-blocks, there is a dashed box labeled 'Plated' with a 'type' field 'JEP30-D10:EmptyType'.</p> |
| type | HoleShapeType , RoundedRectangleValueType , CircleValueType , Double-DValueType , ContourShapeValueType , JEP30-D10:EmptyType . |

The definition of each shape is outlined in Annex A (informative) Shape Definitions & Dimensions.

5.16.2.1.2.2 Mount Side Pad

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/RecommendedPadOrHoleShape-Array/RecommendedPadOrHoleShape/ThroughHole/MountSidePad |
| diagram | |
| type | MountSidePadType, RectangleValueType, RoundedRectangleValueType, CircleValueType, D-ShapeValueType, Double-DValueType, ContourShapeValueType. |

The definition of each shape is outlined in Annex A (informative) Shape Definitions & Dimensions.

5.16.2.1.2.3 Internal Pad

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/RecommendedPadOrHoleShape-Array/RecommendedPadOrHoleShape/ThroughHole/InternalPad |
| diagram | <p>The diagram illustrates the structure of the InternalPad class and its associated InternalPadType. The InternalPad class has a type attribute of type InternalPadType. This attribute points to a dashed box representing the InternalPadType enumeration. Inside this box, there are five possible shapes, each with its own value type attribute: Rectangle (RectangleValueType), RoundedRectangle (RoundedRectangleValueType), Circle (CircleValueType), Double-D (Double-DValueType), and Contour (ContourShapeValueType). Additionally, a Clearance attribute of type xs:decimal is shown within the InternalPadType box.</p> |
| type | InternalPadType, RectangleValueType, RoundedRectangleValueType, CircleValueType, D-ShapeValueType, Double-DValueType, ContourShapeValueType. |

The definition of each shape is outlined in Annex A (informative) Shape Definitions & Dimensions.

5.16.2.1.2.4 Opposite Side Pad

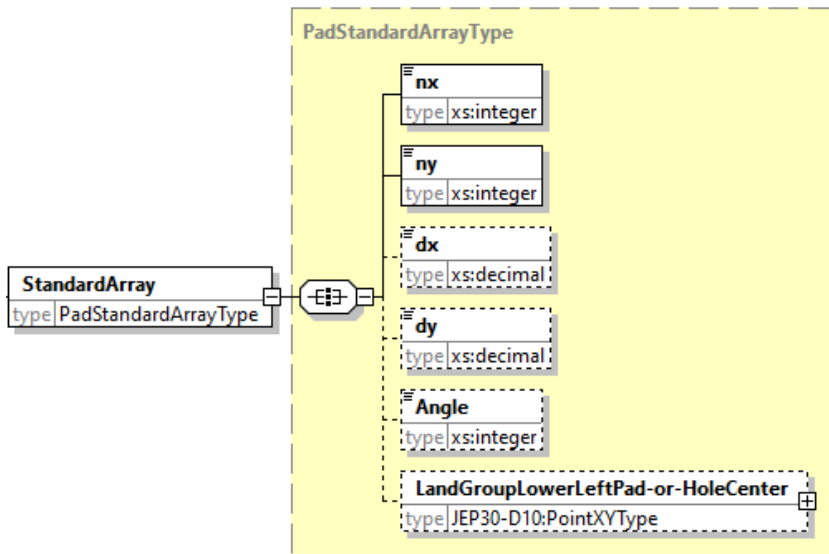
| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/RecommendedPadOrHoleShape-Array/RecommendedPadOrHoleShape/ThroughHole/OppositeSidePad |
| diagram | <p>The diagram illustrates the structure of the OppositeSidePadType. It is a dashed yellow box containing several elements. On the left, outside the box, is the OppositeSidePad element with type OppositeSidePadType. A line connects it to a connector box inside the dashed box. This connector box is linked to a group of shapes: Rectangle (type RectangleValueType), RoundedRectangle (type RoundedRectangleValueType), Circle (type CircleValueType), Double-D (type Double-DValueType), and Contour (type ContourShapeValueType). Below these shapes, also within the dashed box, is the Clearance element with type xs:decimal.</p> |
| type | OppositeSideType , RectangleHoleType , RoundedRectangleHoleType , CircleHoleType , D-ShapeHoleType , Double-DHoleType , ContourShapeHoleType . |

The definition of each shape is outlined in Annex A (informative) Shape Definitions & Dimensions.

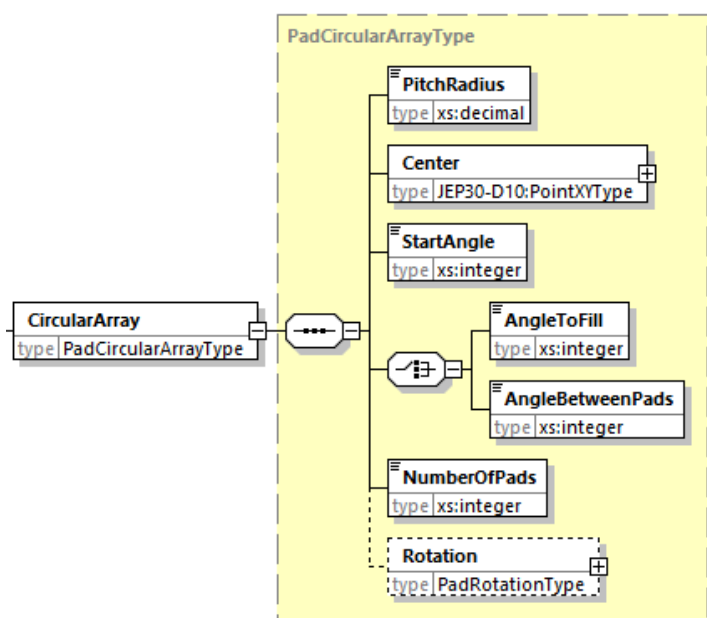
5.16.2.1.3 Location

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/RecommendedPadOrHoleShape-Array/RecommendedPadOrHoleShape/Location |
| diagram | <p>The diagram illustrates the structure of the RecommendedPadOrHoleLocationType. It is a dashed yellow box containing several elements. On the left, outside the box, is the Location element with type RecommendedPadOrHoleLocationType and a cardinality of 1..∞. A line connects it to a connector box inside the dashed box. This connector box is linked to a group of elements: ID (type xs:string), Name (type xs:string), and a group of three arrays: StandardArray (type PadStandardArrayType), CircularArray (type PadCircularArrayType), and RandomArray (type Pad-or-HoleRandomArrayType). Below these arrays, also within the dashed box, is the Pad-or-HoleStatus element with type DeletedStatusType.</p> |
| type | RecommendedPadOrHoleLocationType , PadStandardArrayType , PadCircularArrayType , DeletedStatusType , Pad-or-HoleRandomArrayType . |

5.16.2.1.3.1 Standard Array


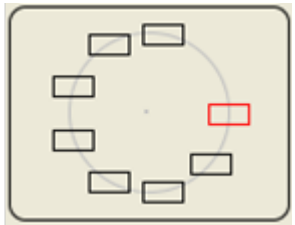
| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/RecommendedPadOrHoleShape-Array/RecommendedPadOrHoleShape/Location/StandardArray |
| diagram |  <p>The diagram shows a StandardArray box with a type attribute of PadStandardArrayType. It is connected to a dashed yellow box labeled PadStandardArrayType. Inside this box, the following properties are listed:</p> <ul style="list-style-type: none"> nx: type xs:integer ny: type xs:integer dx: type xs:decimal dy: type xs:decimal Angle: type xs:integer LandGroupLowerLeftPad-or-HoleCenter: type JEP30-D10:PointXYType |
| type | PadStandardArrayType , JEP30-D10:PointType. |

5.16.2.1.3.2 Circular Array

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/RecommendedPadOrHoleShape-Array/RecommendedPadOrHoleShape/Location/CircularArray |
| diagram |  <p>The diagram shows a CircularArray box with a type attribute of PadCircularArrayType. It is connected to a dashed yellow box labeled PadCircularArrayType. Inside this box, the following properties are listed:</p> <ul style="list-style-type: none"> PitchRadius: type xs:decimal Center: type JEP30-D10:PointXYType StartAngle: type xs:integer AngleToFill: type xs:integer AngleBetweenPads: type xs:integer NumberOfPads: type xs:integer Rotation: type PadRotationType |
| type | PadCircularArrayType , PadRotationType , JEP30-D10:PointXYType. |

5.16.2.1.3.2 Circular Array (cont'd)

Table 30 - Pad-or-Hole Circular Array Elements Definition

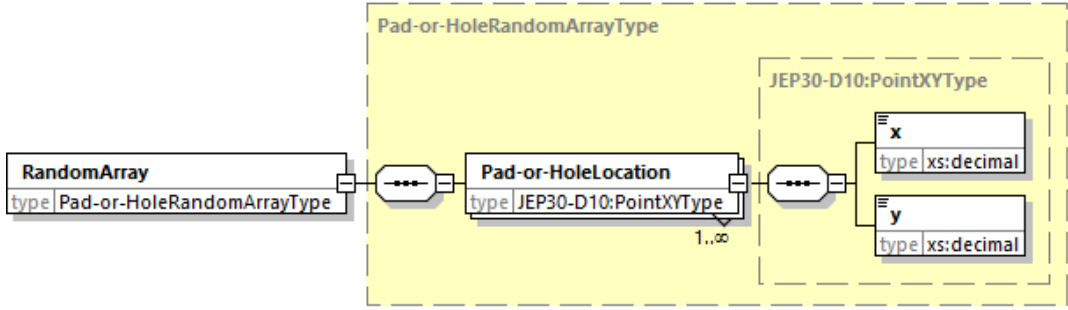
| Element | Explanation |
|---------------------|---|
| Pitch Radius | The radius of the circle of the Pads-or-Holes. |
| Center | The center of the circle of the Pads-or-Holes. |
| Start Angle | The angle at which to place the first Pad-or-Hole in the array. By default, the Pad-or-Hole is at the right-most point on the circle (i.e. the number 3 position on a clock face). This angle specifies a rotation clockwise around the circle from this point. |
| Angle to Fill | The angle through which the Pads-or-Holes are distributed, starting from the first Pad-or-Hole and continuing clockwise, until the last Pad-or-Hole is reached. |
| Angle between voids | Instead of specifying Angle to fill, you can specify the angle between the centers of each Pad-or-Hole. |
| Number of voids | The number of the Pads-or-Holes to be arrayed around the circle. The first Pad-or-Hole is displayed in red. This number includes any deleted the Pads-or-Holes in the array. |
| Rotate with Circle |  <p>If the Pads-or-Holes are rotated so that they are oriented perpendicular to the circle. This does not apply to circular or contour the Pad-or-Hole shapes.</p> |
| Rotation Angle |  <p>If the Pads-or-Holes are not oriented perpendicular to the circle, then the Pads-or-Holes could be rotated around the center of the Pad-or-Hole itself. In this image, each the Pad-or-Hole has a "0" degree rotation with respect to the "3 O'clock position". This does not apply to circular or contour Pad-or-Hole shapes.</p> |

5.16.2.1.3.3 Pad-or-Hole Status

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/RecommendedPadOrHoleShape-Array/RecommendedPadOrHoleShape/Location/Pad-or-HoleStatus |
| diagram | |
| type | DeletedStatusType , JEP30-D10:PointXYType , JEP30-D10:MinIntegerOfOneType , JEP30-D10:EmptyType . |

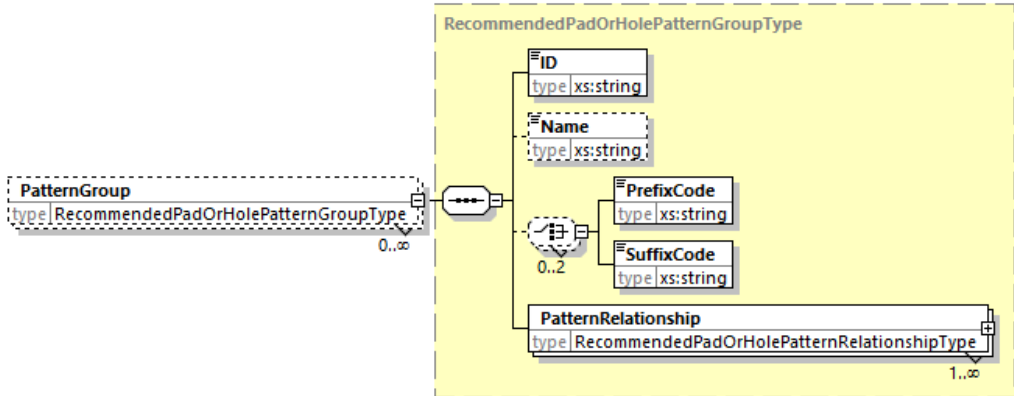
For [Pad-or-HoleLocation](#) defined via [StandardArray](#) or [CircularArray](#), some Pads-or-Holes in the array may be deleted. This is covered in the [Pad-or-HoleStatus](#) branch.

5.16.2.1.3.4 Random Array

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/RecommendedPadOrHoleShape-Array/RecommendedPadOrHoleShape/Location/RandomArray |
| diagram |  <p>The diagram illustrates the structure of the RandomArray type. It is defined as a sequence of Pad-or-HoleLocation elements. The RandomArray type is associated with the Pad-or-HoleRandomArrayType. The Pad-or-HoleLocation type is associated with the JEP30-D10:PointXYType. The JEP30-D10:PointXYType is a complex type containing two elements: x (type xs:decimal) and y (type xs:decimal). The RandomArray type is associated with the Pad-or-HoleRandomArrayType. The Pad-or-HoleLocation type is associated with the JEP30-D10:PointXYType. The JEP30-D10:PointXYType is a complex type containing two elements: x (type xs:decimal) and y (type xs:decimal). The RandomArray type is associated with the Pad-or-HoleRandomArrayType. The Pad-or-HoleLocation type is associated with the JEP30-D10:PointXYType. The JEP30-D10:PointXYType is a complex type containing two elements: x (type xs:decimal) and y (type xs:decimal).</p> |
| type | Pad-or-HoleRandomArrayType , JEP30-D10:PointType . |

[Pad-or-HoleLocation](#) can also be defined via [RandomArray](#) when there is just 1 Pad-or-Hole, or when there is no logical structure to the location of the Pads-or-Holes.

5.16.2.1.4 Pattern Groups

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/RecommendedPadOrHoleShape-Array/RecommendedPadOrHoleShape/PatternGroup |
| diagram |  <p>The diagram illustrates the structure of the PatternGroup type. It is defined as a sequence of RecommendedPadOrHolePatternGroupType elements. The PatternGroup type is associated with the RecommendedPadOrHolePatternGroupType. The RecommendedPadOrHolePatternGroupType is a complex type containing several elements: ID (type xs:string), Name (type xs:string), PrefixCode (type xs:string), SuffixCode (type xs:string), and PatternRelationship (type RecommendedPadOrHolePatternRelationshipType). The PatternGroup type is associated with the RecommendedPadOrHolePatternGroupType. The RecommendedPadOrHolePatternGroupType is a complex type containing several elements: ID (type xs:string), Name (type xs:string), PrefixCode (type xs:string), SuffixCode (type xs:string), and PatternRelationship (type RecommendedPadOrHolePatternRelationshipType). The PatternGroup type is associated with the RecommendedPadOrHolePatternGroupType. The RecommendedPadOrHolePatternGroupType is a complex type containing several elements: ID (type xs:string), Name (type xs:string), PrefixCode (type xs:string), SuffixCode (type xs:string), and PatternRelationship (type RecommendedPadOrHolePatternRelationshipType).</p> |
| type | RecommendedPadOrHolePatternGroupType , RecommendedPadOrHolePatternRelationshipType . |

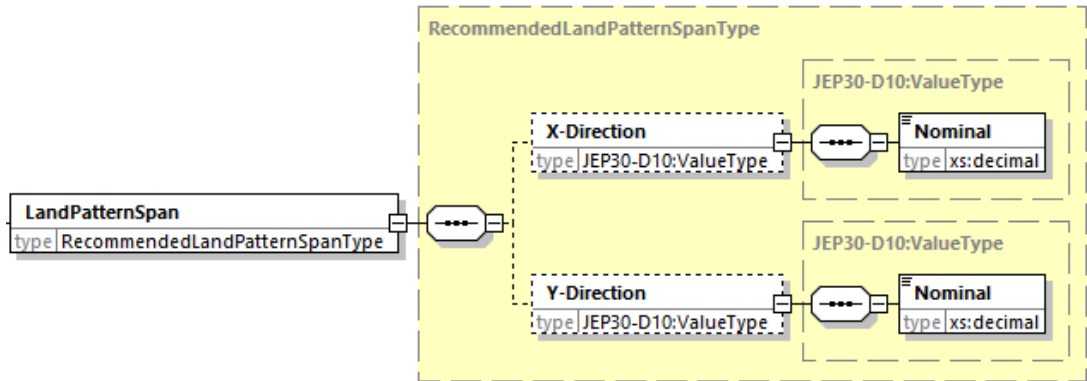
5.16.2.1.4.1 Pattern Relationship

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/RecommendedPadOrHoleShape-Array/RecommendedPadOrHoleShape/PatternGroup/PatternRelationship |
| diagram | |
| type | RecommendedPadOrHolePatternRelationshipType , RecommendedPadOrHoleRelationshipTransformationsType , TransformDuplicateType |

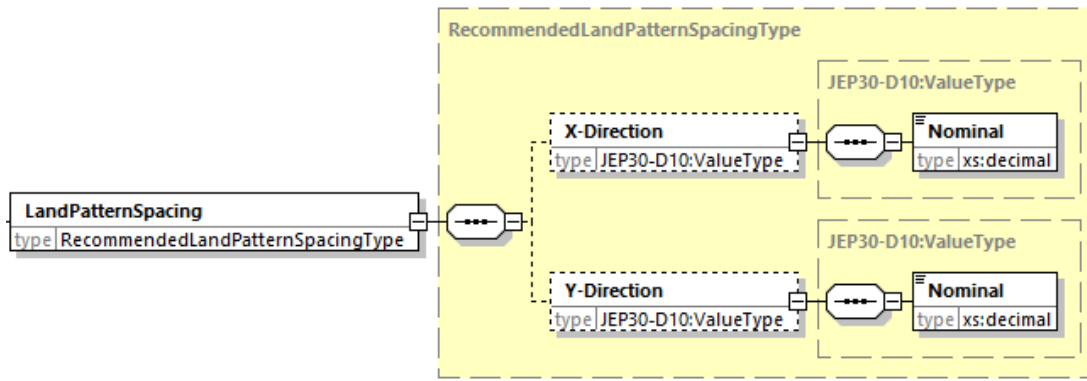
5.16.2.1.4.1.1 Relationship Transformations

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/RecommendedPadOrHoleShape-Array/RecommendedPadOrHoleShape/PatternGroup/PatternRelationship/RelationshipTransformations |
| diagram | |
| type | RecommendedPadOrHoleRelationshipTransformationsType , TransformMirrorType , TransformRotateType , JEP30-D10:PointXYType , |

5.16.2.1.5 Land Pattern Span

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/RecommendedPadOrHoleShape-Array/RecommendedPadOrHoleShape/LandPatternSpan |
| diagram |  <p>The diagram illustrates the structure of the LandPatternSpan element. It is a rectangular box with a label LandPatternSpan and a type attribute <code>type RecommendedLandPatternSpanType</code>. This box is connected via a dashed line to a larger, yellow-shaded dashed box labeled RecommendedLandPatternSpanType. Inside this yellow box, there are two sub-sections: X-Direction and Y-Direction. Each sub-section contains a dashed box with a label (X-Direction or Y-Direction) and a type attribute <code>type JEP30-D10:ValueType</code>. These are connected to a central oval shape, which is then connected to a Nominal box. The Nominal box has a type attribute <code>type xs:decimal</code>. The entire structure is enclosed in a yellow dashed box.</p> |
| type | RecommendedLandPatternSpanType , JEP30-D10:ValueType . |

5.16.2.1.6 Land Pattern Spacing

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/RecommendedPadOrHoleShape-Array/RecommendedPadOrHoleShape/LandPatternSpacing |
| diagram |  <p>The diagram illustrates the structure of the LandPatternSpacing element. It is a rectangular box with a label LandPatternSpacing and a type attribute <code>type RecommendedLandPatternSpacingType</code>. This box is connected via a dashed line to a larger, yellow-shaded dashed box labeled RecommendedLandPatternSpacingType. Inside this yellow box, there are two sub-sections: X-Direction and Y-Direction. Each sub-section contains a dashed box with a label (X-Direction or Y-Direction) and a type attribute <code>type JEP30-D10:ValueType</code>. These are connected to a central oval shape, which is then connected to a Nominal box. The Nominal box has a type attribute <code>type xs:decimal</code>. The entire structure is enclosed in a yellow dashed box.</p> |
| type | RecommendedLandPatternSpacingType , JEP30-D10:ValueType . |

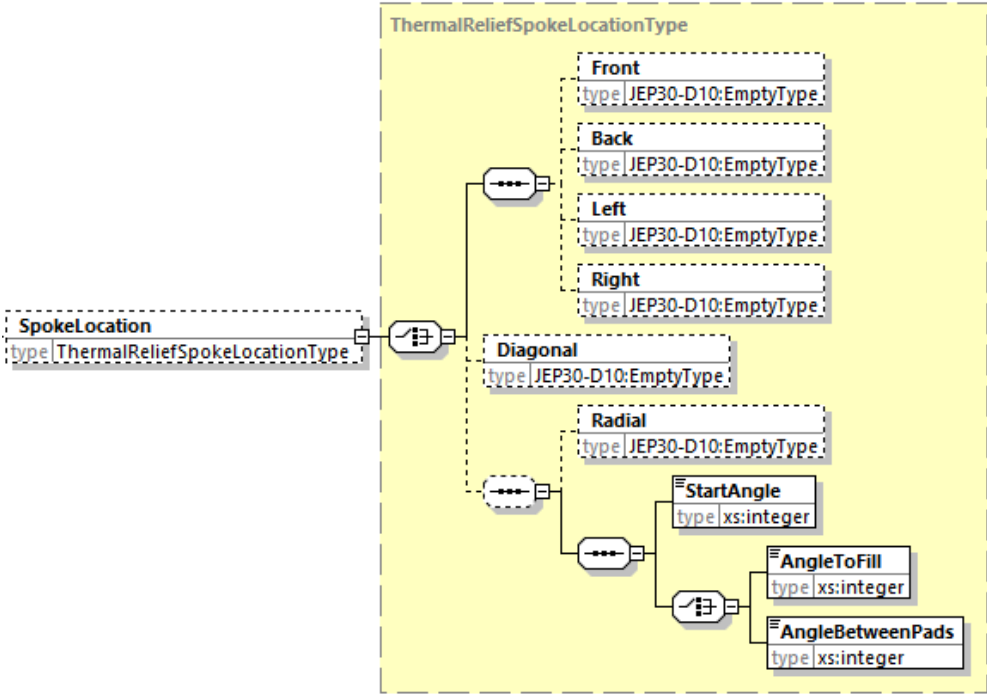
5.16.2.2 Pad Group To Pad Group Relationship

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/RecommendedPadOrHoleShape-Array/ PadGroupToPadGroupRelationship |
| diagram | |
| type | PadGroupToPadGroupRelationshipType , LandPatternSpanType , LandPatternSpacingType . |

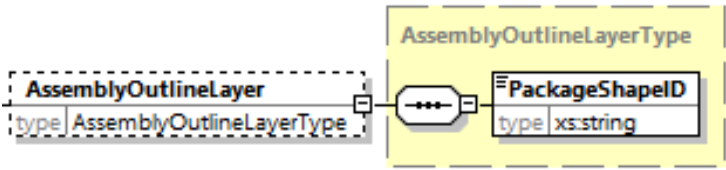
5.16.3 Thermal Relief - Array

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/ThermalRelief-Array |
| diagram | |
| type | ThermalRelief-ArrayType , ThermalReliefType , ThermalReliefSpokeLocationType |

5.16.3.1 Spoke Location

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/ThermalRelief-Array/ThermalRelief/SpokeLocation |
| diagram |  <p>The diagram illustrates the structure of the ThermalReliefSpokeLocationType. It shows a tree-like hierarchy starting from a 'SpokeLocation' node (type ThermalReliefSpokeLocationType) which connects to a 'ThermalReliefSpokeLocationType' container. This container branches into 'Front', 'Back', 'Left', 'Right', 'Diagonal', and 'Radial' nodes, all of type JEP30-D10:EmptyType. The 'Radial' node further branches into 'StartAngle' (type xs:integer) and a sub-container that branches into 'AngleToFill' (type xs:integer) and 'AngleBetweenPads' (type xs:integer).</p> |
| type | ThermalReliefSpokeLocationType, JEP30-D10:EmptyType. |

5.16.4 Assembly Outline Layer

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/AssemblyOutlineLayer |
| diagram |  <p>The diagram illustrates the structure of the AssemblyOutlineLayerType. It shows a tree-like hierarchy starting from an 'AssemblyOutlineLayer' node (type AssemblyOutlineLayerType) which connects to an 'AssemblyOutlineLayerType' container. This container branches into a 'PackageShapeID' node (type xs:string).</p> |
| type | AssemblyOutlineLayerType. |

5.16.5 Conductive Area - Array

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/ConductiveArea-Array |
| diagram | |
| type | ConductiveArea-ArrayType, ConductiveAreaType, RectangleValueType, ModifiedRectangleValueType, CircleValueType, JEP30-D10:PointXYType, ContourShapeValueType, ConductiveAreaVoid-ArrayType. |

5.16.5.1 Conductive Area Void - Array

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/ConductiveArea-Array/ConductiveAreaVoid-Array |
| diagram | |
| type | ConductiveAreaVoid-ArrayType, ConductiveAreaVoidType, ConductiveAreaVoidShapeType, ConductiveAreaVoidLocationType. |

5.16.5.1.1 Conductive Area Void Shape

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/ConductiveArea-Array/ConductiveAreaVoid-Array/ConductiveAreaVoid/ConductiveAreaVoidShape |
| diagram | |
| type | ConductiveAreaVoidShapeType, RectangleValueType, RoundedRectangleValueType, ModifiedRectangleValueType, CircleValueType, Double-DValueType, ContourShapeValueType. |

5.16.5.1.2 Conductive Area Void Location

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/ConductiveArea-Array/ConductiveAreaVoid-Array/ConductiveAreaVoid/ConductiveAreaVoidLocation |
| diagram | |
| type | ConductiveAreaVoidLocationType, VoidStandardArrayType, VoidCircularArrayType, DeletedStatusType, VoidRandomArrayType. |

5.16.6 Placement Outline

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/PlacementOutline |
| diagram | |
| type | PlacementOutlineType, RectangleValueType, ModifiedRectangleValueType, CircleValueType, JEP30-D10:PointXYType, ContourShapeValueType |

5.16.7 Keep-in Layer - Array

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/Keep-inLayer-Array |
| diagram | <pre> classDiagram class Keep_inLayer_Array["Keep-inLayer-Array"] { type Keep-inLayer-ArrayType } class Keep_inLayer["Keep-inLayer"] { type Keep-inLayerType } class Restrictive_Layer["RestrictiveLayer"] { type Keep-inRestrictiveLayerType } class Restriction["Restriction"] { type Keep-inLayerRestrictionType } class Keep_inLayer_Restriction_Type["Keep-inLayerRestrictionType"] { type Keep-inLayerRestrictionType } class Contour["Contour"] { type ContourShapeValueType } class Shape_Center["ShapeCenter"] { type JEP30-D10:PointXYType } class Rectangle["Rectangle"] { type RectangleValueType } class Modified_Rectangle["ModifiedRectangle"] { type ModifiedRectangleValueType } class Circle["Circle"] { type CircleValueType } Keep_inLayer_Array "1" -- "1..∞" Keep_inLayer Keep_inLayer "1" -- "1" Restrictive_Layer Restrictive_Layer "1" -- "1" Restriction Restriction "1" -- "1" Keep_inLayer_Restriction_Type Keep_inLayer "1" -- "0..∞" Shape_Center Shape_Center "1" -- "1" Rectangle Shape_Center "1" -- "1" Modified_Rectangle Shape_Center "1" -- "1" Circle </pre> |
| type | Keep-inLayer-ArrayType, Keep-inLayerType, RectangleValueType, ModifiedRectangleValueType, CircleValueType, JEP30-D10:PointXYType, ContourShapeValueType, Keep-inRestrictiveLayerType, Keep-inLayerRestrictionType. |

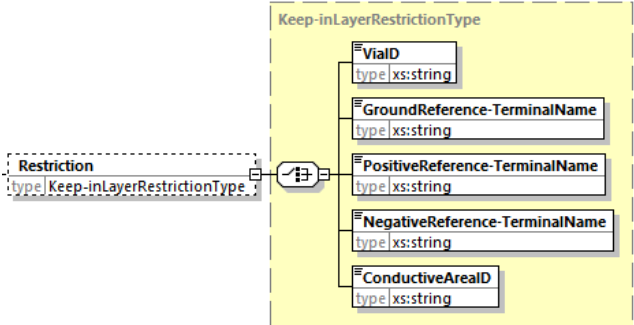
5.16.7.1 Restrictive Layer

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/Keep-inLayer-Array/Keep-inLayer/RestrictiveLayer |
| diagram | <p>The diagram illustrates the structure of the RestrictiveLayer type. It is composed of several nested and associated components:</p> <ul style="list-style-type: none"> RestrictiveLayer (type: <code>Keep-inRestrictiveLayerType</code>) is the root element. It contains an InnerLayer (type: <code>Keep-inRestrictiveInnerLayerType</code>). The InnerLayer is further defined by a Keep-inRestrictiveInnerLayerType container, which includes: <ul style="list-style-type: none"> MountSide (type: <code>JEP30-D10:EmptyType</code>) OppositeSide (type: <code>JEP30-D10:EmptyType</code>) AllLayers (type: <code>JEP30-D10:EmptyType</code>) FromOuterLayer (type: <code>RestrictiveInnerLayerFromOuterLayerType</code>) GroundReference-TerminalName (type: <code>xs:string</code>) PositiveReference-TerminalName (type: <code>xs:string</code>) NegativeReference-TerminalName (type: <code>xs:string</code>) |
| type | Keep-inRestrictiveLayerType , JEP30-D10:EmptyType , Keep-inRestrictiveInnerLayerType , RestrictiveInnerLayerFromOuterLayerType |

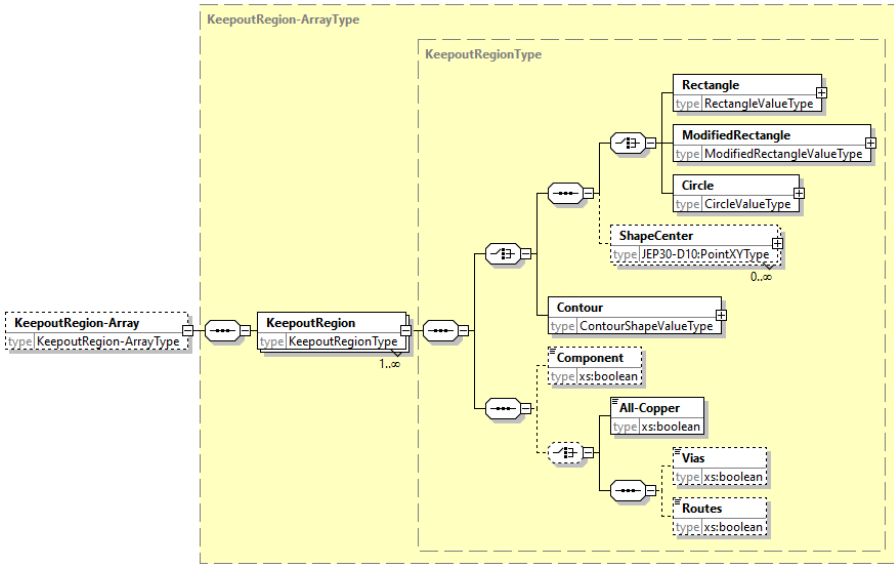
5.16.7.2 From Outer Layer

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/Keep-inLayer-Array/Keep-inLayer/RestrictiveLayer/Keep-inRestrictiveInnerLayer/FromOuterLayer |
| diagram | <p>The diagram illustrates the structure of the FromOuterLayer type. It is composed of several nested and associated components:</p> <ul style="list-style-type: none"> FromOuterLayer (type: <code>RestrictiveInnerLayerFromOuterLayerType</code>) is the root element. It contains a RestrictiveInnerLayerFromOuterLayerType container, which includes: <ul style="list-style-type: none"> MountSide (type: <code>JEP30-D10:EmptyType</code>) OppositeSide (type: <code>JEP30-D10:EmptyType</code>) LayerLevel (type: <code>xs:integer</code>) |
| type | RestrictiveInnerLayerFromOuterLayerType , JEP30-D10:EmptyType |

5.16.7.3 Restriction

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/Keep-inLayer-Array/Keep-inLayer/Restriction |
| diagram |  <p>The diagram shows the structure of the Keep-inLayerRestrictionType. It is a dashed box containing several elements: VialID (type xs:string), GroundReference-TerminalName (type xs:string), PositiveReference-TerminalName (type xs:string), NegativeReference-TerminalName (type xs:string), and ConductiveAreaID (type xs:string). A Restriction element (type Keep-inLayerRestrictionType) is shown on the left, connected to the main structure by a dashed line.</p> |
| type | Keep-inLayerRestrictionType . |

5.16.8 Keepout Region - Array

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/KeepoutRegion-Array |
| diagram |  <p>The diagram shows the structure of the KeepoutRegion-Array. It is a dashed box containing several elements: KeepoutRegion-Array (type KeepoutRegion-ArrayType), KeepoutRegion (type KeepoutRegionType), Rectangle (type RectangleValueType), ModifiedRectangle (type ModifiedRectangleValueType), Circle (type CircleValueType), ShapeCenter (type JEP30-D10:PointXYType), Contour (type ContourShapeValueType), Component (type xs:boolean), All-Copper (type xs:boolean), Vias (type xs:boolean), and Routes (type xs:boolean). The KeepoutRegion element is connected to the main structure by a dashed line.</p> |
| type | KeepoutRegionArray , KeepoutRegionType , ContourShapeType , JEP30-D10:PointType . |

Keepout are sometimes required around Parts to ensure the proper functionality of the Part. The definition of the contour shape is outlined in Annex A (informative) Shape Dimensions. The various type of Keepouts are:

- Component
- All-Copper
- Vias
- Routes

5.16.9 Soldermask Layer - Array

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/SoldermaskLayer-Array |
| diagram | |
| type | SoldermaskLayer-ArrayType , SoldermaskLayerType , SoldermaskGangRelief-ArrayType , SM-ShapeType , SoldermaskLayerToSoldermaskLayerRelationshipType . |

5.16.9.1 Soldermask Gang Relief- Array

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/SoldermaskLayer-Array/SoldermaskLayer/SoldermaskGangRelief-Array |
| diagram | |
| type | SoldermaskGangRelief-ArrayType , SoldermaskGangReliefType , TerminalCenter-ArrayType , TerminalIndexType , TerminalNumberType |

5.16.9.1.1 Terminal Center - Array

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/SoldermaskLayer-Array/SoldermaskLayer/SoldermaskGangRelief-Array/SoldermaskGangRelief/TerminalCenter-Array |
| diagram | |
| type | TerminalCenter-ArrayType, TerminalCenterType, JEP30-D10:PointXYType. |

5.16.9.2 Soldermask Shape

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/SoldermaskLayer-Array/SoldermaskLayer/SoldermaskShape |
| diagram | <p>The diagram illustrates the XSD structure for the SoldermaskShape element. The SoldermaskShape element is of type SM-ShapeType and has a cardinality of 1..∞. It is connected to a dashed box representing the SM-ShapeType complex type. This complex type includes an ID attribute (type xs:string), a choice of shape types (Rectangle, RoundedRectangle, HalfRoundedRectangle, ModifiedRectangle, Circle, D-Shape, RoundedRectangleD-Shape, Double-D, Contour), an SM-ShapeLocation element (type SM-ShapeLocationType, 1..∞), and a choice of pattern types (PatternGroup, SM-ShapePatternSpan, SM-ShapePatternSpacing).</p> |
| type | SM-ShapeType, RectangleValueType, RoundedRectangleValueType, HalfRoundedRectangleValueType, ModifiedRectangleValueType, CircleValueType, D-ShapeValueType, D-ShapeRoundedRectangleValueType, Double-DValueType, ContourShapeValueType, SM-ShapeLocationType, SM-ShapePatternGroupType, SM-ShapePatternSpanType, SM-ShapePatternSpacingType. |

5.16.9.2.1 SM - Shape Location

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/SoldermaskLayer-Array/SoldermaskLayer/SoldermaskShape/SM-ShapeLocation |
| diagram | <p>The diagram illustrates the structure of the SM-ShapeLocationType. It is a sequence of elements: ID (type xs:string), Name (type xs:string), and a choice between StandardArray (type SM-ShapeStandardArrayType), CircularArray (type SM-ShapeCircularArrayType), SM-ShapeStatus (type SM-ShapeStatusType), and RandomArray (type SM-ShapeRandomArrayType). The SM-ShapeLocation element is shown with a cardinality of 1..∞.</p> |
| type | SM-ShapeLocationType, SM-ShapeStandardArrayType, SM-ShapeCircularArrayType, SM-ShapeStatusType, SM-ShapeRandomArrayType. |

5.16.9.2.1.1 Standard Array

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/SoldermaskLayer-Array/SoldermaskLayer/SoldermaskShape/SM-ShapeLocation |
| diagram | <p>The diagram illustrates the structure of the SM-ShapeStandardArrayType. It is a sequence of elements: nx (type xs:integer), ny (type xs:integer), dx (type xs:decimal), dy (type xs:decimal), Angle (type xs:integer), and SM-ShapeGroupLowerLeftCenter (type JEP30-D10:PointXYType). The StandardArray element is shown with a cardinality of 1.</p> |
| type | SM-ShapeStandardArrayType, JEP30-D10:PointXYType. |

5.16.9.2.1.2 Circular Array

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/SoldermaskLayer-Array/SoldermaskLayer/SoldermaskShape/SM-ShapeLocation/CircularArray |
| diagram | |
| type | SM-ShapeCircularArrayType , JEP30-D10:PointXYType , SM-ShapeRotationType , JEP30-D10:EmptyType . |

5.16.9.2.1.3 SM - Shape Status

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/SoldermaskLayer-Array/SoldermaskLayer/SoldermaskShape/SM-ShapeLocation/ShapeStatus |
| diagram | |
| type | SM-ShapeStatusType , SM-ShapeCenter-ArrayType , SM-ShapeIndexType , JEP30-D10:EmptyType . |

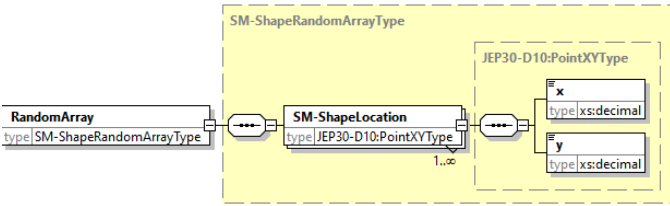
5.16.9.2.1.3.1 SM – Shape Center - Array

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/SoldermaskLayer-Array/SoldermaskLayer/SoldermaskShape/SM-ShapeLocation/ShapeStatus/SM-ShapeCenter-Array |
| diagram | <p>The diagram illustrates the structure of the SM-ShapeCenter-Array. It features a class SM-ShapeCenter-ArrayType which contains an array of SM-ShapeCenterType objects. The SM-ShapeCenterType class is defined with the following attributes:</p> <ul style="list-style-type: none">x: type xs:decimaly: type xs:decimalPitchRadius: type xs:decimalCenter: type JEP30-D10:PointXYTypeAngle: type xs:integer <p>The SM-ShapeCenter-ArrayType class is shown with a multiplicity of 0..∞. The SM-ShapeCenterType class is shown with a multiplicity of 1. The Center attribute is shown with a multiplicity of 1. The Angle attribute is shown with a multiplicity of 1.</p> |
| type | SM-ShapeCenter-ArrayType , SM-ShapeCenterType , JEP30-D10:PointXYType . |

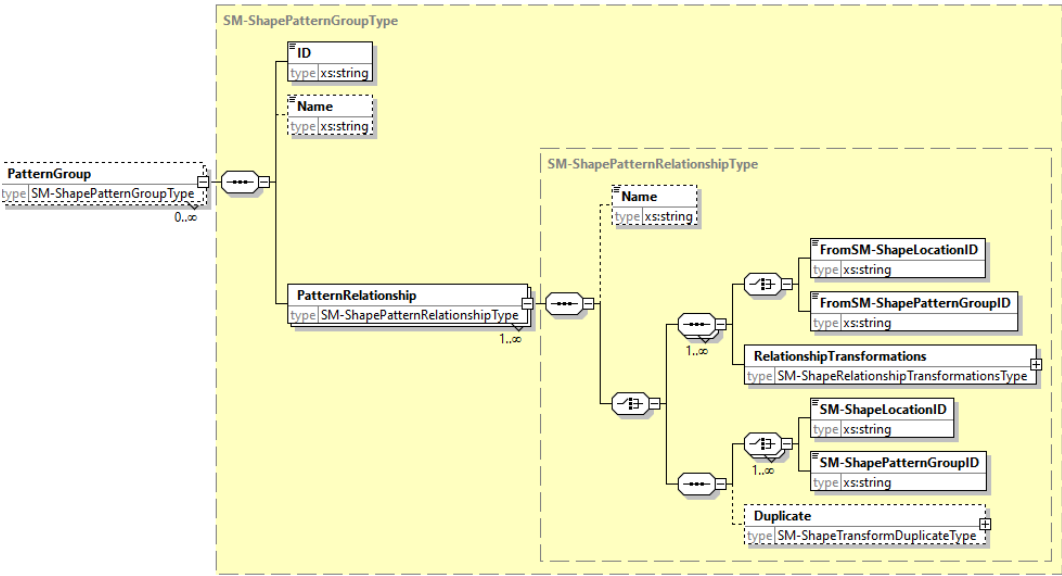
5.16.9.2.1.3.2 SM - Shape Index

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/SoldermaskLayer-Array/SoldermaskLayer/SoldermaskShape/SM-ShapeLocation/ShapeStatus/ShapeIndex |
| diagram | <pre> classDiagram class SMShapeIndex { type SMShapeIndexType } class Center { type JEP30-D10:PointXYType } class RowTerminalIndex { type JEP30-D10:MinIntegerOfOneType minIncl/maxIncl 1 } class FromRowTerminalIndex { type JEP30-D10:MinIntegerOfOneType minIncl/maxIncl 1 } class ToRowTerminalIndex { type JEP30-D10:MinIntegerOfOneType minIncl/maxIncl 1 } class ColumnTerminalIndex { type JEP30-D10:MinIntegerOfOneType minIncl/maxIncl 1 } class FromColumnTerminalIndex { type JEP30-D10:MinIntegerOfOneType minIncl/maxIncl 1 } class ToColumnTerminalIndex { type JEP30-D10:MinIntegerOfOneType minIncl/maxIncl 1 } class PolarTerminalIndex { type JEP30-D10:MinIntegerOfOneType minIncl/maxIncl 1 } class FromPolarTerminalIndex { type JEP30-D10:MinIntegerOfOneType minIncl/maxIncl 1 } class ToPolarTerminalIndex { type JEP30-D10:MinIntegerOfOneType minIncl/maxIncl 1 } SMShapeIndex "1" -- "1..∞" Center SMShapeIndex "1" -- "1..∞" RowTerminalIndex SMShapeIndex "1" -- "1..∞" ColumnTerminalIndex SMShapeIndex "1" -- "1..∞" PolarTerminalIndex RowTerminalIndex "1..∞" -- "1" FromRowTerminalIndex RowTerminalIndex "1..∞" -- "1" ToRowTerminalIndex ColumnTerminalIndex "1..∞" -- "1" FromColumnTerminalIndex ColumnTerminalIndex "1..∞" -- "1" ToColumnTerminalIndex PolarTerminalIndex "1..∞" -- "1" FromPolarTerminalIndex PolarTerminalIndex "1..∞" -- "1" ToPolarTerminalIndex </pre> |
| type | SM-ShapeIndexType, JEP30-D10:PointXYType, JEP30-D10:MinIntegerOfOneType, JEP30-D10:EmptyType. |

5.16.9.2.1.4 Random Array

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/SoldermaskLayer-Array/SoldermaskLayer/SoldermaskShape/SM-ShapeLocation/RandonArray |
| diagram |  <p>The diagram shows the structure of SM-ShapeRandomArrayType. It consists of a RandomArray element (type SM-ShapeRandomArrayType) connected to a SM-ShapeLocation element (type JEP30-D10:PointXYType) with a cardinality of 1..∞. The SM-ShapeLocation element is further connected to a JEP30-D10:PointXYType element, which contains two decimal fields: x (type xs:decimal) and y (type xs:decimal).</p> |
| type | SM-ShapeRandomArrayType , JEP30-D10:PointXYType . |

5.16.9.2.2 Pattern Group

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/SoldermaskLayer-Array/SoldermaskLayer/SoldermaskShape/PatternGroup |
| diagram |  <p>The diagram shows the structure of SM-ShapePatternGroupType. It consists of a PatternGroup element (type SM-ShapePatternGroupType) with a cardinality of 0..∞, connected to a PatternRelationship element (type SM-ShapePatternRelationshipType) with a cardinality of 1..∞. The PatternRelationship element is further connected to a SM-ShapePatternRelationshipType element, which contains several sub-elements: ID (type xs:string), Name (type xs:string), FromSM-ShapeLocationID (type xs:string), FromSM-ShapePatternGroupID (type xs:string), RelationshipTransformations (type SM-ShapeRelationshipTransformationsType), SM-ShapeLocationID (type xs:string), SM-ShapePatternGroupID (type xs:string), and Duplicate (type SM-ShapeTransformDuplicateType).</p> |
| type | SM-ShapePatternGroupType , SM-ShapePatternRelationshipType , SM-ShapeRelationshipTransformationsType , SM-ShapeTransformDuplicateType . |

5.16.9.2.2.1 RelationshipTransformations

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/SoldermaskLayer-Array/SoldermaskLayer/SoldermaskShape/PatternGroup/ RelationshipTransformations |
| diagram | |
| type | SM-ShapeRelationshipTransformationsType |

5.16.9.2.2.2 RelationshipTransformations

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/SoldermaskLayer-Array/SoldermaskLayer/SoldermaskShape/PatternGroup/Duplicate |
| diagram | |
| type | SM-ShapeTransformDuplicateType. |

5.16.9.2.3 SM – Shape Pattern Span

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/SoldermaskLayer-Array/SoldermaskLayer/SoldermaskShape/SM-ShapePatternSpan |
| diagram | <pre> classDiagram class SMShapePatternSpan { type SMShapePatternSpanType } class SMShapePatternSpanType { X-Direction type JEP30-D10:ValueType Y-Direction type JEP30-D10:ValueType } SMShapePatternSpan --> SMShapePatternSpanType </pre> |
| type | SM-ShapePatternSpanType , JEP30-D10:ValueType . |

5.16.9.2.4 SM – Shape Pattern Spacing

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/SoldermaskLayer-Array/SoldermaskLayer/SoldermaskShape/SM-ShapePatternSpacing |
| diagram | <pre> classDiagram class SMShapePatternSpacing { type SMShapePatternSpacingType } class SMShapePatternSpacingType { X-Direction type JEP30-D10:ValueType Y-Direction type JEP30-D10:ValueType } SMShapePatternSpacing --> SMShapePatternSpacingType </pre> |
| type | SM-ShapePatternSpacingType , JEP30-D10:ValueType . |

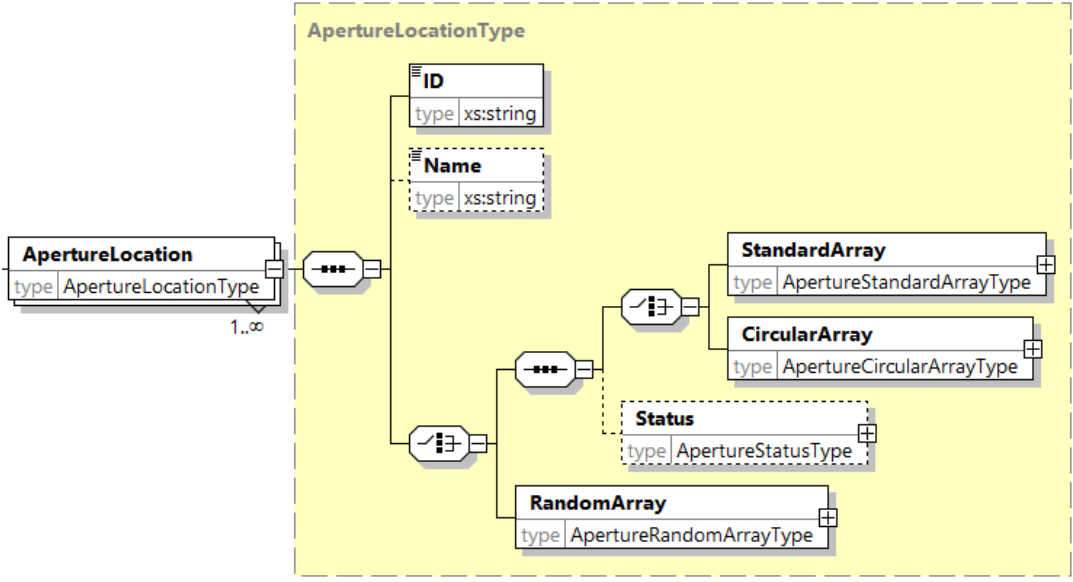
5.16.10 Pastemask Layer- Array

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/PastemaskLayer-Array |
| diagram | |
| type | PasteMaskLayer-ArrayType, RecommendedStencilThicknessType, JEP30-D10:MinNomMaxValueSetType, StencilThicknessUOMType, PasteMaskLayerType, ApertureShapeType, PastemaskLayerToPastemaskLayerRelationshipType. |

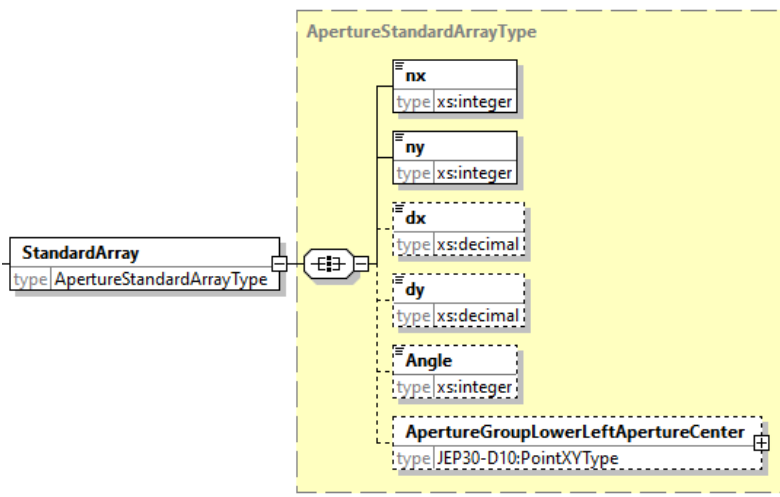
5.16.10.1 Aperture Shape

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/PastemaskLayer-Array/PastemaskLayer/ApertureShape |
| diagram | <p>The diagram illustrates the XSD structure for the ApertureShape element. The root element is ApertureShapeType, which is a complex type. It contains the following elements and attributes:</p> <ul style="list-style-type: none"> ID: A required attribute of type <code>xs:string</code>. Rectangle: An optional element of type <code>RectangleValueType</code>. RoundedRectangle: An optional element of type <code>RoundedRectangleValueType</code>. ModifiedRectangle: An optional element of type <code>ModifiedApertureRectangleType</code>. Circle: An optional element of type <code>CircleValueType</code>. Contour: An optional element of type <code>ContourShapeValueType</code>. ApertureThicknessType: A choice element containing: <ul style="list-style-type: none"> Step-upStep-downApertureThickness: An optional element of type <code>ApertureThicknessType</code>. Thickness: An optional element of type <code>JEP30-D10:MinNomMaxValueType</code>. ApertureThicknessUOM: An optional element of type <code>StencilThicknessUOMType</code>. ApertureLocation: An optional element of type <code>ApertureLocationType</code>. PatternGroup: An optional element of type <code>AperturePatternGroupType</code>. AperturePatternSpan: An optional element of type <code>AperturePatternSpanType</code>, which contains a choice of: <ul style="list-style-type: none"> X-Direction: An optional element of type <code>JEP30-D10:ValueType</code>. Y-Direction: An optional element of type <code>JEP30-D10:ValueType</code>. AperturePatternSpacing: An optional element of type <code>AperturePatternSpacingType</code>, which contains a choice of: <ul style="list-style-type: none"> X-Direction: An optional element of type <code>JEP30-D10:ValueType</code>. Y-Direction: An optional element of type <code>JEP30-D10:ValueType</code>. <p>The ApertureShape element is shown with a cardinality of <code>1..∞</code>.</p> |
| type | ApertureShapeType, RectangleValueType, RoundedRectangleValueType, ModifiedApertureRectangleType, CircleValueType, ContourShapeValueType, ApertureThicknessType, JEP30-D10:MinNomMaxValueType, StencilThicknessUOMType, ApertureLocationType, AperturePatternGroupType, AperturePatternSpanType, AperturePatternSpacingType, JEP30-D10:ValueType. |

5.16.10.1.1 Aperture Location

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/PastemaskLayer-Array/PastemaskLayer/ApertureShape/ApertureLocation |
| diagram |  <p>The diagram illustrates the structure of the ApertureLocationType. It is a complex type containing several elements:</p> <ul style="list-style-type: none"> ID: A required element of type <code>xs:string</code>. Name: A required element of type <code>xs:string</code>. StandardArray: An optional element of type <code>ApertureStandardArrayType</code>. CircularArray: An optional element of type <code>ApertureCircularArrayType</code>. Status: An optional element of type <code>ApertureStatusType</code>. RandomArray: An optional element of type <code>ApertureRandomArrayType</code>. <p>The ApertureLocation element is shown as a container for the ApertureLocationType, with a cardinality of 1..∞.</p> |
| type | ApertureLocationType , ApertureStandardArrayType , ApertureCircularArrayType , ApertureStatusType , ApertureRandomArrayType . |

5.16.10.1.1.1 Standard Array

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/PastemaskLayer-Array/PastemaskLayer/ApertureShape/ApertureLocation/StandardArray |
| diagram |  <p>The diagram illustrates the structure of the ApertureStandardArrayType. It is a complex type containing several elements:</p> <ul style="list-style-type: none"> nx: A required element of type <code>xs:integer</code>. ny: A required element of type <code>xs:integer</code>. dx: A required element of type <code>xs:decimal</code>. dy: A required element of type <code>xs:decimal</code>. Angle: A required element of type <code>xs:integer</code>. ApertureGroupLowerLeftApertureCenter: An optional element of type <code>JEP30-D10:PointXYType</code>. <p>The StandardArray element is shown as a container for the ApertureStandardArrayType, with a cardinality of 1..1.</p> |
| type | ApertureStandardArrayType , JEP30-D10:PointXYType . |

5.16.10.1.1.2 Circular Array

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/PastemaskLayer-Array/PastemaskLayer/ApertureShape/ApertureLocation/CircularArray |
| diagram | <pre> xsd:element name="CircularArray" type="ApertureCircularArrayType"> <div> <div>PitchRadius</div> <div>type xs:decimal</div> </div> <div> <div>Center</div> <div>type JEP30-D10:PointXYType</div> </div> <div> <div>StartAngle</div> <div>type xs:integer</div> </div> <div> <div>AngleToFill</div> <div>type xs:integer</div> </div> <div> <div>AngleBetweenPads</div> <div>type xs:integer</div> </div> <div> <div>NumberOfApertures</div> <div>type xs:integer</div> </div> <div> <div>ApertureRotation</div> <div>type ApertureRotationType</div> </div> </pre> |

ApertureRotationType

RotateWithCircle

type JEP30-D10:EmptyType

RotationAngle

type xs:integer

5.16.10.1.1.3 Status

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/PastemaskLayer-Array/PastemaskLayer/ApertureShape/ApertureLocation/Status |
| diagram | <p>The diagram illustrates the structure of ApertureStatusType. It is a tree structure where the root is Status (type: ApertureStatusType). Status branches into Deleted (type: JEP30-D10:EmptyType) and Center (type: JEP30-D10:PointXYType). The Center node branches into RowTerminalIndex, ColumnTerminalIndex, and PolarTerminalIndex. Each of these branches into From and To sub-indices. All indices are of type JEP30-D10:MinIntegerOfOneType. Multiplicities are indicated by numbers in circles: 1..∞ for the main branches and 1 for the sub-indices.</p> |
| type | ApertureStatusType, JEP30-D10:MinIntegerOfOneType, JEP30-D10:PointXYType, JEP30-D10:EmptyType. |

5.16.10.1.1.4 RandomArray

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/PastemaskLayer-Array/PastemaskLayer/ApertureShape/ApertureLocation/RandomArray |
| diagram | <p>The diagram illustrates the structure of ApertureRandomArrayType. It is a tree structure where the root is RandomArray (type: ApertureRandomArrayType). RandomArray branches into Location (type: JEP30-D10:PointXYType). The Location node has a multiplicity of 1..∞.</p> |
| type | ApertureRandomArrayType, JEP30-D10:PointXYType. |

5.16.10.1.2 Pattern Groups

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/PastemaskLayer-Array/PastemaskLayer/ApertureShape/PatternGroup |
| diagram | |
| type | AperturePatternGroupType , AperturePatternRelationshipType , AperturePatternRelationshipTransformationsType , ApertureTransformDuplicateType . |

5.16.10.1.2.1 Relationship Transformations

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/PastemaskLayer-Array/PastemaskLayer/ApertureShape/PatternGroup/PatternRelationship/RelationshipTransformations |
| diagram | |
| type | AperturePatternRelationshipTransformationsType , TransformMirrorType , TransformRotateType , JEP30-D10:PointXYType |

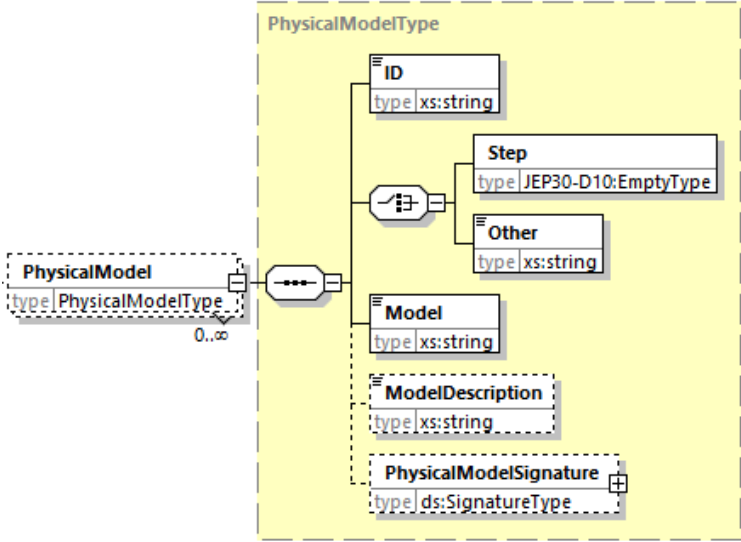
5.16.10.1.2.2 Duplicate

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/PastemaskLayer-Array/PastemaskLayer/ApertureShape/PatternGroup/PatternRelationship/Duplicate |
| diagram | <p>The diagram illustrates the structure of the ApertureTransformDuplicateType. It shows a dashed box representing the type, which contains a solid box for the ApertureTransformDuplicate element. This element is connected to a container that holds three sub-elements: nx (type xs:integer), ny (type xs:integer), and a group containing dx (type xs:decimal) and dy (type xs:decimal).</p> |
| type | ApertureTransformDuplicateType . |

5.16.10.2 Pastemask Layer To Pastemask Layer Relationship

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/RecommendedFootprint-Array/RecommendedFootprint/PastemaskLayer-Array/PastemaskLayer/ApertureGroupToApertureGroupRelationship |
| diagram | <p>The diagram illustrates the structure of the ApertureGroupToApertureGroupRelationshipType. It shows a dashed box representing the type, which contains a solid box for the ApertureGroupToApertureGroupRelationship element. This element is connected to a container that holds three sub-elements: FromApertureShapeID (type xs:string), ToApertureShapeID (type xs:string), and a group containing ApertureGroupSpan (type ApertureGroupSpanType) and ApertureGroupSpacing (type ApertureGroupSpacingType). The ApertureGroupSpan element is further connected to a container that holds X-Direction (type xs:decimal) and Y-Direction (type xs:decimal). The ApertureGroupSpacing element is further connected to a container that holds X-Direction (type xs:decimal) and Y-Direction (type xs:decimal).</p> |
| type | ApertureGroupToApertureGroupRelationshipType , ApertureGroupSpanType , ApertureGroupSpacingType . |

5.17 Physical Model

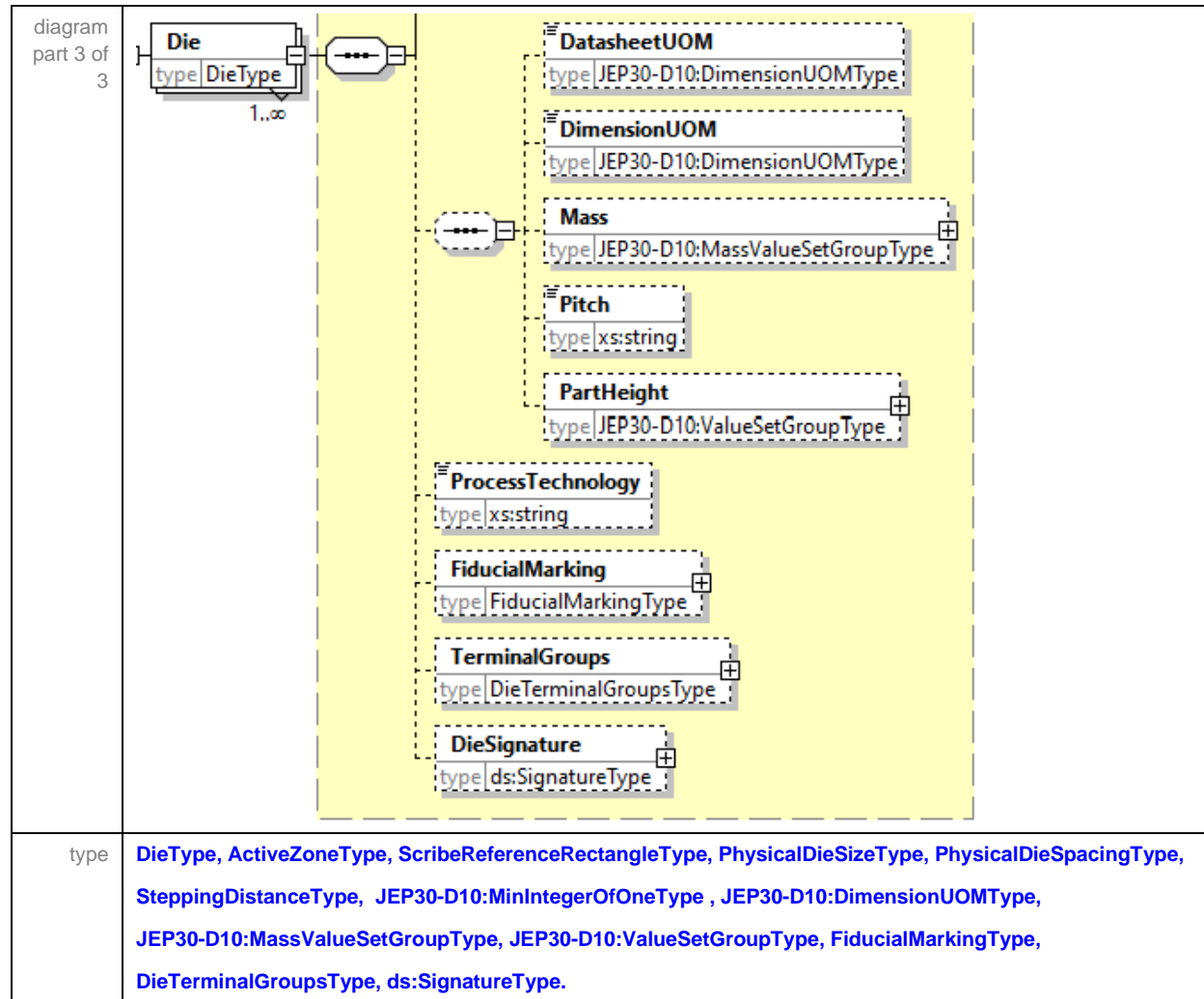
| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/PhysicalModel |
| diagram |  <p>The diagram illustrates the structure of the PhysicalModelType. It is a dashed box containing several elements: ID (type xs:string), Step (type JEP30-D10:EmptyType), Other (type xs:string), Model (type xs:string), ModelDescription (type xs:string), and PhysicalModelSignature (type ds:SignatureType). A PhysicalModel element (type PhysicalModelType) is shown outside the dashed box, connected to the PhysicalModelType box by a line with a multiplicity of 0..∞.</p> |
| type | PhysicalModelType, ModelType, JEP30-D10:EmptyType, ds:SignatureType. |

One or more references to external physical model files can be defined by the *PhysicalModel* section. The *Model* element can refer to either a file name of a file that is provided together with the corresponding JEP30 xml archive or a URL definition. An *Other* file format is accommodated to account for non-standard formats where it is advised that the optional *ModelDescription* element is used to describe the format and its intended usage.

6 Package Section - Die-Array

| | |
|---------------------------|---|
| path | PartModel/PackageSection |
| diagram part 1 of 3 | <p>This diagram shows the structure of the PackageSection type. It is defined as <code>JEP30-P101:PackageSectionType</code>. The structure includes a Package-Array (type <code>Package-ArrayType</code>) and a Die-Array (type <code>Die-ArrayType</code>). The Die-Array contains a Die (type <code>DieType</code>) with a cardinality of <code>1..∞</code>. A constraints element is also shown. The entire structure is enclosed in a dashed box labeled <code>JEP30-P101:PackageSectionType</code>.</p> |
| type | PackageSectionType, Package-ArrayType, Die-ArrayType, DieType. |
| path | PartModel/PackageSection/Die-Array/Die |
| diagram part 2 of 3 | <p>This diagram shows the structure of the DieType. It is defined as <code>DieType</code>. The structure includes an ID (type <code>xs:string</code>), an ActiveZone (type <code>ActiveZoneType</code>), a ScribeReferenceRectangle (type <code>ScribeReferenceRectangleType</code>), a PhysicalDieSize (type <code>PhysicalDieSizeType</code>), a PhysicalDieSpacing (type <code>PhysicalDieSpacingType</code>), a SteppingDistance (type <code>SteppingDistanceType</code>), an ExtendedTerminalCount (type <code>JEP30-D10:MinIntegerOfOneType</code>, <code>minIncl/maxIncl</code> <code>1</code>), a TerminalCount (type <code>JEP30-D10:MinIntegerOfOneType</code>, <code>minIncl/maxIncl</code> <code>1</code>), a DeletedTerminalCount (type <code>xs:integer</code>), and a MissingTerminalCount (type <code>xs:integer</code>). The entire structure is enclosed in a dashed box labeled <code>DieType</code>.</p> |

6 Package Section - Die-Array (cont'd)



The definitions of the [ExtendedTerminalCount](#), [TerminalCount](#), [DeletedTerminalCount](#) and [MissingTerminalCount](#) are described in the JESD30 document. The Die/Pitch follows the same rules as Package/Pitch as defined in the JESD30 document.

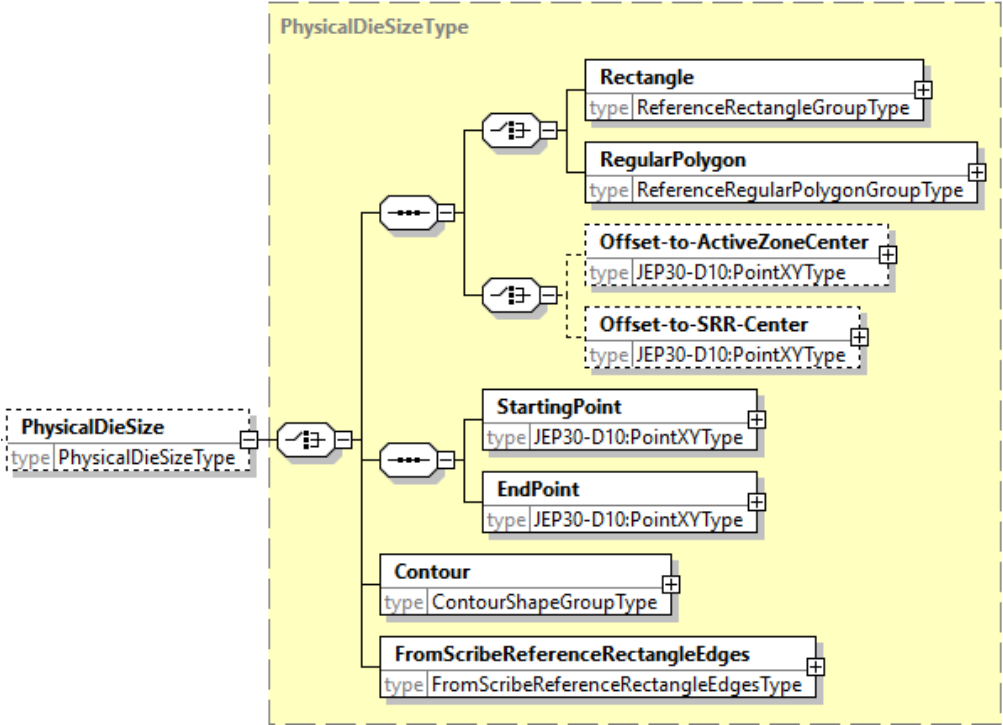
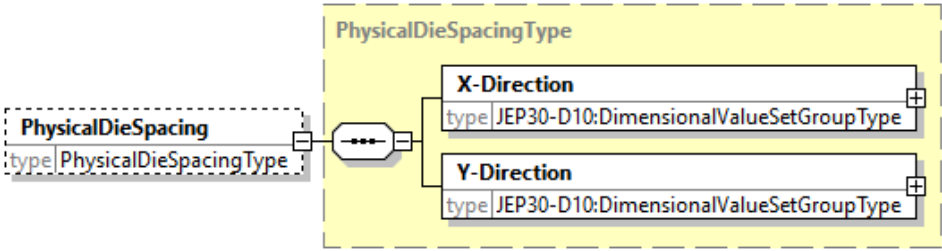
The enumerate values of the [JEP30-D10:DimensionUOMType](#) are [nm](#), [um](#), [mm](#), [m](#), [in](#) and [mil](#), while [nm](#), [um](#), [mm](#), are the units of measure used in the die's.

The component Manufacturer can provide details the [ProcessTechnology](#) that was used for the die fabrication, such as 5nm, or company specific internal code name.

6.1 Die Zones

| | | | |
|---------------------------|---|--|--|
| path | PartModel/PackageSection/Die-Array/Die/ActiveZone | | |
| diagram part 1 of 4 | | | |
| type | ActiveZoneType, ReferenceRectangleGroupType, ReferenceRegularPolygonGroupType, D10:PointXYType, ContourShapeGroupType. | | |
| path | PartModel/PackageSection/Die-Array/Die/ScribeReferenceRectangle | | |
| diagram part 2 of 4 | | | |
| type | ScribeReferenceRectangleType, ReferenceRectangleGroupType, ReferenceRegularPolygonGroupType, JEP30-D10:PointXYType, ContourShapeGroupType, FromActiveZoneEdgesType. | | |

6.1 Die Zones (cont'd)

| | | |
|---------------------------|--|--|
| path | PartModel/PackageSection/Die-Array/Die/PhysicalDieSize | |
| diagram part 3 of 4 |  <p>The diagram shows the structure of PhysicalDieSize. It is a dashed box containing a solid box labeled PhysicalDieSize with the type <code>PhysicalDieSizeType</code>. This box is connected to a central node, which then branches into several other boxes: Rectangle (type <code>ReferenceRectangleGroupType</code>), RegularPolygon (type <code>ReferenceRegularPolygonGroupType</code>), Offset-to-ActiveZoneCenter (type <code>JEP30-D10:PointXYType</code>), Offset-to-SRR-Center (type <code>JEP30-D10:PointXYType</code>), StartingPoint (type <code>JEP30-D10:PointXYType</code>), EndPoint (type <code>JEP30-D10:PointXYType</code>), Contour (type <code>ContourShapeGroupType</code>), and FromScribeReferenceRectangleEdges (type <code>FromScribeReferenceRectangleEdgesType</code>). The entire structure is enclosed in a yellow dashed box labeled PhysicalDieSizeType.</p> | |
| type | PhysicalDieSizeType, ReferenceRectangleGroupType, ReferenceRegularPolygonGroupType, JEP30-D10:PointXYType, ContourShapeGroupType, FromScribeReferenceRectangleEdgesType. | |
| path | PartModel/PackageSection/Die-Array/Die/PhysicalDieSizeSpacing | |
| diagram part 4 of 4 |  <p>The diagram shows the structure of PhysicalDieSizeSpacing. It is a dashed box containing a solid box labeled PhysicalDieSizeSpacing with the type <code>PhysicalDieSizeSpacingType</code>. This box is connected to a central node, which then branches into two boxes: X-Direction (type <code>JEP30-D10:DimensionalValueSetGroupType</code>) and Y-Direction (type <code>JEP30-D10:DimensionalValueSetGroupType</code>). The entire structure is enclosed in a yellow dashed box labeled PhysicalDieSizeSpacingType.</p> | |
| type | PhysicalDieSizeSpacingType, JEP30-D10:DimensionalValueSetGroupType. | |

The centers of the [ActiveZone](#), [ScribeReferenceRectangle](#), [PhysicalDieSize](#) do not have to be at the same point. Each can have different offsets with respect to the [ActiveZone](#) center. The center of the active die is the primary datum.

The distance between the edge of the [ActiveZone](#) to the edge of the [ScribeReferenceRectangle](#) is defined as the [SealRing](#) for that side. The [SealRing](#) is sometimes referred to as the “Etch Ring” or “Guard Ring”. The [SealRing](#) can be a different dimensions on each of the 4 sides, although in most cases, the right side equals the left side and the top side equals the bottom side, but the top/bottom side values does not typically equal to the right/left side values.

6.1 Die Zones (cont'd)

The distance between the edge of the *ScribeReferenceRectangle* to the edge of the *PhysicalDieSize* is defined as the *RemnantScribe* for that side. The *RemnantScribe* can also be a different dimensions on each of the 4 sides, similar to that of the *SealRing*.

The *PhysicalDieSizeSpacing* is the physical distance between each *PhysicalDieSize* on the wafer. This *PhysicalDieSizeSpacing* can be a different dimension in the X and Y directions.

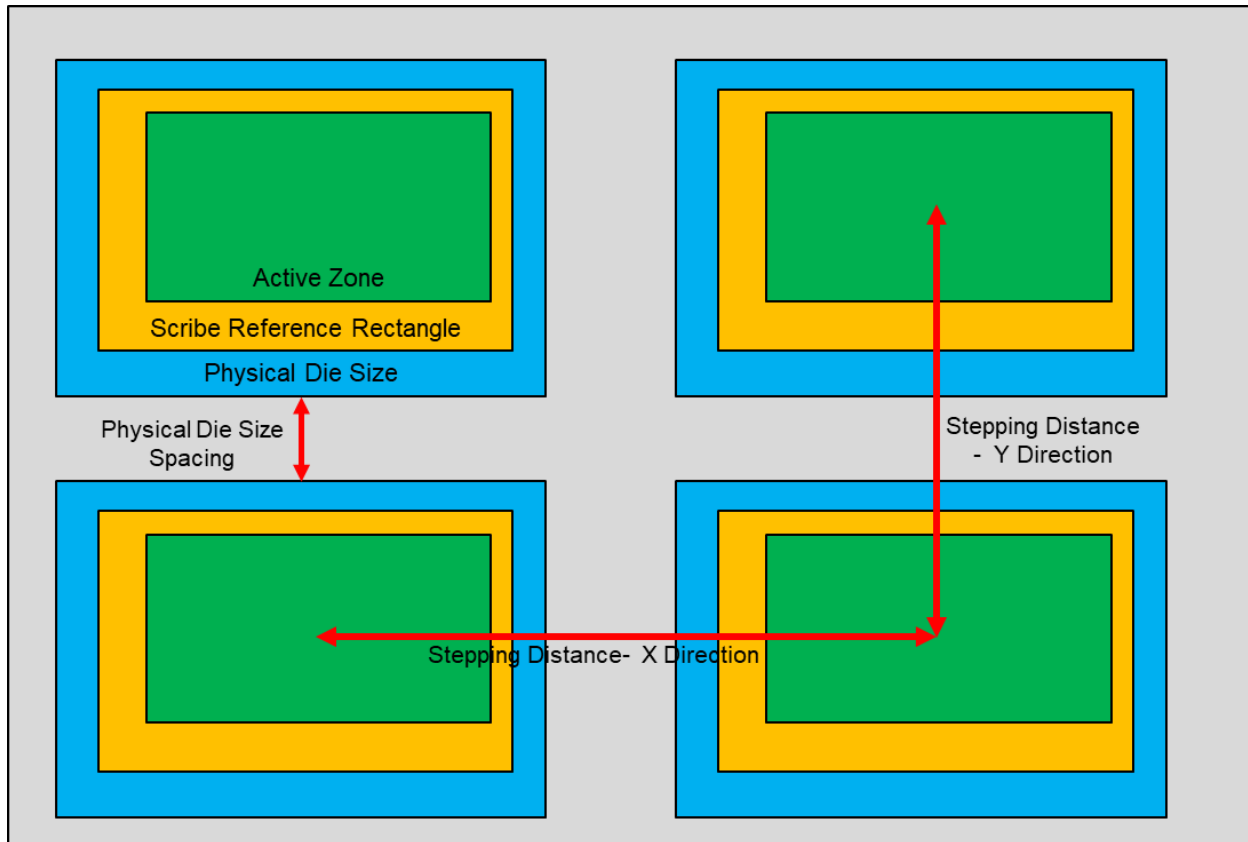


Figure 19 - Zone shapes for Physical Dies on a Wafer

6.2 Stepping Distance

| | |
|---------|---|
| path | PartModel/PackageSection/Die-Array/Die/SteppingDistance |
| diagram | |
| type | SteppingDistanceType . |

As shown in Figure 19 above, the *SteppingDistance* can be a different value in the X and Y direction. This is sometimes referred to as the “Wafer Pitch”.

6.3 Mass

| | |
|---------|--|
| path | PartModel/PackageSection/Die-Array/Die/Mass |
| diagram | |
| type | JEP30-D10:MassValueSetGroupType, MassUOMType |
| group | ValueSetGroup |

The *ValueSetGroup* is widely used throughout the Schema. It provides for all the following combinations in a structured way and enables the xml file to have all of the following elements directly underneath its parent element to which it is assigned to. :

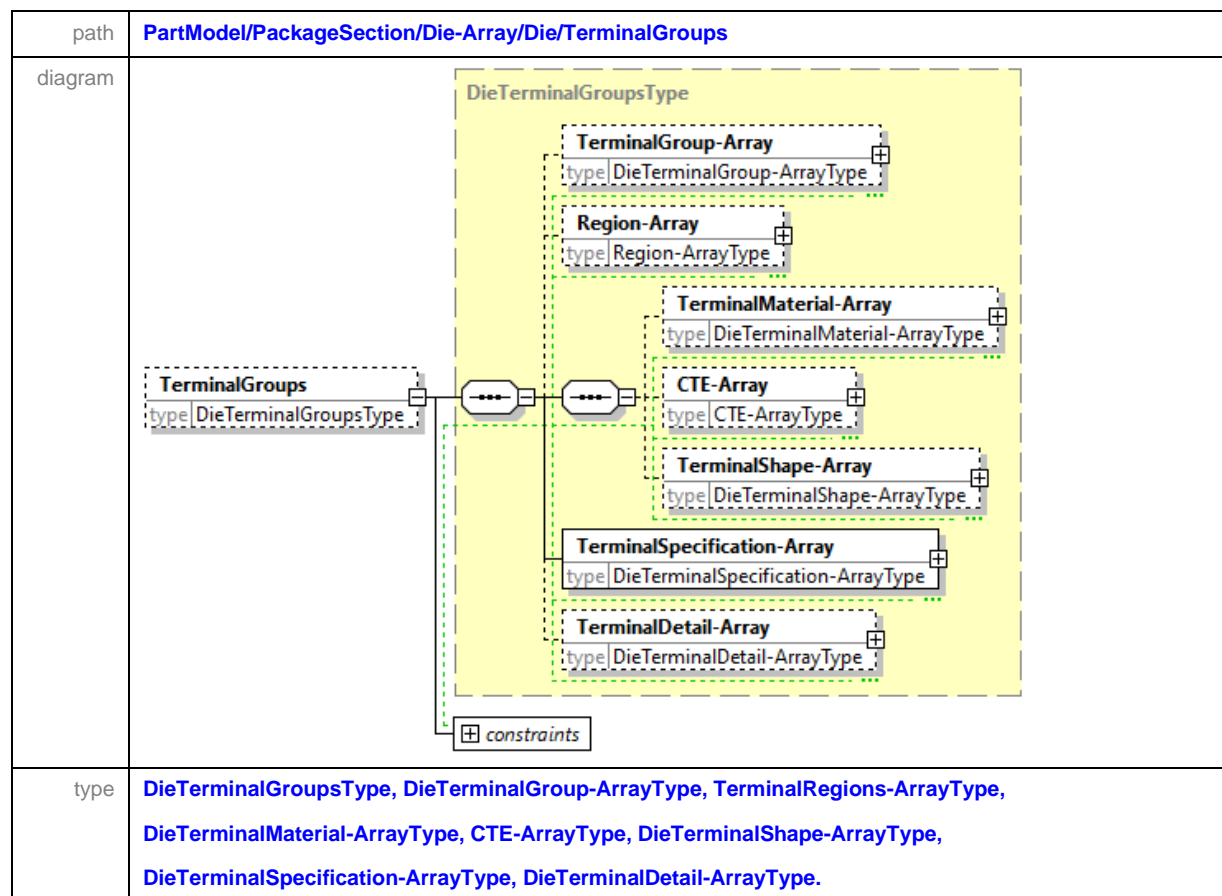
- 1) Nominal,
- 2) Positive and Negative Tolerances, or Total tolerance based on the same UOM or based on a percentage of the nominal value, and
- 3) Minimum and or Maximum.

6.4 Part Height

| | |
|---------|---|
| path | PartModel/PackageSection/Die-Array/Die/Mass |
| diagram | |
| type | JEP30-D10:ValueSetGroupType. |
| group | ValueSetGroup |

PartHeight for a *Die* represents the total height of the die part from its seating plane.

6.5 Terminal Groups



Let's assume that we have a complex array of 12 instantiations of a UCle function organized in two columns and 6 rows as shown in Figure 20 below. Let's define these 12 instantiations as Bank 1.

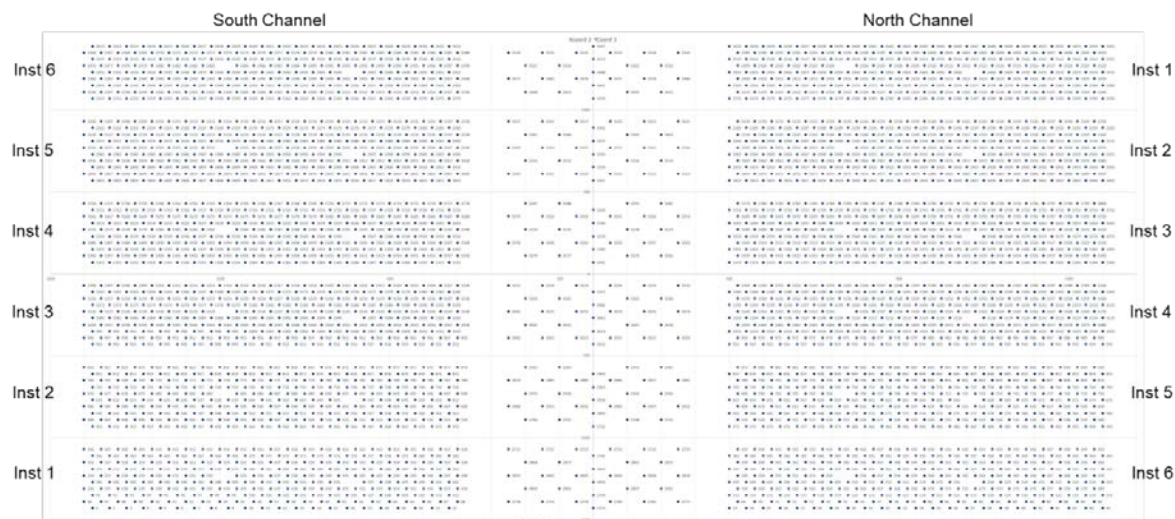


Figure 20 - Complex Array of 12 instantiations of a UCle Function

6.5 Terminal Groups (cont'd)

In the South channel, Inst 1 is duplicated 6 times in the Y direction. Then the entirety of the South channel is mirrored in both the x axis and y axis and offset to the right to create a duplicate North Channel.

There are many ways in which terminal numbering can be applied to this set of instantiations from maintaining a common numbering pattern for each instantiation or defining One contiguous sequential numbering pattern that covers all the terminals in Bank 1.

Let's also assume that each respective terminal location within each instantiation (the respective location is taken prior to any instantiations being mirrored), has the same StandardTerminalName, then, the assignment of properties, functions, terminal groupings to data/addresses busses, differential pairs, or any other logical groupings, only needs to be defined for one instantiation, and inherited by all other 11 instantiations.

The following sections will explore this example in more detail, to demonstrate the efficient compression of data, so that the resultant xml file remains as compact as possible.

6.5.1 Terminal Group - Array

| | |
|---------|--|
| path | PartModel/PackageSection/Die-Array/Die/TerminalGroups/TerminalGroup-Array |
| diagram | <pre> classDiagram class TerminalGroupArray { type DieTerminalGroup-ArrayType } class DieTerminalGroupArrayType { TerminalGroup DieTerminalGroupType 1..∞ Relationship-to-ActiveZone Relationship-to-ActiveZoneType 0..∞ TerminalGroupToTerminalGroupRelationships DieTerminalGroupToTerminalGroupRelationshipsType 1..∞ Coplanarity xs:decimal } TerminalGroupArray --> DieTerminalGroupArrayType </pre> |
| type | DieTerminalGroup-ArrayType , Relationship-to-ActiveZoneType , DieTerminalGroupToTerminalGroupRelationshipsType |

6.5.1.1 Terminal Group

| | |
|---------|---|
| path | PartModel/PackageSection/Die-Array/Die/TerminalGroups/TerminalGroup-Array/TerminalGroup |
| diagram | |
| type | DieTerminalGroupType, DieTerminalPositionType, BottomUpperPositionConfigurationType, DieTerminalPattern-ArrayType, DieTerminalPatternType, DieTerminalPatternGroupType. |

6.5.1.1.1 Terminal Pattern

| | |
|---------|---|
| path | PartModel/PackageSection/Die-Array/Die/TerminalGroups/TerminalGroup-Array/TerminalGroup/ TerminalPattern-Array/TerminalPattern |
| diagram | <pre> classDiagram class TerminalPattern { type DieTerminalPatternType } class DieTerminalPatternType { ID type xs:string Name type xs:string } class DieStandardArrayType { nx type xs:integer ny type xs:integer dx type xs:decimal dy type xs:decimal Angle type xs:integer TerminalPatternLowerLeftTerminalCenter type JEP30-D10:PointXYType } class CircularArrayType { type CircularArrayType } class RandomArrayType { type RandomArrayType } class JEP30D10PointXYType { type JEP30-D10:PointXYType } TerminalPattern "1..∞" -- "*" DieTerminalPatternType DieTerminalPatternType -- "*" DieStandardArrayType DieTerminalPatternType -- "*" CircularArrayType DieTerminalPatternType -- "*" RandomArrayType DieStandardArrayType -- "*" JEP30D10PointXYType </pre> <p>The diagram illustrates the structure of the TerminalPattern class and its associated types. The TerminalPattern class has a type attribute of DieTerminalPatternType and is associated with DieTerminalPatternType (1..∞ to *). The DieTerminalPatternType class has two attributes: ID (type xs:string) and Name (type xs:string). It is associated with DieStandardArrayType (* to *), CircularArrayType (* to *), and RandomArrayType (* to *). The DieStandardArrayType class has six attributes: nx (type xs:integer), ny (type xs:integer), dx (type xs:decimal), dy (type xs:decimal), Angle (type xs:integer), and TerminalPatternLowerLeftTerminalCenter (type JEP30-D10:PointXYType). The CircularArrayType class has a type attribute of CircularArrayType. The RandomArrayType class has a type attribute of RandomArrayType. The JEP30-D10:PointXYType class has a type attribute of JEP30-D10:PointXYType.</p> |
| type | DieTerminalPatternType, DieStandardArrayType, JEP30-D10:PointXYType, CircularArrayType, RandomArrayType |

6.5.1.1.1 Terminal Pattern (cont'd)

A more detailed examination of a single UCle instantiation shows several distinct layout patterns of terminals, as shown by the various colored patterns in Figure 21 and Figure 22 below.

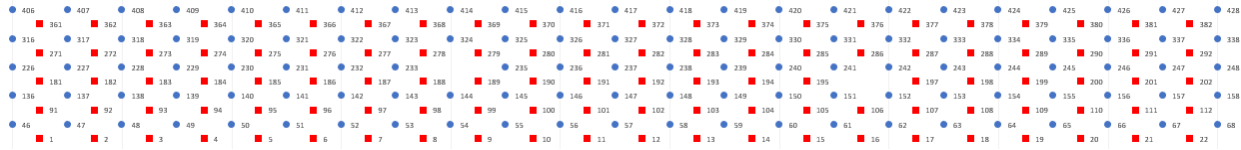


Figure 21 - Single UCle Instantiation in South Channel Inst 1



Figure 22 - Power Nodes for South Channel Inst 1 & 2

Each pattern can be represented by the standard array as shown in Table 31 below and in the subsequent xml structure.

Table 31 - Pattern Details for the UCle Instantiation

| Terminal Pattern ID | nx | ny | dx | dy | Missing | |
|----------------------|----|----|-----|-----|-----------|-----------|
| | | | | | Col Index | Row Index |
| 1 (Blue circles) | 22 | 5 | 50 | 80 | 16 | 3 |
| 2 (Red squares) | 23 | 5 | 50 | 80 | 9 | 3 |
| 3 (Gold diamonds) | 3 | 3 | 100 | 160 | None | None |
| 4 (Black triangles) | 2 | 2 | 100 | 160 | None | None |
| 5 (Purple asterisks) | 1 | 5 | 0 | 80 | None | None |
| 6 (Orange circles) | 3 | 2 | 100 | 160 | None | None |
| 7 (Blue squares) | 2 | 3 | 100 | 160 | None | None |

Note that some of the terminals are missing from the pattern ID's 1 and 2 which will be addressed in 6.5.7 Terminal Detail - Array below.

The VSS terminals as shown by Terminal pattern ID 5 (Purple asterisks) are shared between the South and North channels.

The VCCIO terminals as identified by Terminal patterns 3 (Gold diamonds) and 4 (Black triangles) for Instantiation 1 and Terminal patterns 6 (Orange circles) and 7 (Blue squares) for Instantiation 2 for the base of the repetitive pattern when applied to the rest of Bank 1.

6.5.1.1.1 Terminal Pattern (cont'd)

There may be additional VSS and VCCIO terminals intermingled throughout Terminal patterns 1 and 2, but there are treated as part of those patterns. Below is the xml representation of these Terminal Patterns.

```
<TerminalPattern-Array>
  <TerminalPattern>
    <ID>Terminal Pattern ID 1</ID>
    <Name>South Inst 1 Base</Name>
    <StandardArray>
      <nx>22</nx>
      <ny>5</ny>
      <dx>50.00</dx>
      <dy>80.00</dy>
      <TerminalPatternLowerLeftTerminalCenter>
        <x>-1475.00</x>
        <y>-1430.00</y>
      </TerminalPatternLowerLeftTerminalCenter>
    </StandardArray>
  </TerminalPattern>
  <TerminalPattern>
    <ID>Terminal Pattern ID 2</ID>
    <Name>South Inst 1 Staggered</Name>
    <StandardArray>
      <nx>23</nx>
      <ny>5</ny>
      <dx>50.00</dx>
      <dy>80.00</dy>
      <TerminalPatternLowerLeftTerminalCenter>
        <x>-1500.00</x>
        <y>-1390.00</y>
      </TerminalPatternLowerLeftTerminalCenter>
    </StandardArray>
  </TerminalPattern>
  <TerminalPattern>
    <ID>Terminal Pattern ID 3</ID>
    <Name>South Inst 1 VCCIO Base</Name>
    <StandardArray>
      <nx>3</nx>
      <ny>3</ny>
      <dx>100.00</dx>
      <dy>160.00</dy>
      <TerminalPatternLowerLeftTerminalCenter>
        <x>-250.00</x>
        <y>-1390.00</y>
      </TerminalPatternLowerLeftTerminalCenter>
    </StandardArray>
  </TerminalPattern>
  <TerminalPattern>
    <ID>Terminal Pattern ID 4</ID>
    <Name>South Inst 1 VCCIO Staggered</Name>
    <StandardArray>
      <nx>2</nx>
      <ny>2</ny>
      <dx>100.00</dx>
      <dy>160.00</dy>
      <TerminalPatternLowerLeftTerminalCenter>
        <x>-200.00</x>
        <y>-1310.00</y>
      </TerminalPatternLowerLeftTerminalCenter>
    </StandardArray>
  </TerminalPattern>
</TerminalPattern-Array>
```

6.5.1.1.1 Terminal Pattern (cont'd)

```

</TerminalPattern>
<TerminalPattern>
  <ID>Terminal Pattern ID 5</ID>
  <Name>VSS Center Inst 1</Name>
  <StandardArray>
    <nx>1</nx>
    <ny>5</ny>
    <dx>0</dx>
    <dy>80.00</dy>
    <TerminalPatternLowerLeftTerminalCenter>
      <x>0</x>
      <y>-1430.00</y>
    </TerminalPatternLowerLeftTerminalCenter>
  </StandardArray>
</TerminalPattern>
<TerminalPattern>
  <ID>Terminal Pattern ID 6</ID>
  <Name>South Inst 2 VCCIO Base</Name>
  <StandardArray>
    <nx>2</nx>
    <ny>3</ny>
    <dx>100.00</dx>
    <dy>160.00</dy>
    <TerminalPatternLowerLeftTerminalCenter>
      <x>-250.00</x>
      <y>-810.00</y>
    </TerminalPatternLowerLeftTerminalCenter>
  </StandardArray>
</TerminalPattern>
<TerminalPattern>
  <ID>Terminal Pattern ID 7</ID>
  <Name>South Inst 2 VCCIO Staggered</Name>
  <StandardArray>
    <nx>2</nx>
    <ny>3</ny>
    <dx>100.00</dx>
    <dy>160.00</dy>
    <TerminalPatternLowerLeftTerminalCenter>
      <x>-200.00</x>
      <y>-890.00</y>
    </TerminalPatternLowerLeftTerminalCenter>
  </StandardArray>
</TerminalPattern>
<TerminalPattern-Array>

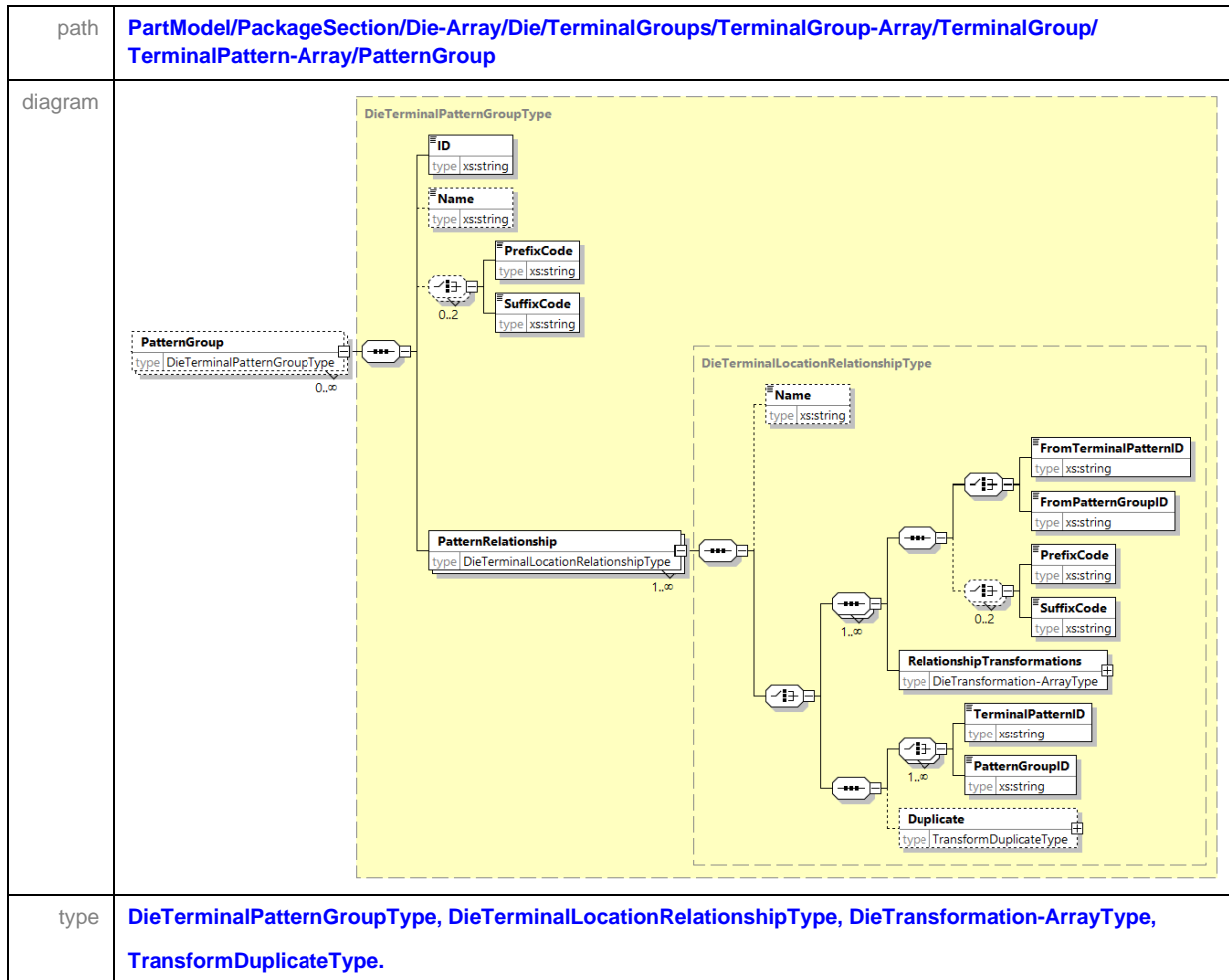
```

NOTE: The *TerminalPatternLowerLeftTerminalCenter* has been specified in each of these seven Terminal Patterns above, yet this is un-necessary, as the offset from one terminal pattern to the next terminal pattern could have been specified under section 6.5.1.2 Relationship – to – Active Zone below. The above *TerminalPatternLowerLeftTerminalCenter* coordinates are calculated based off the assumption that the entirety of Bank 1 is centered with the centered of the *ActiveZone*. In reality, Bank 1 may not be the final full configuration of all the die circuitry within the *ActiveZone*, and it may be just simply easier to adjust the final configuration with respect to the Active Zone instead of calculating all of the intermediate coordinates for every *TerminalPatternLowerLeftTerminalCenter*.

6.5.1.1.1 Terminal Pattern (cont'd)

Alternatively, *TerminalPatterns* can be grouped under a *PatternGroup* and their relationship between each other been determined via the *Offset* values. The *PatternGroup* could then be placed with respect to the *ActiveZone* under section 6.5.1.2 below.

6.5.1.1.2 Pattern Group



Before creating any *PatternGroup*, it is best to understand the most efficient form of defining the grouping to represent the duplication for the North and South channels as define in Figure 20 - Complex Array of 12 instantiations of a UCle Function above.

- Terminal Pattern ID 1 and 2 across all 12 instantiations of the UCle in the Bank 1 example have a contiguous sequential terminal numbering.
- Terminal Pattern ID 5 is a shared set of VSS terminals between the South and North channels and is not duplicated when a copy of the South channel is duplicated to form the North channel. For this reason, Terminal Pattern ID 5 does not form part of the *PatternGroup* that will be used later to create the North channel. However, Terminal Pattern ID 5 continues the terminal numbering from the Back to the Front for Bank 1 for that singular column of terminals.

6.5.1.1.2 Pattern Group (cont'd)

- The remaining terminals as defined within Terminal Pattern ID 3, 4, 6 and 7 across all instantiations in Bank 1 follow the S-H (Snake – Horizontal) pattern and continue the terminal numbering sequence from the shared VSS terminals.

| | |
|---------|--|
| path | PartModel/PackageSection/Die-Array/Die/TerminalGroups/TerminalGroup-Array/TerminalGroup/ TerminalPattern-Array/PatternGroup/PatternRelationship/RelationshipTransformations |
| diagram | <p>The diagram illustrates the structure of the RelationshipTransformations class, which is an array type (DieTransformation-ArrayType). It contains two main transformation types: Mirror and Rotate.</p> <ul style="list-style-type: none"> Mirror (type TransformMirrorType): <ul style="list-style-type: none"> Optional Origin (type JEP30-D10:EmptyType). Optional SelectionCenter (type JEP30-D10:EmptyType). Optional Coordinate (type JEP30-D10:PointXYType). Optional choice between Horizontal (type JEP30-D10:EmptyType) and Vertical (type JEP30-D10:EmptyType). Rotate (type TransformRotateType): <ul style="list-style-type: none"> Optional Origin (type JEP30-D10:EmptyType). Optional SelectionCenter (type JEP30-D10:EmptyType). Optional Coordinate (type JEP30-D10:PointXYType). Optional Angle (type xs:decimal). <p>The JEP30-D10:PointXYType structure consists of two optional fields: x (type xs:decimal) and y (type xs:decimal).</p> |
| type | DieTransformation-ArrayType, TransformMirrorType, JEP30-D10:EmptyType, JEP30-D10:PointXYType, TransformRotateType. |

6.5.1.1.2 Pattern Group (cont'd)

| | |
|---------|---|
| path | PartModel/PackageSection/Die-Array/Die/TerminalGroups/TerminalGroup-Array/TerminalGroup/ TerminalPattern-Array/PatternGroup/PatternRelationship/Duplicate |
| diagram | <p>The diagram illustrates the structure of the <code>TransformDuplicateType</code> and its associated components. It is enclosed in a dashed yellow box.</p> <ul style="list-style-type: none">TransformDuplicateType (type: <code>xs:integer</code>) is the root type, connected to nx (type: <code>xs:integer</code>) and ny (type: <code>xs:integer</code>) via a <code>1..*</code> relationship.Duplicate (type: <code>TransformDuplicateType</code>) is a dashed box containing a <code>1..*</code> relationship to TransformDuplicateType.TransformDuplicateType is connected to a <code>0..*</code> relationship, which is further connected to a <code>1..2</code> relationship.The <code>1..2</code> relationship branches into two paths:<ul style="list-style-type: none">Path 1: A <code>1..2</code> relationship connects to a <code>1..*</code> relationship, which then branches into:<ul style="list-style-type: none">X-PrefixCode (type: <code>xs:string</code>) with a <code>1..*</code> relationship.X-PrefixRowCode (type: <code>RowTerminalNumberOrderingType</code>) with a <code>1..*</code> relationship.X-SuffixCode (type: <code>xs:string</code>) with a <code>1..*</code> relationship.X-SuffixRowCode (type: <code>RowTerminalNumberOrderingType</code>) with a <code>1..*</code> relationship.Path 2: A <code>1..2</code> relationship connects to a <code>1..*</code> relationship, which then branches into:<ul style="list-style-type: none">Y-PrefixCode (type: <code>xs:string</code>) with a <code>1..*</code> relationship.Y-PrefixColumnCode (type: <code>ColumnTerminalNumberOrderingType</code>) with a <code>1..*</code> relationship.Y-SuffixCode (type: <code>xs:string</code>) with a <code>1..*</code> relationship.Y-SuffixColumnCode (type: <code>ColumnTerminalNumberOrderingType</code>) with a <code>1..*</code> relationship.Path 3: A <code>1..2</code> relationship connects to a <code>1..*</code> relationship, which then branches into:<ul style="list-style-type: none">PrefixCode (type: <code>xs:string</code>) with a <code>1..*</code> relationship.PrefixCode is connected to a <code>1..*</code> relationship, which branches into:<ul style="list-style-type: none">PrefixNumericalSequence (type: <code>NumericalSequenceType</code>) with a <code>1..*</code> relationship.PrefixAlphabeticalSequence (type: <code>AlphabeticalSequenceType</code>) with a <code>1..*</code> relationship.PrefixSeparator (type: <code>xs:string</code>) with a <code>1..*</code> relationship.SuffixCode (type: <code>xs:string</code>) with a <code>1..*</code> relationship.SuffixCode is connected to a <code>1..*</code> relationship, which branches into:<ul style="list-style-type: none">SuffixNumericalSequence (type: <code>NumericalSequenceType</code>) with a <code>1..*</code> relationship.SuffixAlphabeticalSequence (type: <code>AlphabeticalSequenceType</code>) with a <code>1..*</code> relationship.SuffixSeparator (type: <code>xs:string</code>) with a <code>1..*</code> relationship. dx (type: <code>xs:decimal</code>) and dy (type: <code>xs:decimal</code>) are shown at the bottom, connected to the main structure via a <code>1..*</code> relationship. |
| type | TransformDuplicateType, RowTerminalNumberOrderingType, ColumnTerminalNumberOrderingType, NumericalSequenceType, AlphabeticalSequenceType. |

6.5.1.1.2 Pattern Group (cont'd)

Option 1 – Leveraging Pattern Groups for the purpose of applying Terminal numbering.

- Group *TerminalPattern* ID 1 and 2 into a single *PatternGroup* ID 1 called “South_Inst1_TrmlPtrn1-2”.
- Create an array of *PatternGroup* ID 1 “South_Inst1_TrmlPtrn1-2” six times in the Y direction via a new *PatternGroup* ID 2 called “South_Inst1-6_PtrnGrp1”.
- Duplicate *PatternGroup* ID 2 “South_Inst1-6_TrmlPtrn1-2” one time in the X direction into a new *PatternGroup* ID 3 called “South-North_Inst1-6_PtrnGrp2”, ensuring that you mirror in both the horizontal and vertical axis for the North channel.
 - This enables the assignment of a unique sequence of terminal numbering to be applied specifically to the terminals defined within *PatternGroup* ID 3.
- Group *TerminalPattern* ID 3 and 4 into a single *PatternGroup* ID 4 called “South_Inst1_TrmlPtrn3-4”.
- Create an array of *PatternGroup* ID 4 “South_Inst1, 3, 5_PtrnGrp4” three times in the Y direction into a new *PatternGroup* ID 5 “South_Inst1, 3, 5_PtrnGrp4”.
- Group *TerminalPattern* ID 6 and 7 into a single *PatternGroup* ID 6 called “South_Inst2_TrmlPtrn6-7”.
- Create an array of *PatternGroup* ID 6 “South_Inst2, 4, 6_PtrnGrp6” three times in the Y direction into a new *PatternGroup* ID 7 “South_Inst2, 4, 6_PtrnGrp6”.
- Group *PatternGroup* ID 5 and 7 into a single *PatternGroup* ID 8 called “South_Inst1-6_PtrnGrp5, 7”
- Duplicate *PatternGroup* ID 8 “South_Inst1-6_PtrnGrp5, 7” one time in the X direction into a new *PatternGroup* ID 9 called “South-North_Inst1-6_PtrnGrp8”, with a straight offset and without rotation or mirroring.
 - This enables the assignment of a unique sequence of terminal numbering to be applied specifically to the terminals defined within *PatternGroup* ID 9.
- Create an array of *TerminalPattern* ID 5 “VSS Center Inst 1” six times in the Y direction via a new *PatternGroup* ID 10 called “VSS Center Inst 1-6”.
 - This enables the assignment of a unique sequence of terminal numbering to be applied specifically to the terminals defined within *PatternGroup* ID 10.
- Group *PatternGroup* ID 3, 9 and 10 into a single *PatternGroup* ID 11 called “Bank 1”.

6.5.1.1.2 Pattern Group (cont'd)

Table 32 - Pattern Group Construction

| Pattern Group ID | Name | Pattern Relationship | nx | ny | Pitch (dx) / Offset (x) | Pitch (dy) / Offset (y) |
|------------------|------------------------------|---|----|----|-------------------------|-------------------------|
| 1 | South_Inst1_TrmlPtrn1-2 | Terminal Pattern ID 1 Terminal Pattern ID 2 | | | | |
| 2 | South_Inst1-6_PtrnGrp1 | Duplicate Pattern Group ID 1 | 1 | 6 | 0 | 500.00 |
| 3 | South-North_Inst1-6_PtrnGrp2 | Offset & Mirror (H & V) Pattern Group ID 2 | | | 1900.00 | 0 |
| 4 | South_Inst1_TrmlPtrn3-4 | Terminal Pattern ID 3 Terminal Pattern ID 4 | | | | |
| 5 | South_Inst1, 3, 5_PtrnGrp2 | Duplicate Pattern Group ID 4 | 1 | 3 | 0 | 1000.00 |
| 6 | South_Inst2_TrmlPtrn6-7 | Terminal Pattern ID 6 Terminal Pattern ID 7 | | | | |
| 7 | South_Inst2, 4, 6_PtrnGrp6 | Duplicate Pattern Group ID 6 | 1 | 3 | 0 | 1000.00 |
| 8 | South_Inst1-6_PtrnGrp5, 7 | Pattern Group ID 5 Pattern Group ID 7 | | | | |
| 9 | South-North_Inst1-6_PtrnGrp8 | Offset & Mirror (H & V) Pattern Group ID 2 | | | 300.00 | 0 |
| 10 | VSS Center Inst 1-6 | Terminal Pattern ID 5 | 1 | 6 | 0 | 500.00 |
| 11 | Bank 1 | Pattern Group ID 3 Pattern Group ID 9 Pattern Group ID 10 | | | | |

```

<PatternGroup>
  <ID>Pattern Group ID 1</ID>
  <Name>South_Inst1_TrmlPtrn1-2</Name>
  <PatternRelationship>
    <TerminalPatternID>Terminal Pattern ID 1</TerminalPatternID>
    <TerminalPatternID>Terminal Pattern ID 2</TerminalPatternID>
  </PatternRelationship>
</PatternGroup>
<PatternGroup>
  <ID>Pattern Group ID 2</ID>
  <Name>South_Inst1-6_PtrnGrp1</Name>
  <PatternRelationship>
    <PatternGroupID>Pattern Group ID 1</PatternGroupID>
    <Duplicate>
      <nx>1</nx>
      <ny>6</ny>
      <Y-PrefixCode>Inst 1</Y-PrefixCode>
      <Y-PrefixCode>Inst 2</Y-PrefixCode>
      <Y-PrefixCode>Inst 3</Y-PrefixCode>
      <Y-PrefixCode>Inst 4</Y-PrefixCode>
      <Y-PrefixCode>Inst 5</Y-PrefixCode>
      <Y-PrefixCode>Inst 6</Y-PrefixCode>
      <dy>500.00</dy>
    </Duplicate>
  </PatternRelationship>
</PatternGroup>

```

6.5.1.1.2 Pattern Group (cont'd)

```

<PatternGroup>
  <ID>Pattern Group ID 3</ID>
  <Name>South-North_Inst1-6_PtrnGrp2</Name>
  <PatternRelationship>
    <FromPatternGroupID>Pattern Group ID 2</FromPatternGroupID>
    <RelationshipTransformations>
      <ToPatternGroupID>Pattern Group ID 2</ToPatternGroupID>
      <Offset>
        <x>1900</x>
        <y>0</y>
      </Offset>
      <Mirror>
        <SelectionCenter/>
        <Horizontal/>
      </Mirror>
      <Mirror>
        <SelectionCenter/>
        <Vertical/>
      </Mirror>
    </RelationshipTransformations>
  </PatternRelationship>
</PatternGroup>
<PatternGroup>
  <ID>Pattern Group ID 4</ID>
  <Name>South_Inst1_Trm1Ptrn3-4</Name>
  <PatternRelationship>
    <TerminalPatternID>Terminal Pattern ID 3</TerminalPatternID>
    <TerminalPatternID>Terminal Pattern ID 4</TerminalPatternID>
  </PatternRelationship>
</PatternGroup>
<PatternGroup>
  <ID>Pattern Group ID 5</ID>
  <Name>South_Inst1, 3, 5_PtrnGrp4</Name>
  <PatternRelationship>
    <PatternGroupID>Pattern Group ID 4</PatternGroupID>
    <Duplicate>
      <nx>1</nx>
      <ny>3</ny>
      <Y-PrefixCode>Inst 1</Y-PrefixCode>
      <Y-PrefixCode>Inst 3</Y-PrefixCode>
      <Y-PrefixCode>Inst 5</Y-PrefixCode>
      <dy>1000.00</dy>
    </Duplicate>
  </PatternRelationship>
</PatternGroup>
<PatternGroup>
  <ID>Pattern Group ID 6</ID>
  <Name>South_Inst2_Trm1Ptrn6-7</Name>
  <PatternRelationship>
    <TerminalPatternID>Terminal Pattern ID 6</TerminalPatternID>
    <TerminalPatternID>Terminal Pattern ID 7</TerminalPatternID>
  </PatternRelationship>
</PatternGroup>

```

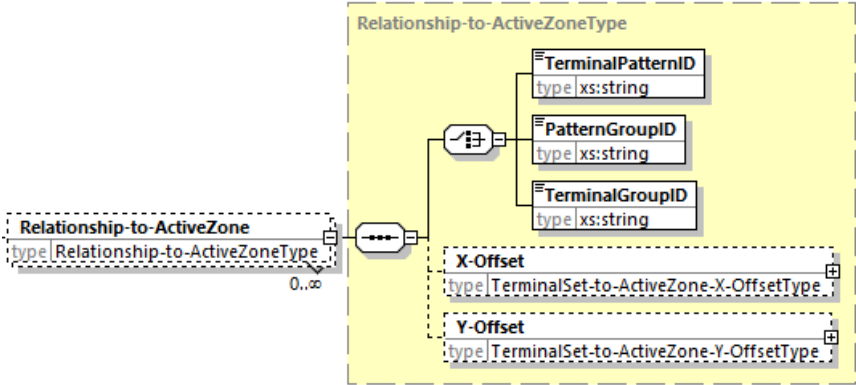
6.5.1.1.2 Pattern Group (cont'd)

```
<PatternGroup>
  <ID>Pattern Group ID 7</ID>
  <Name>South_Inst2, 4, 6_PtrnGrp6</Name>
  <PatternRelationship>
    <PatternGroupID>Pattern Group ID 6</PatternGroupID>
    <Duplicate>
      <nx>1</nx>
      <ny>3</ny>
      <Y-PrefixCode>Inst 2</Y-PrefixCode>
      <Y-PrefixCode>Inst 4</Y-PrefixCode>
      <Y-PrefixCode>Inst 6</Y-PrefixCode>
      <dy>1000.00</dy>
    </Duplicate>
  </PatternRelationship>
</PatternGroup>
<PatternGroup>
  <ID>Pattern Group ID 8</ID>
  <Name>South_Inst1-6_PtrnGrp5, 7</Name>
  <PatternRelationship>
    <PatternGroupID>Pattern Group ID 5</PatternGroupID>
    <PatternGroupID>Pattern Group ID 7</PatternGroupID>
  </PatternRelationship>
</PatternGroup>
<PatternGroup>
  <ID>Pattern Group ID 9</ID>
  <Name>South-North_Inst1-6_PtrnGrp8</Name>
  <PatternRelationship>
    <FromPatternGroupID>Pattern Group ID 8</FromPatternGroupID>
    <RelationshipTransformations>
      <ToPatternGroupID>Pattern Group ID 8</ToPatternGroupID>
      <Offset>
        <x>300</x>
        <y>0</y>
      </Offset>
    </RelationshipTransformations>
  </PatternRelationship>
</PatternGroup>
<PatternGroup>
  <ID>Pattern Group ID 10</ID>
  <Name>VSS Center Inst 1-6</Name>
  <PatternRelationship>
    <TerminalPatternID>Terminal Pattern ID 5</TerminalPatternID>
    <Duplicate>
      <nx>1</nx>
      <ny>6</ny>
      <dy>500.00</dy>
    </Duplicate>
  </PatternRelationship>
</PatternGroup>
```

6.5.1.1.2 Pattern Group (cont'd)

```
<PatternGroup>
  <ID>Pattern Group ID 11</ID>
  <Name>Bank 1</Name>
  <PatternRelationship>
    <PatternGroupID>Pattern Group ID 3</PatternGroupID>
    <PatternGroupID>Pattern Group ID 9</PatternGroupID>
    <PatternGroupID>Pattern Group ID 10</PatternGroupID>
  </PatternRelationship>
  :
  :
</PatternGroup>
```

6.5.1.2 Relationship – to – Active Zone

| | |
|---------|--|
| path | PartModel/PackageSection/Die-Array/Die/TerminalGroups/TerminalGroup-Array/Relationship-to-ActiveZone |
| diagram |  |
| type | Relationship-to-ActiveZoneType, TerminalSet-to-ActiveZone-X-OffsetType, TerminalSet-to-ActiveZone-Y-OffsetType |

For non-symmetrical Terminals around the [ActiveZone](#) center, offset dimensions may be referenced from the Terminal Contact Area to the [ActiveZone](#) area. This data can be captured in the [Relationship-to-ActiveZone](#) section.

6.5.1.2.1 X-Offset



6.5.1.2.1 X-Offset (cont'd)

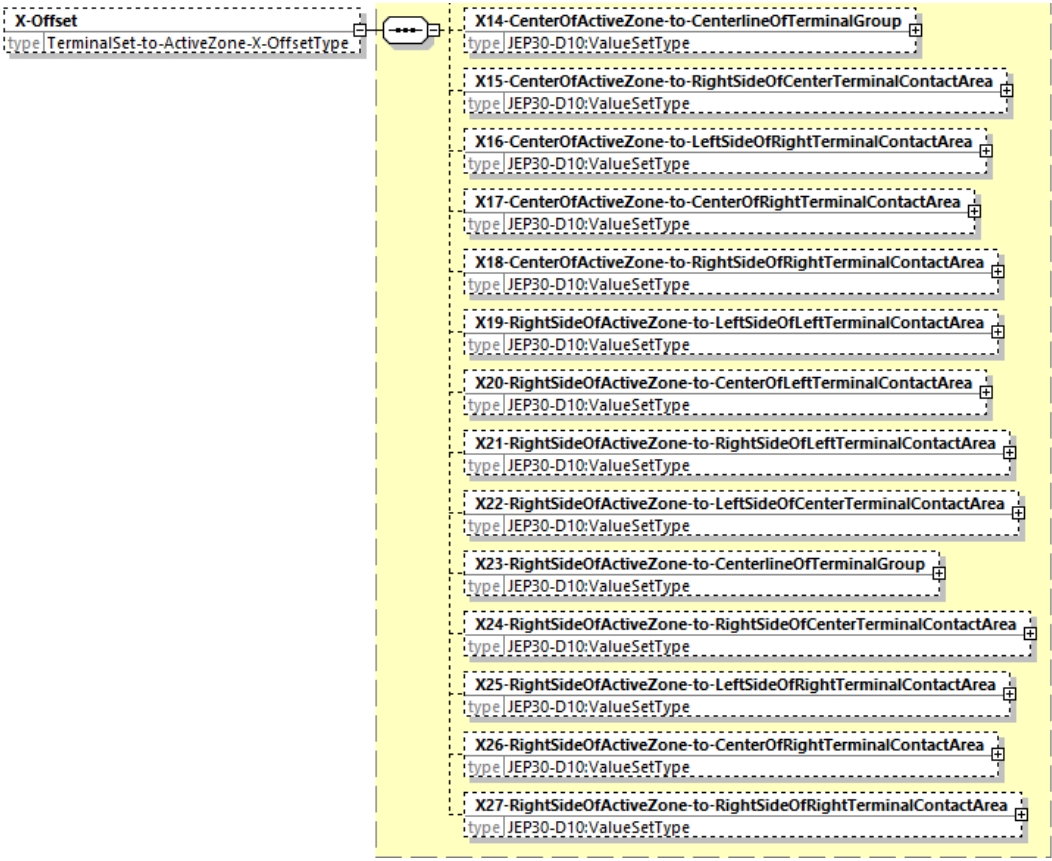
| | |
|---------------------------|---|
| diagram part 2 of 2 |  <p>The diagram illustrates the X-Offset dimensions from a terminal group to various active zone points. It shows a terminal group on the left and a series of active zone points on the right, each with a corresponding X-Offset dimension. The dimensions are listed as follows:</p> <ul style="list-style-type: none"> X14-CenterOfActiveZone-to-CenterlineOfTerminalGroup X15-CenterOfActiveZone-to-RightSideOfCenterTerminalContactArea X16-CenterOfActiveZone-to-LeftSideOfRightTerminalContactArea X17-CenterOfActiveZone-to-CenterOfRightTerminalContactArea X18-CenterOfActiveZone-to-RightSideOfRightTerminalContactArea X19-RightSideOfActiveZone-to-LeftSideOfLeftTerminalContactArea X20-RightSideOfActiveZone-to-CenterOfLeftTerminalContactArea X21-RightSideOfActiveZone-to-RightSideOfLeftTerminalContactArea X22-RightSideOfActiveZone-to-LeftSideOfCenterTerminalContactArea X23-RightSideOfActiveZone-to-CenterlineOfTerminalGroup X24-RightSideOfActiveZone-to-RightSideOfCenterTerminalContactArea X25-RightSideOfActiveZone-to-LeftSideOfRightTerminalContactArea X26-RightSideOfActiveZone-to-CenterOfRightTerminalContactArea X27-RightSideOfActiveZone-to-RightSideOfRightTerminalContactArea |
| type | TerminalSet-to-ActiveZone-X-OffsetType, JEP30-D10:ValueSetType |

Table 33 — Terminal Group to Active Zone X-Offset lists out all the various dimensions that can be selected from any major point of the [ActiveZone](#) to any point of the terminal contact area in X direction.

Center of Center Terminal Contact Area is the same as Centerline of Terminal Group, however the use of the "...Centerline of Terminal Group" is more encompassing since it also includes the centerline of a row or column of an even number of terminals, i.e., midway between terminal 2 and terminal 3 of 4 terminals in a row or column.

6.5.1.2.1 X-Offset (cont'd)

Table 33 - Terminal Group to Active Zone X-Offset

| Symbol | Description |
|--------|---|
| X1 | Left side of Active Zone to Left Side of Left Terminal Contact Area |
| X2 | Left side of Active Zone to Center of Left Terminal Contact Area |
| X3 | Left side of Active Zone to Right side of Left Terminal Contact Area |
| X4 | Left side of Active Zone to Left Side of Center Terminal Contact Area |
| X5 | Left side of Active Zone to Centerline of Terminal Group |
| X6 | Left side of Active Zone to Right side of Center Terminal Contact Area |
| X7 | Left side of Active Zone to Left side of Right Terminal Contact Area |
| X8 | Left side of Active Zone to Center of Right Terminal Contact Area |
| X9 | Left side of Active Zone to Right side of Right Terminal Contact Area |
| X10 | Center of Active Zone to Left side of Left Terminal Contact Area |
| X11 | Center of Active Zone to Center of Left Terminal Contact Area |
| X12 | Center of Active Zone to Right side of Left Terminal Contact Area |
| X13 | Center of Active Zone to Left side of Center Terminal Contact Area |
| X14 | Center of Active Zone to Centerline of Terminal Group |
| X15 | Center of Active Zone to Right side of Center Terminal Contact Area |
| X16 | Center of Active Zone to Left side of Right Terminal Contact Area |
| X17 | Center of Active Zone to Center of Right Terminal Contact Area |
| X18 | Center of Active Zone to Right side of Right Terminal Contact Area |
| X19 | Right side of Active Zone to Left side of Left Terminal Contact Area |
| X20 | Right side of Active Zone to Center of Left Terminal Contact Area |
| X21 | Right side of Active Zone to Right side of Left Terminal Contact Area |
| X22 | Right side of Active Zone to Left side of Center Terminal Contact Area |
| X23 | Right side of Active Zone to Centerline of Terminal Group |
| X24 | Right side of Active Zone to Right side of Center Terminal Contact Area |
| X25 | Right side of Active Zone to Left side of Right Terminal Contact Area |
| X26 | Right side of Active Zone to Center of Right Terminal Contact Area |
| X27 | Right side of Active Zone to Right side of Right Terminal Contact Area |

6.5.1.2.2 Y-Offset

| | | |
|---------------------------|---|--|
| path | PartModel/PackageSection/Die-Array/Die/TerminalGroups/TerminalGroup-Array/Relationship-to-ActiveZone/Y-Offset | |
| diagram part 1 of 2 | <div><div>TerminalSet-to-ActiveZone-Y-OffsetType</div><div><div>Y1-BackSideOfActiveZone-to-BackSideOfBackTerminalContactArea</div><div>type JEP30-D10:ValueSetType</div></div><div><div>Y2-BackSideOfActiveZone-to-CenterOfBackTerminalContactArea</div><div>type JEP30-D10:ValueSetType</div></div><div><div>Y3-BackSideOfActiveZone-to-FrontSideOfBackTerminalContactArea</div><div>type JEP30-D10:ValueSetType</div></div><div><div>Y4-BackSideOfActiveZone-to-BackSideOfCenterTerminalContactArea</div><div>type JEP30-D10:ValueSetType</div></div><div><div>Y5-BackSideOfActiveZone-to-CenterlineOfTerminalGroup</div><div>type JEP30-D10:ValueSetType</div></div><div><div>Y6-BackSideOfActiveZone-to-FrontSideOfCenterTerminalContactArea</div><div>type JEP30-D10:ValueSetType</div></div><div><div>Y7-BackSideOfActiveZone-to-BackSideOfFrontTerminalContactArea</div><div>type JEP30-D10:ValueSetType</div></div><div><div>Y8-BackSideOfActiveZone-to-CenterOfFrontTerminalContactArea</div><div>type JEP30-D10:ValueSetType</div></div><div><div>Y9-BackSideOfActiveZone-to-FrontSideOfFrontTerminalContactArea</div><div>type JEP30-D10:ValueSetType</div></div><div><div>Y10-CenterOfActiveZone-to-BackSideOfBackTerminalContactArea</div><div>type JEP30-D10:ValueSetType</div></div><div><div>Y11-CenterOfActiveZone-to-CenterOfBackTerminalContactArea</div><div>type JEP30-D10:ValueSetType</div></div><div><div>Y12-CenterOfActiveZone-to-FrontSideOfBackTerminalContactArea</div><div>type JEP30-D10:ValueSetType</div></div><div><div>Y13-CenterOfActiveZone-to-BackSideOfCenterTerminalContactArea</div><div>type JEP30-D10:ValueSetType</div></div><div><div>Y14-CenterOfActiveZone-to-CenterlineOfTerminalGroup</div><div>type JEP30-D10:ValueSetType</div></div></div> <div><div>Y-Offset</div><div>type TerminalSet-to-ActiveZone-Y-OffsetType</div></div> | |

6.5.1.2.2 Y-Offset (cont'd)


| | |
|------------------------------------|---|
| <p>diagram part 2 of 2</p> |  |
| <p>type</p> | <p>TerminalSet-to-ActiveZone-X-OffsetType, JEP30-D10:ValueSetType</p> |

Table 34 — Terminal Group to Active Zone Y-Offset lists out all the various dimensions that can be selected from any major point of the *ActiveZone* to any point of the terminal contact area in X direction.

6.5.1.2.2 Y-Offset (cont'd)

Table 34 - Terminal Group to Active Zone Y-Offset

| Symbol | Description |
|--------|---|
| Y1 | Back side of Active Zone to Back side of Back Terminal Contact Area |
| Y2 | Back side of Active Zone to Center of Back Terminal Contact Area |
| Y3 | Back side of Active Zone to Front side of Back Terminal Contact Area |
| Y4 | Back side of Active Zone to Back side of Center Terminal Contact Area |
| Y5 | Back side of Active Zone to Centerline of Terminal Group |
| Y6 | Back side of Active Zone to Front side of Center Terminal Contact Area |
| Y7 | Back side of Active Zone to Back side of Front Terminal Contact Area |
| Y8 | Back side of Active Zone to Center of Front Terminal Contact Area |
| Y9 | Back side of Active Zone to Front side of Front Terminal Contact Area |
| Y10 | Center of Active Zone to Back side of Back Terminal Contact Area |
| Y11 | Center of Active Zone to Center of Back Terminal Contact Area |
| Y12 | Center of Active Zone to Front side of Back Terminal Contact Area |
| Y13 | Center of Active Zone to Back side of Center Terminal Contact Area |
| Y14 | Center of Active Zone to Centerline of Terminal Group |
| Y15 | Center of Active Zone to Front side of Center Terminal Contact Area |
| Y16 | Center of Active Zone to Back side of Front Terminal Contact Area |
| Y17 | Center of Active Zone to Center of Front Terminal Contact Area |
| Y18 | Center of Active Zone to Front side of Front Terminal Contact Area |
| Y19 | Front side of Active Zone to Back side of Back Terminal Contact Area |
| Y20 | Front side of Active Zone to Center of Back Terminal Contact Area |
| Y21 | Front side of Active Zone to Front side of Back Terminal Contact Area |
| Y22 | Front side of Active Zone to Back side of Center Terminal Contact Area |
| Y23 | Front side of Active Zone to Centerline of Terminal Group |
| Y24 | Front side of Active Zone to Front side of Center Terminal Contact Area |
| Y25 | Front side of Active Zone to Back side of Front Terminal Contact Area |
| Y26 | Front side of Active Zone to Center of Front Terminal Contact Area |
| Y27 | Front side of Active Zone to Front side of Front Terminal Contact Area |

6.5.1.3 Terminal Group to Terminal Group Relationships

| | |
|---------|--|
| path | PartModel/PackageSection/Die-Array/Die/TerminalGroups/TerminalGroup-Array/ TerminalGroupToTerminalGroupRelationships |
| diagram | |
| type | DieTerminalGroupToTerminalGroupRelationshipsType , JEP30-D10:PointXYType |

6.5.2 Terminal Regions - Array

| | |
|---------|---|
| path | PartModel/PackageSection/Die-Array/Die/TerminalGroups/Regions-Array |
| diagram | |
| type | Region-ArrayType , RegionType , RegionAreaSelectionType , RegionTerminalSelectionType . |

6.5.2.1 Area Selection

| | |
|---------|---|
| path | PartModel/PackageSection/Die-Array/Die/TerminalGroups/Regions-Array/Region/AreaSelection |
| diagram | |
| type | RegionAreaSelectionType, ReferenceRectangleGroupType, ReferenceRegularPolygonGroupType, JEP30-D10:PointXYType, ContourShapeGroupType. |

6.5.2.2 Terminal Selection

| | |
|---------|--|
| path | PartModel/PackageSection/Die-Array/Die/TerminalGroups/Regions-Array/Region/TerminalSelection |
| diagram | |
| type | RegionTerminalSelectionType, RegionalTerminalSelectionType. |

6.5.2.2.1 Select

| | |
|---------|--|
| path | <p>PartModel/PackageSection/Die-Array/Die/TerminalGroups/Regions-Array/Region/TerminalSelection/Select,</p> <p>PartModel/PackageSection/Die-Array/Die/TerminalGroups/Regions-Array/Region/TerminalSelection/Deselect</p> |
| diagram | <p>The diagram illustrates the structure of the <code>RegionalTerminalSelectionType</code> element. It is a complex type with several optional elements and a repeating group. The elements are as follows:</p> <ul style="list-style-type: none"> <code>TerminalPatternID</code> (type: <code>xs:string</code>) <code>PatternGroupID</code> (type: <code>xs:string</code>) <code>TerminalGroupID</code> (type: <code>xs:string</code>) <code>RowTerminalIndex</code> (type: <code>JEP30-D10:MinIntegerOfOneType</code>, minOccurs: 1, maxIncl: 1) <code>FromRowTerminalIndex</code> (type: <code>JEP30-D10:MinIntegerOfOneType</code>, minOccurs: 1, maxIncl: 1) <code>ToRowTerminalIndex</code> (type: <code>JEP30-D10:MinIntegerOfOneType</code>, minOccurs: 1, maxIncl: 1) <code>ColumnTerminalIndex</code> (type: <code>JEP30-D10:MinIntegerOfOneType</code>, minOccurs: 1, maxIncl: 1) <code>FromColumnTerminalIndex</code> (type: <code>JEP30-D10:MinIntegerOfOneType</code>, minOccurs: 1, maxIncl: 1) <code>ToColumnTerminalIndex</code> (type: <code>JEP30-D10:MinIntegerOfOneType</code>, minOccurs: 1, maxIncl: 1) <code>PolarTerminalIndex</code> (type: <code>JEP30-D10:MinIntegerOfOneType</code>, minOccurs: 1, maxIncl: 1) <code>FromPolarTerminalIndex</code> (type: <code>JEP30-D10:MinIntegerOfOneType</code>, minOccurs: 1, maxIncl: 1) <code>ToPolarTerminalIndex</code> (type: <code>JEP30-D10:MinIntegerOfOneType</code>, minOccurs: 1, maxIncl: 1) <code>FromTerminalNumber</code> (type: <code>xs:integer</code>) <code>ToTerminalNumber</code> (type: <code>xs:integer</code>) <p>The diagram also shows a repeating group (indicated by a dashed rectangle) containing the <code>FromTerminalNumber</code> and <code>ToTerminalNumber</code> elements. The <code>FromTerminalNumber</code> element is optional and has a minOccurs of 1 and a maxIncl of 1. The <code>ToTerminalNumber</code> element is optional and has a minOccurs of 1 and a maxIncl of 1.</p> |
| type | RegionalTerminalSelectionType. |

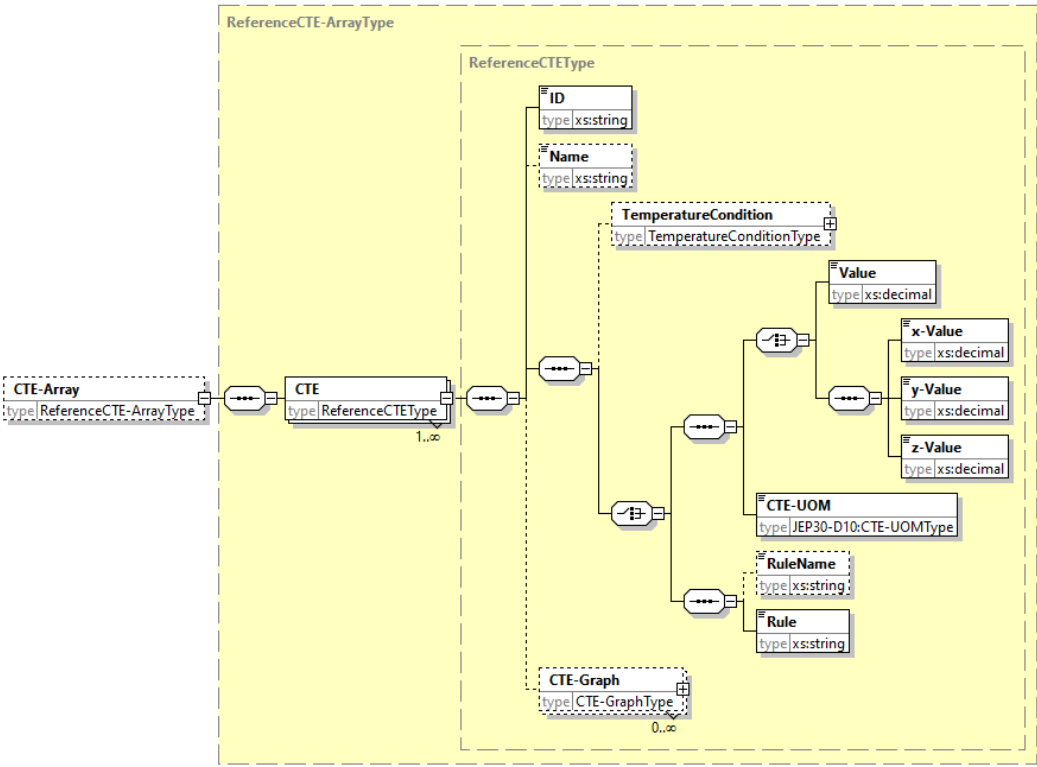
6.5.3 Terminal Materials - Array

| | |
|---------|--|
| path | PartModel/PackageSection/Die-Array/Die/TerminalGroups/TerminalMaterial-Array |
| diagram | <p>The diagram illustrates the XSD structure for the TerminalMaterial-Array. It is an array of TerminalMaterial elements, each of type DieTerminalMaterialType. The DieTerminalMaterialType contains the following elements:</p> <ul style="list-style-type: none"> ID: type <code>xs:string</code> Name: type <code>xs:string</code> J-Std-609AlloyComposition: type <code>J-Std-609AlloyCompositionType</code> J-Std-609e-code: type <code>J-Std-609e-codeType</code> BaseMaterial: type <code>BaseMaterialType</code> OtherBaseMaterial: type <code>xs:string</code> Plating-Array: type <code>Plating-ArrayType</code> |
| type | DieTerminalMaterial-ArrayType, DieTerminalMaterialType, J-Std-609AlloyCompositionType, J-Std-609e-codeType, BaseMaterialType, Plating-ArrayType. |

The *J-Std-609AlloyComposition* and *J-Std-609e-code* enumerated values are defined in “Annex A (informative) Example Alloys and Associated Material Codes” in the J-Std-609, “Marking, Symbols, and Labels of Leaded and Lead-Free Terminal Finished Materials Used in Electronic Assembly” standard.

The is *PlatingThicknessUOM* specified in *um*.

6.5.4 CTE - Array

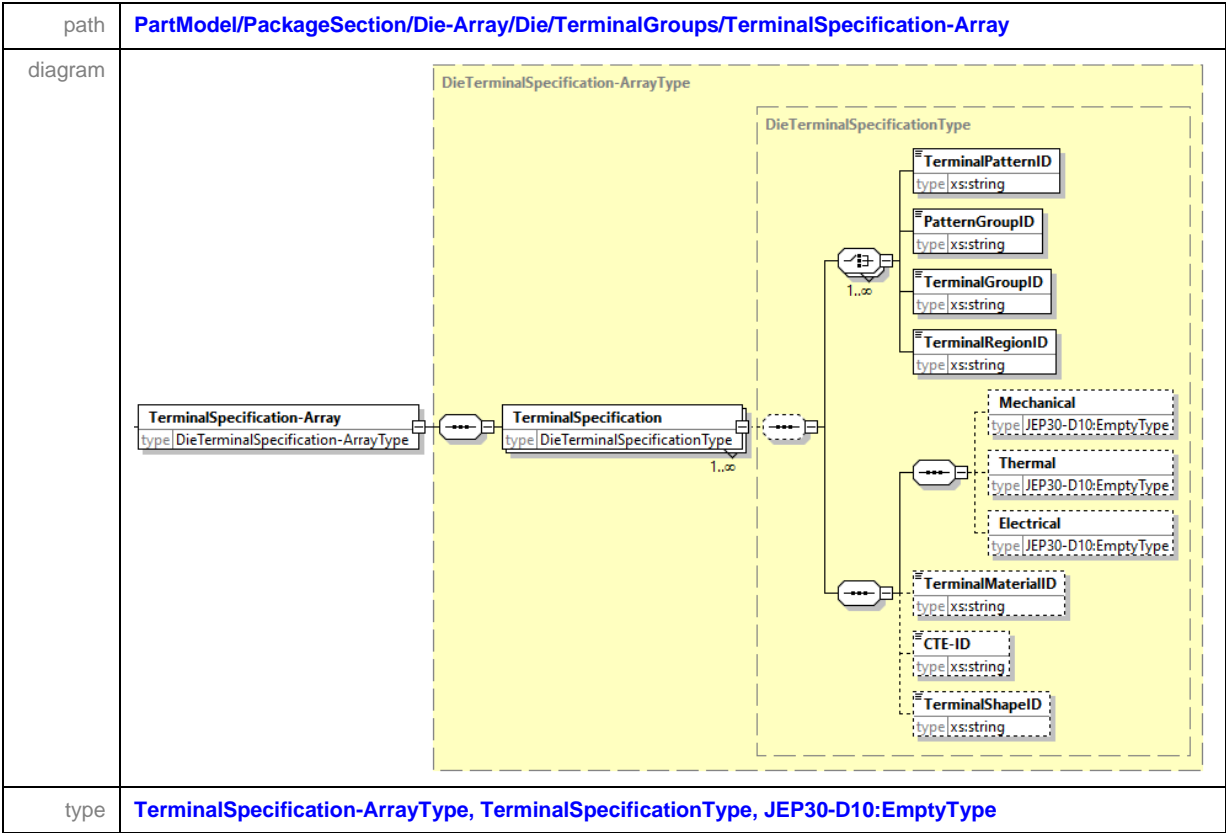
| | |
|---------|--|
| path | PartModel/PackageSection/Die-Array/Die/TerminalGroups/CTE-Array |
| diagram |  <p>The diagram illustrates the XSD structure for the CTE-Array. It features a main container ReferenceCTE-ArrayType (dashed box) which contains a CTE element (solid box, type ReferenceCTEType, cardinality 1..∞). The CTE element is further detailed within a ReferenceCTEType container (dashed box). This container includes an ID element (type xs:string), a Name element (type xs:string), a TemperatureCondition element (type TemperatureConditionType), a Value element (type xs:decimal), a CTE-UOM element (type JEP30-D10:CTE-UOMType), a RuleName element (type xs:string), a Rule element (type xs:string), and a CTE-Graph element (type CTE-GraphType, cardinality 0..∞). The Value element is further broken down into x-Value, y-Value, and z-Value elements, all of type xs:decimal. The TemperatureCondition element is also detailed with its own structure.</p> |
| type | ReferenceCTE-ArrayType, ReferenceCTEType, TemperatureConditionType, JEP30-D10:CTE-UOMType, CTE-GraphType. |

6.5.5 Terminal Shape - Array

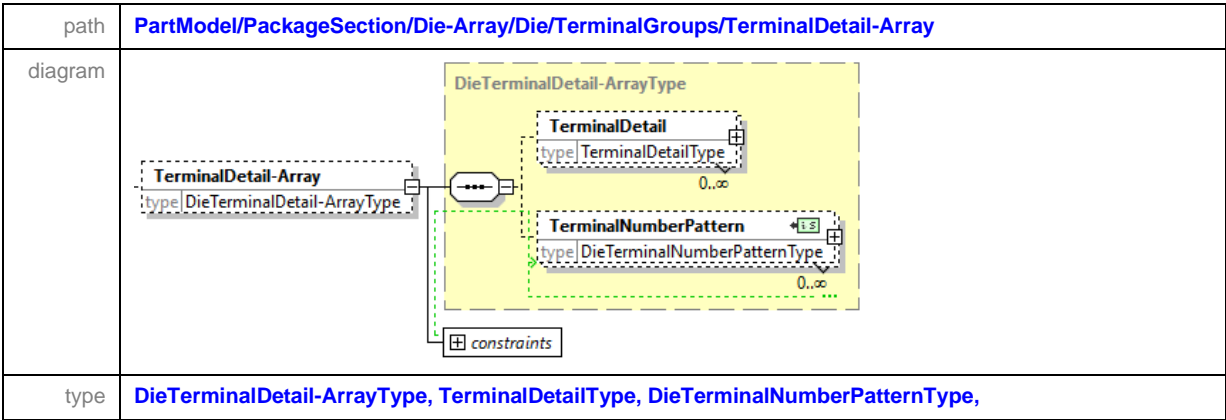
| | |
|---------|---|
| path | PartModel/PackageSection/Die-Array/Die/TerminalGroups/TerminalShape-Array |
| diagram | <p>The diagram illustrates the XSD structure for the TerminalShape-Array. It is an array of TerminalShape elements (0..∞). Each TerminalShape element contains an ID (xs:string) and a Name (xs:string). It also contains a choice of the following elements:</p> <ul style="list-style-type: none"> SMD (type: JEP30-D10:EmptyType) NSMD (type: JEP30-D10:EmptyType) Terminal (type: JEP30-D10:EmptyType) SoldermaskOpening (type: JEP30-D10:EmptyType) Pad (type: JEP30-D10:EmptyType) TerminalHeight (type: JEP30-D10:UnspecifiedDimensionalValueSetType) <ul style="list-style-type: none"> Rectangle (type: ReferenceRectangleType) RoundedRectangle (type: ReferenceRoundedRectangleType) ModifiedRectangle (type: ReferenceModifiedRectangleType) Circle (type: ReferenceCircleType) Double-D (type: ReferenceDouble-DType) Contour (type: ReferenceTerminalContourType) |
| type | DieTerminalShape-ArrayType , DieTerminalShapeType , JEP30-D10:EmptyType , JEP30-D10:UnspecifiedDimensionalValueSetType , ReferenceRectangleType , ReferenceRoundedRectangleType , ReferenceModifiedRectangleType , ReferenceCircleType , ReferenceDouble-DType , ReferenceTerminalContourType |

The [TerminalShape-Array](#) for Die's is a subset of the shapes defined in section 5.13.1.6 Terminal Shape above.

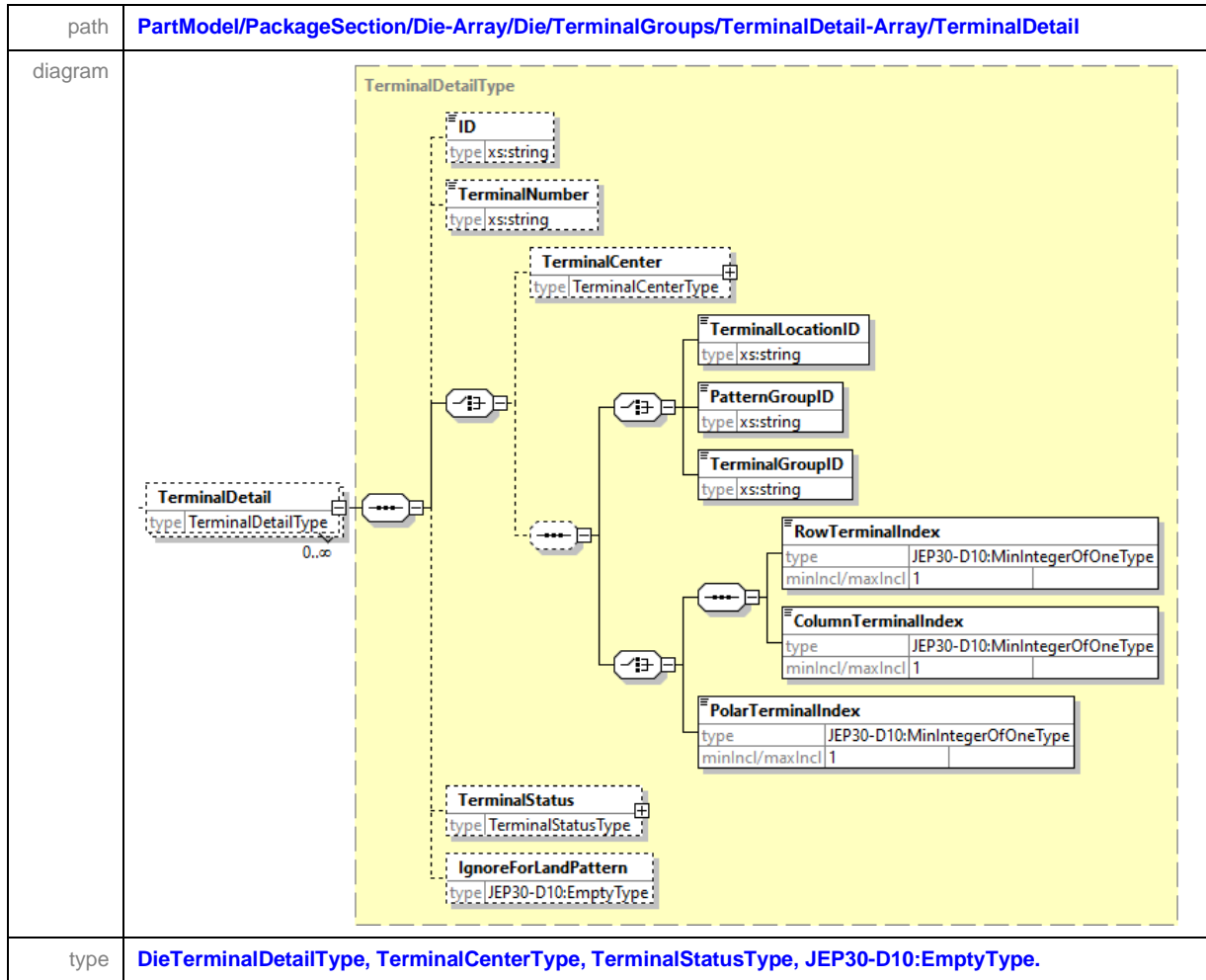
6.5.6 Terminal Specification - Array



6.5.7 Terminal Detail - Array



6.5.7.1 Terminal Detail



Similar to section 5.13.3.1 Terminal Detail above, this section identifies the relationship between the electrical *TerminalNumber* and the physical terminal location in the *ActiveZone*. The physical terminal location can be defined by a set of x,y coordinates, or by a combination of two choices, where the first choice is the identification of either the *TerminalPattern*, *PatternGroup* or the *TerminalGroup*, and the second choice is the combination of *RowTerminalIndex* and the *ColumnTerminalIndex* or the *PolarTerminalIndex*. TerminalDetail is primarily used when there is a need to specify the details for individual terminals that do not fit a pattern. This can be an excessive overhead for quantity of terminals, so it is preferred to use *TerminalNumberingPattern* as defined in section 6.5.7.2 below.

Based on Figure 21 – Single UCle Instantiation in South Channel Inst 1 above, *TerminalNumber* 196 is missing from *TerminalPatternID* 1, and *TerminalNumber* 234 is missing from *TerminalPatternID* 1. However, these Terminal numbers change from one instantiation to the next so a reference to the *TerminalNumber* for the purpose of specifying a missing status is incorrect, if you want to ensure that these relative locations are missing from every instantiation of these terminal patterns. In this case, reference to these missing terminals should be based their index location with respect to their terminal pattern, as shown in Figure 23.

6.5.7.1 Terminal Detail (cont'd)

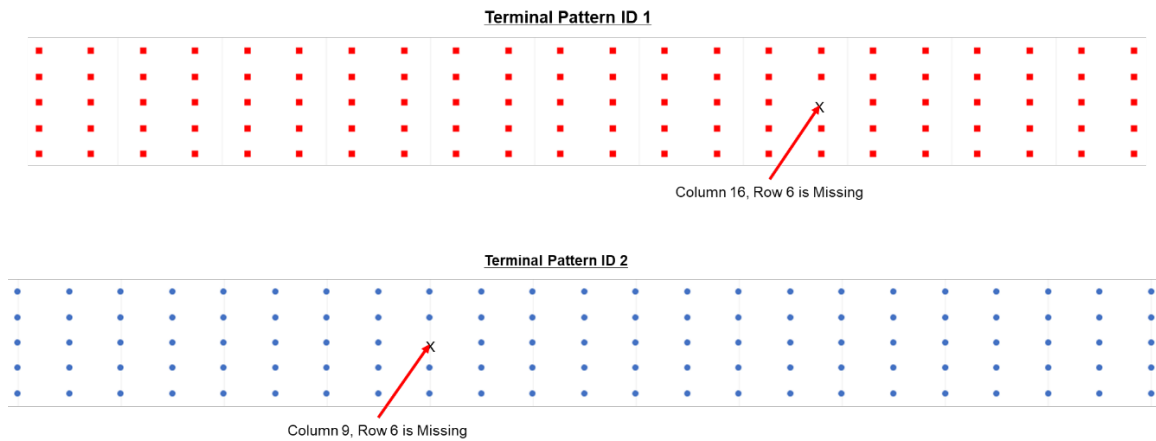


Figure 23 - Missing Terminal for Terminal Patterns

Specifying the missing terminal from the Terminal Pattern by reference to its location as opposed to a specific terminal number, means that by duplicating this pattern to other locations, the same respective terminal location within that pattern ID would have the same terminal status.

The status of *Missing*, *Deleted* and *Excluded* are defined in section called “Terminal-count suffixes” of JESD30.

```
<TerminalDetail>
  <ID>1</ID>
  <TerminalPatternID>1</TerminalPatternID>
  <RowTerminalIndex>3</RowTerminalIndex>
  <ColumnTerminalIndex>16</ColumnTerminalIndex>
  <TerminalStatus>
    <Missing/>
  </TerminalStatus>
</TerminalDetail>
<TerminalDetail>
  <ID>2</ID>
  <TerminalPatternID>2</TerminalPatternID>
  <RowTerminalIndex>3</RowTerminalIndex>
  <ColumnTerminalIndex>9</ColumnTerminalIndex>
  <TerminalStatus>
    <Missing/>
  </TerminalStatus>
</TerminalDetail>
```

6.5.7.2 Terminal Number Pattern

| | |
|---------|---|
| path | PartModel/PackageSection/Die-Array/Die/TerminalGroups/TerminalDetail-Array/TerminalNumberPattern |
| diagram | |
| type | DieTerminalNumberPatternType, DieFirstTerminalLocationType, TerminalDetailExceptionsType, DieTerminalNumberPatternGridSequentialType, DieTerminalNumberPatternGridType. |
| group | NumberingRestrictionGroup |

The *TerminalNumberPattern* section that can be defined specifically to a *TerminalPattern*, *PatternGroup*, *TerminalGroup* or to all the terminals within a specific *TerminalRegion*, or finally to all the terminals within an *ActiveZone*.

6.5.7.2.1 First Terminal Location

| | |
|---------|--|
| path | PartModel/PackageSection/Die-Array/Die/TerminalGroups/TerminalDetail-Array/TerminalNumberPattern/FirstTerminalLocation |
| diagram | |
| type | DieFirstTerminalLocationType, TerminalCenterType, JEP30-D10:MinIntegerOfOneType, LocationRelative-to-SelectionCenterType, LocationRelative-to-SelectionCenterCodeType, LocationRelative-to-SelectionCenterDescriptionType. |

The enumerated values for the *LocationRelative-to-SelectionCenter* (Code and Description) are defined in Table 35.

5.13.3.2.1 First Terminal Location (cont'd)

Table 35 - Location Relative to Selection Center

| Code | Description |
|------|--------------|
| SW | Southwest |
| SE | Southeast |
| NE | Northeast |
| NW | Northwest |
| BL | Back-Left |
| BC | Back-Center |
| BR | Back-Right |
| FL | Front-Left |
| FC | Front-Center |
| FR | Front-Right |
| LB | Left-Bottom |
| LC | Left-Center |
| LT | Left-Top |
| RB | Right-Bottom |
| RC | Right-Center |
| RT | Right-Top |
| L | Left |
| B | Back |
| R | Right |
| F | Front |

Reference the “Single Position” single terminal illustrations in JESD30, Annex A for graphical representations of some of the above descriptions.

6.5.7.2.2 Terminal Details Exception

| | |
|---------|---|
| path | PartModel/PackageSection/Die-Array/Die/TerminalGroups/TerminalDetail-Array/TerminalNumberPattern/TerminalDetailExceptions |
| diagram | |
| type | DieTerminalDetailExceptionsType, TerminalCenterType, TerminalIndexType, TerminalNumberType, TerminalStatusType. |

Referring to the example defined in section 6.5.1.1 Terminal Group above, Figure 24 shows the terminal numbering sequence for all the terminals that make up Bank 1. Here you can see 3 distinct patterns each with their own numbering sequence.

6.5.7.2.2 Terminal Details Exception (cont'd)

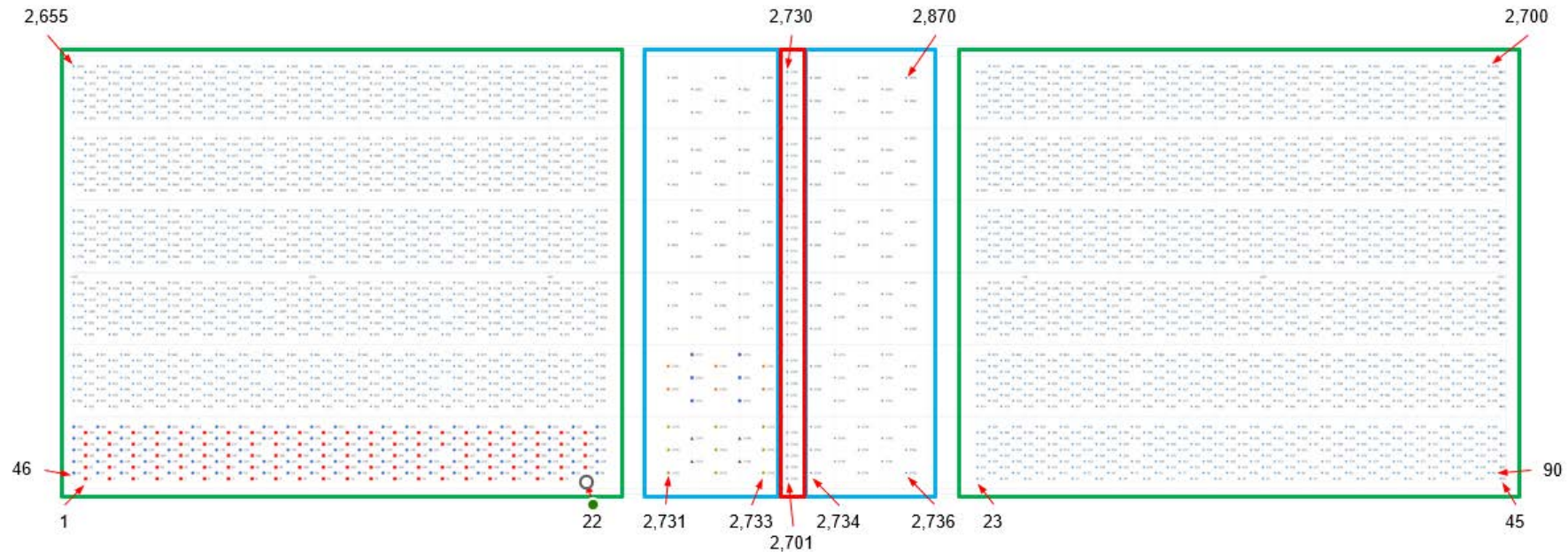


Figure 24 - Terminal Numbering Sequence for Bank 1

Pattern Group ID 3 is represented within the two extremity columns within the green outlines. Refer to Table 32 – Pattern Group Construction above for details of the Terminal patterns that makes the first pattern group which is duplicated 6 times to make the South channel before being duplicated again and mirrored to make the north channel. This pattern group has a terminal numbering sequence of Zig-Zag-Horizontal with a commencing location at the Back-left position starting at the number 1 and ends at 2700.

Pattern Group ID 10 is represented by the center red box and has a commencing location at the Back position starting at the number 2701 and ends at 2730 in the front position.

Pattern Group ID 9 is represented by the blue boxes on either side of pattern group 10 and has a commencing location at the Back-left (Southwest or Left-Bottom) position starting at the number 2731 and ends at 2870 in the Front-right (Northeast or Right-Top) position.

6.5.7.2.2 Terminal Details Exception (cont'd)

The xml representation for this terminal numbering for Bank 1 is as follows:

```
<TerminalDetail-Array>
  <TerminalNumberPattern>
    <ID>Terminal Number Pattern ID 1</ID>
    <PatternGroupID>Pattern Group ID 3</PatternGroupID>
    <FirstTerminalLocation>
      <LocationRelative-to-SelectionCenter>
        <Description>Back-left</Description>
      </LocationRelative-to-SelectionCenter>
    </FirstTerminalLocation>
    <Sequential>
      <Description>Zig-Zag-Horizontal</Description>
    </Sequential>
  </TerminalNumberPattern>
  <TerminalNumberPattern>
    <ID>Terminal Number Pattern ID 2</ID>
    <PatternGroupID>Pattern Group ID 10</PatternGroupID>
    <FirstTerminalLocation>
      <LocationRelative-to-SelectionCenter>
        <Description>Back</Description>
      </LocationRelative-to-SelectionCenter>
    </FirstTerminalLocation>
    <Sequential>
      <Description>Back-to-Front</Description>
      <NumericalSequence>
        <Start>2701</Start>
      </NumericalSequence>
    </Sequential>
  </TerminalNumberPattern>
  <TerminalNumberPattern>
    <ID>Terminal Number Pattern ID 3</ID>
    <PatternGroupID>Pattern Group ID 9</PatternGroupID>
    <FirstTerminalLocation>
      <LocationRelative-to-SelectionCenter>
        <Description>Back</Description>
      </LocationRelative-to-SelectionCenter>
    </FirstTerminalLocation>
    <Sequential>
      <Description>Zig-Zag-Horizontal</Description>
      <NumericalSequence>
        <Start>2731</Start>
      </NumericalSequence>
    </Sequential>
  </TerminalNumberPattern>
</TerminalDetail-Array>
```

Assume now that Bank 1 is just one bank of an array of banks as shown in Figure 25 below

6.5.7.2.2 Terminal Details Exception (cont'd)

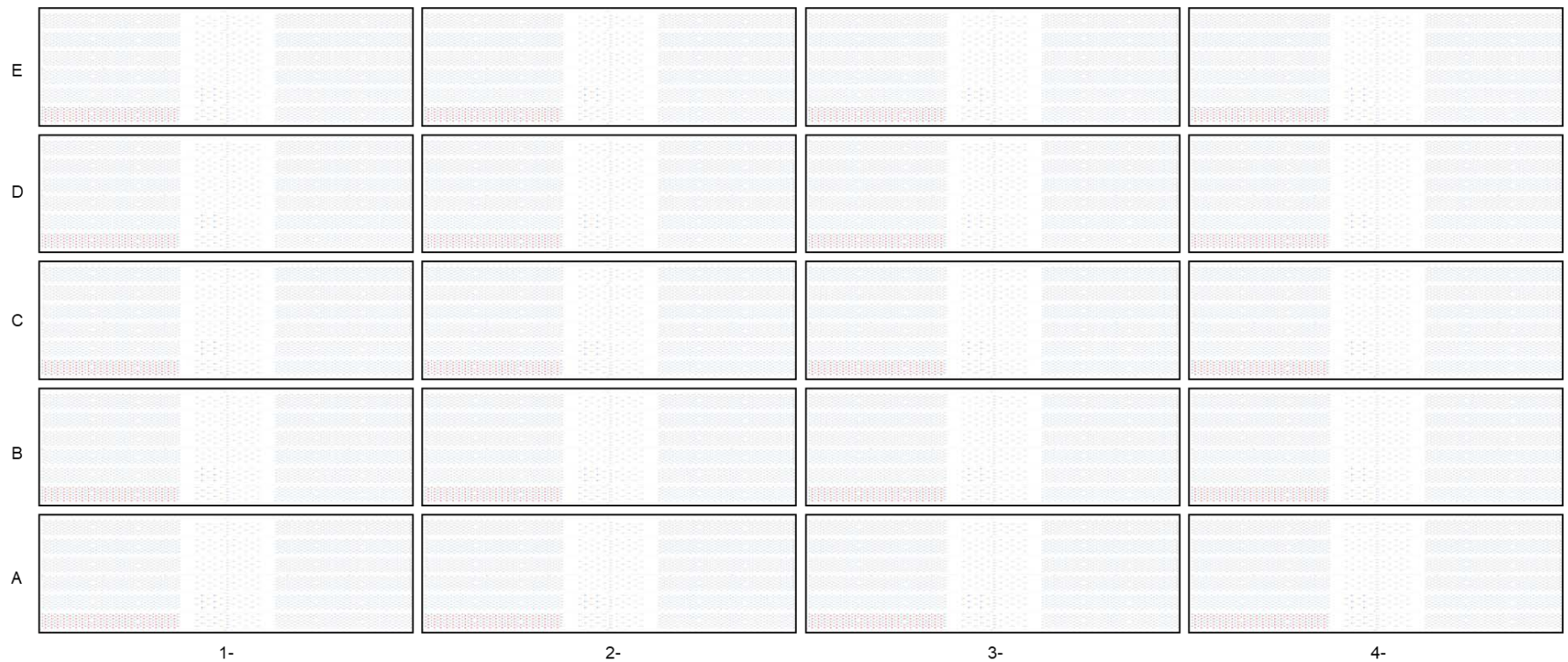


Figure 25 - An Array of Banks in a Zone

Bank 1 as defined in Figure 24 – Terminal Numbering Sequence for Bank 1 above is replicated in an array of 5 rows and 4 columns. Assume that the user wants to maintain the terminal numbering sequence as defined in Bank 1 for each of these blocks within the array, with the only difference being the pattern location within the array as the prefix before the terminal number.

6.5.7.2.2 Terminal Details Exception (cont'd)

```
<PatternGroup>
  <ID>Pattern Group ID 12</ID>
  <Name>South_Inst1-6_PtrnGrp1</Name>
  <PatternRelationship>
    <PatternGroupID>Pattern Group ID 11</PatternGroupID>
    <Duplicate>
      <nx>4</nx>
      <ny>5</ny>
      <Y-PrefixCode>A</Y-PrefixCode>
      <Y-PrefixCode>B</Y-PrefixCode>
      <Y-PrefixCode>C</Y-PrefixCode>
      <Y-PrefixCode>D</Y-PrefixCode>
      <Y-PrefixCode>E</Y-PrefixCode>
      <X-PrefixCode>1</X-PrefixCode>
      <X-PrefixCode>2</X-PrefixCode>
      <X-PrefixCode>3</X-PrefixCode>
      <X-PrefixCode>4</X-PrefixCode>
      <dx>4000.00</dx>
      <dy>3500.00</dy>
    </Duplicate>
  </PatternRelationship>
</PatternGroup>
```

Since the Y-PrefixCode is entered into the xml file before the X-PrefixCode, then the first terminal number for terminal 1 in Bank 1 in the zone position A1 becomes A1-1, The same respective terminal in the A2-1, A3-1, A4-1, B1-1, B2-1, and so on.

If nx and/or ny is a large number, then the prefix and/or suffix codes can be auto generated in a similar way as the numbering sequence is defined.

```
<PatternGroup>
  <ID>Pattern Group ID 12</ID>
  <Name>South_Inst1-6_PtrnGrp1</Name>
  <PatternRelationship>
    <PatternGroupID>Pattern Group ID 11</PatternGroupID>
    <Duplicate>
      <nx>4</nx>
      <ny>5</ny>
      <Y-PrefixColumnCode>
        <AlphabeticalSequence/>
        <Back-to-Front/>
      </Y-PrefixColumnCode>
      <X-PrefixRowCode>
        <NumericalSequence>
          <Suffix>-</Suffix>
        </NumericalSequence>
        <Left-to-Right/>
      </X-PrefixRowCode>
      <dx>4000.00</dx>
      <dy>3500.00</dy>
    </Duplicate>
  </PatternRelationship>
</PatternGroup>
```

6.5.7.2.2 Terminal Details Exception (cont'd)

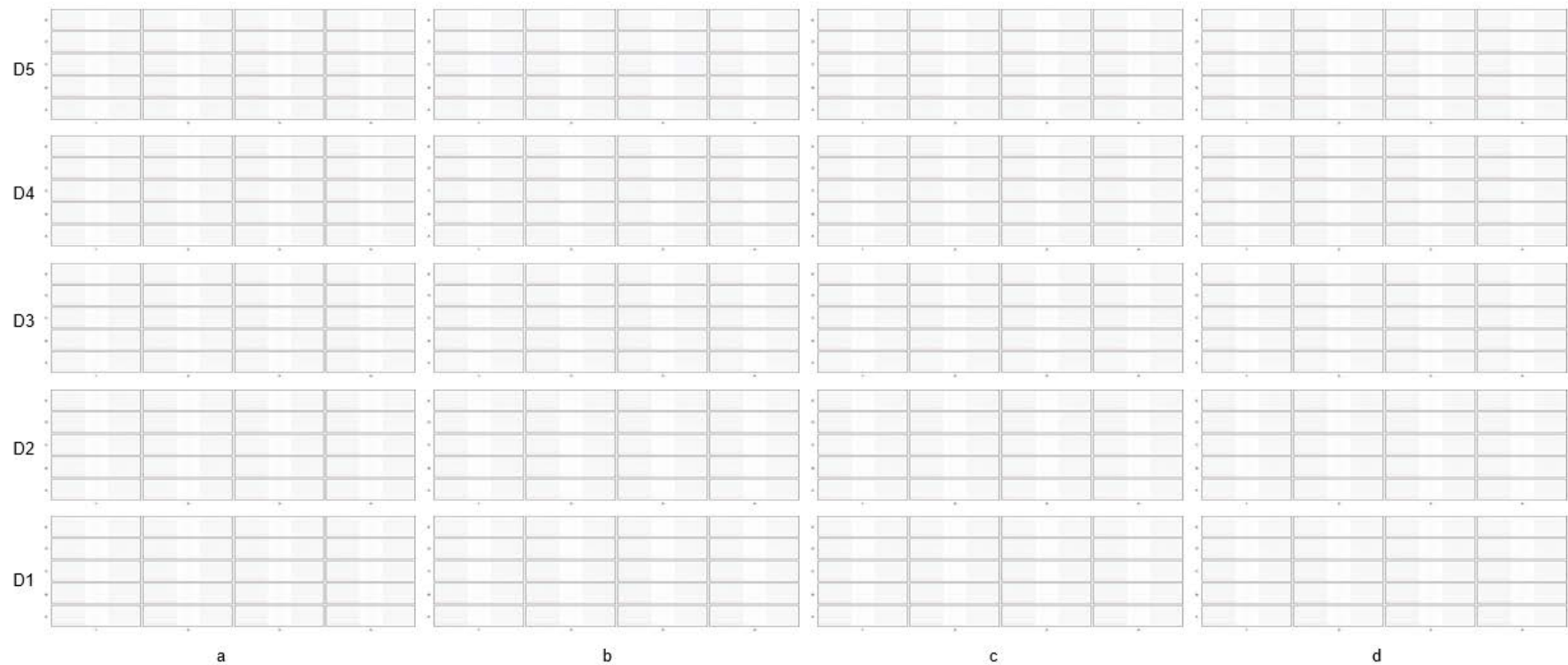


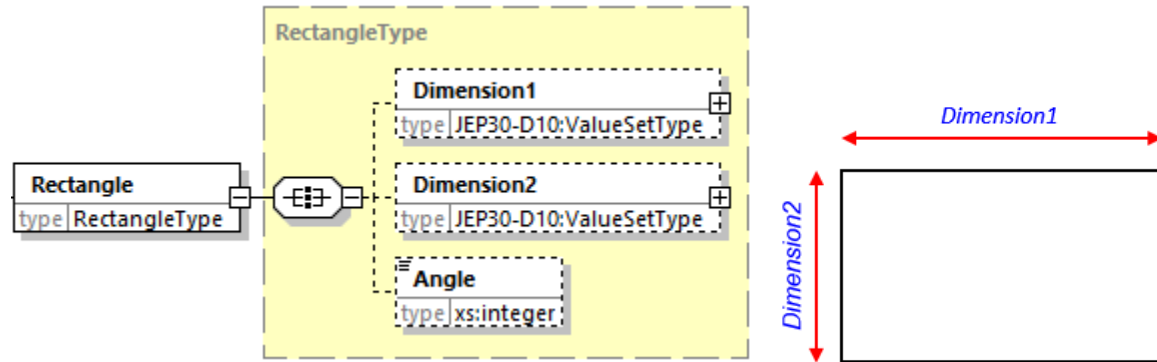
Figure 26 - An Array of Zones in One District

Similar to the concepts of Banks within a Zone, an array of Zones can be embedded within a District. Each hierarchical level in the nesting of pattern groups brings with it a corresponding prefix/suffix that gets appended to the reference label of the terminal number. Therefore, in District D1a, Zone A1- the first terminal now becomes D1aA1-1.

Annex A (informative) Shape Definitions & Dimensions

A.1 Shape Definitions

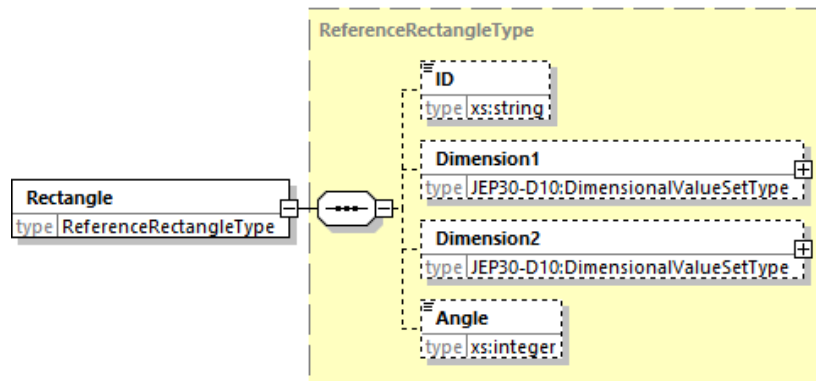
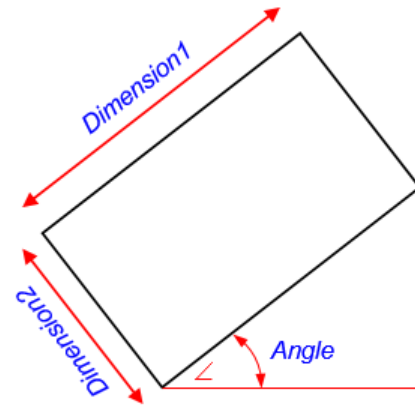
A.1.1 Rectangle



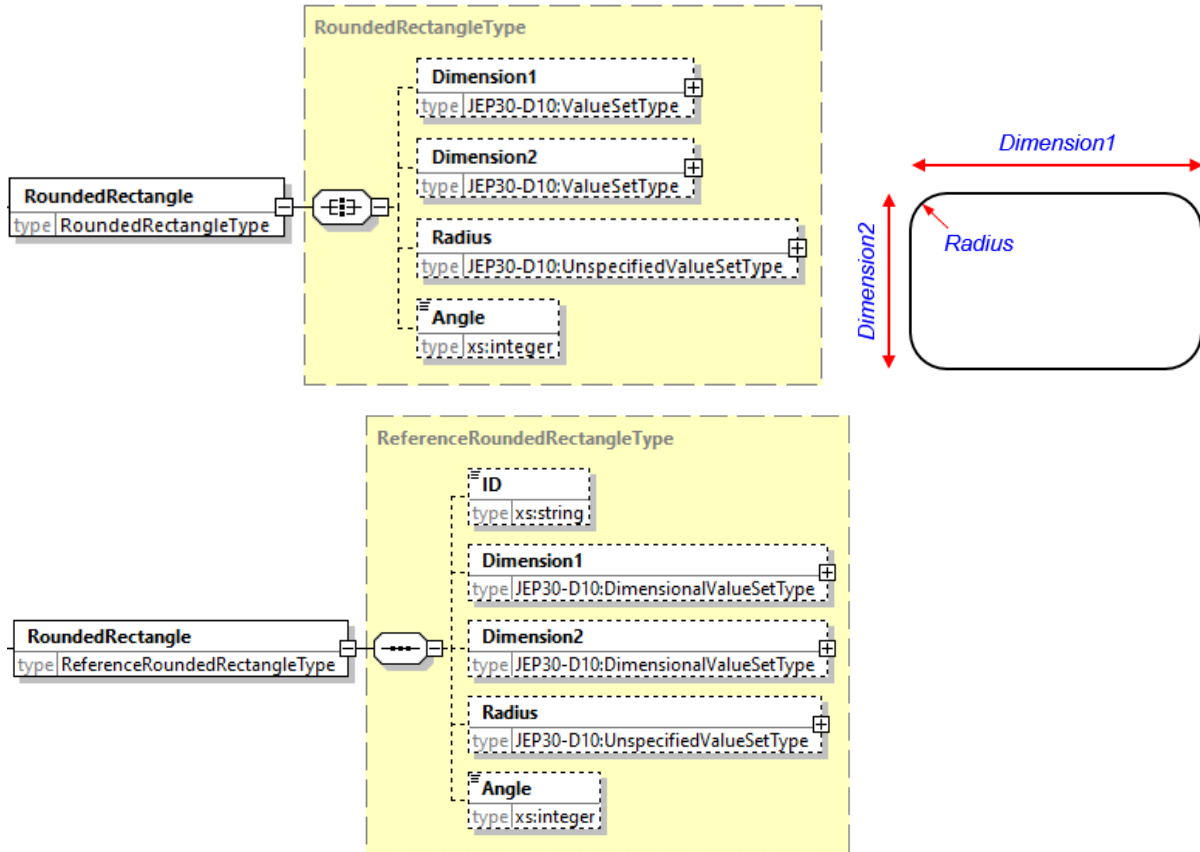
Dimension1 is always in the x direction, horizontal direction, or in the direction from left-to-right.

Dimension2 is always in the y direction, vertical direction, or in the direction from back-to-front.

The **Angle** is measured with respect to 3 o'clock or the horizontal line and is measured in a counter clockwise direction so 12 o'clock is considered 90 degrees, 9 o'clock is considered 180 degrees and 6 o'clock is considered 270 degrees.



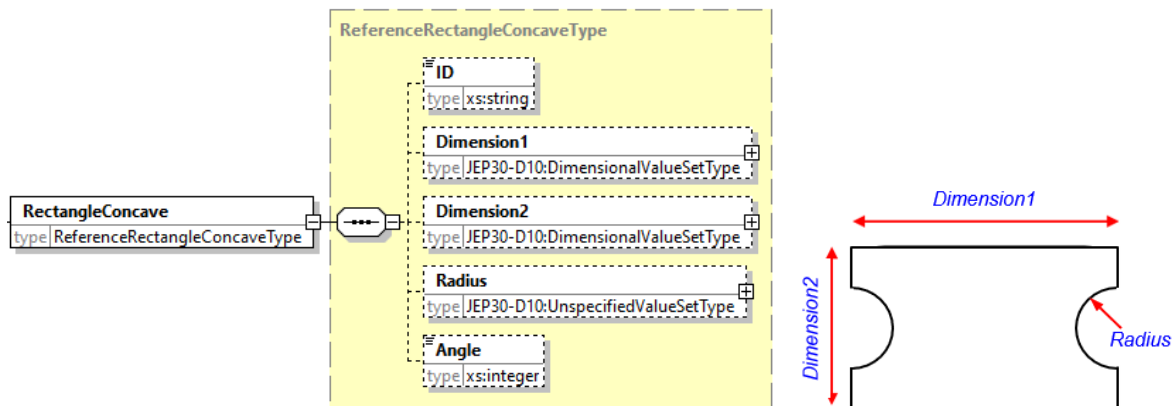
A.1.2 Rounded Rectangle



Dimension1, *Dimension2* and *Angle* are the same as for *Rectangle*.

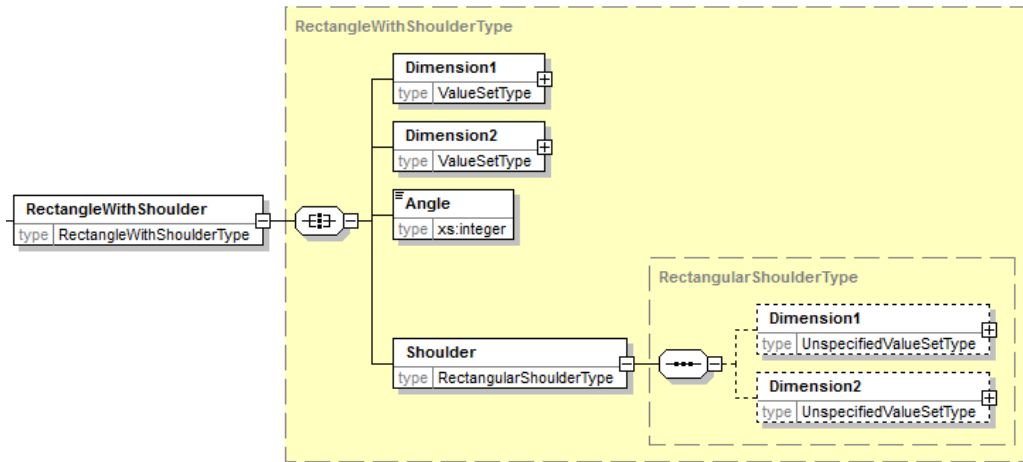
If all 4 corners have the same radius, then *RoundedRectangle* can be used. If the radius is different in any one corner, then *ModifiedRectangle* should be used.

A.1.3 Rounded Concave



Dimension1, *Dimension2* and *Angle* are the same as for *Rectangle*.

A.1.4 Rectangle with Shoulder

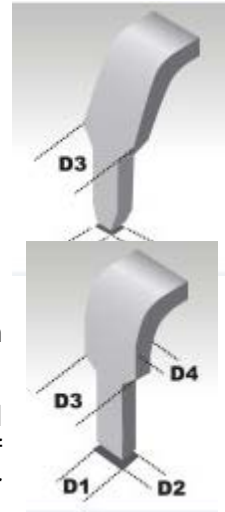


Some terminals have shoulders to control the depth of insertion into the printed board. Because hole diameter on the printed board is dependant upon many factors, one of which is the thickness of the printed board, due to aspect ratio due to ensure hole fill, some of these parts are unsuitable for thicker printed boards. If the hole size increases to be greater than the shoulder dimension, then the part can fall deeper into the hole. For this and similar reasons, shoulder dimensions should be captured.

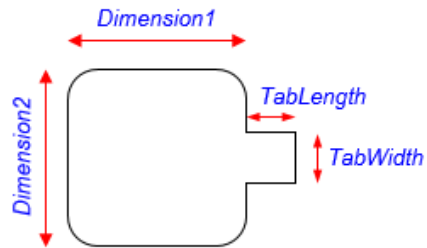
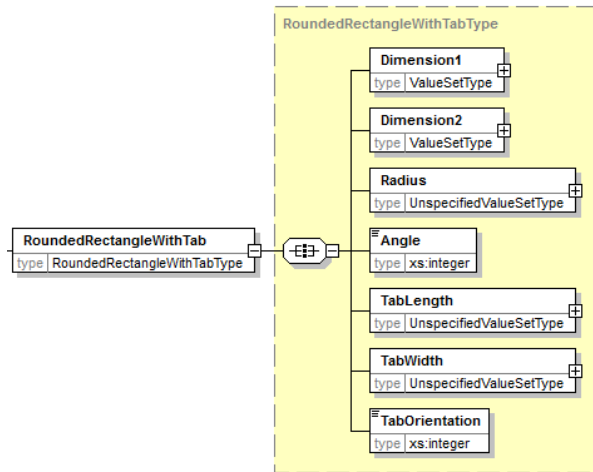
The [RectangleWithShoulder/Dimension1](#) (D1) is in the same direction as [RectangleWithShoulder/Shoulder/Dimension1](#) (D3).

Similarly the [RectangleWithShoulder/Dimension2](#) (D2) is in the same direction as [RectangleWithShoulder/Shoulder/Dimension2](#) (D4)

The [RectangleWithShoulder/Shoulder/Dimension1](#) and [RectangleWithShoulder/Shoulder/Dimension2](#) are optional, since either of these dimensions as [RectangleWithShoulder/Dimension1](#) or [RectangleWithShoulder/Dimension2](#), as can be seen from the 2nd image where D2 = D4. However this is not always the case, and if different, but unspecified, then the [Unspecified](#) element should be set under the [UnspecifiedValueSetType](#) as opposed to not populating the branches under the [Shoulder](#).



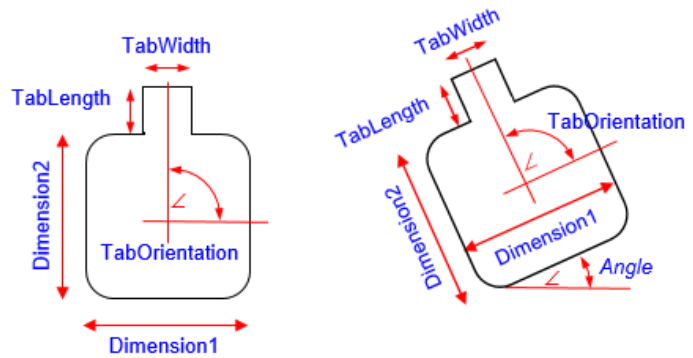
A.1.5 Rounded Rectangle with Tab



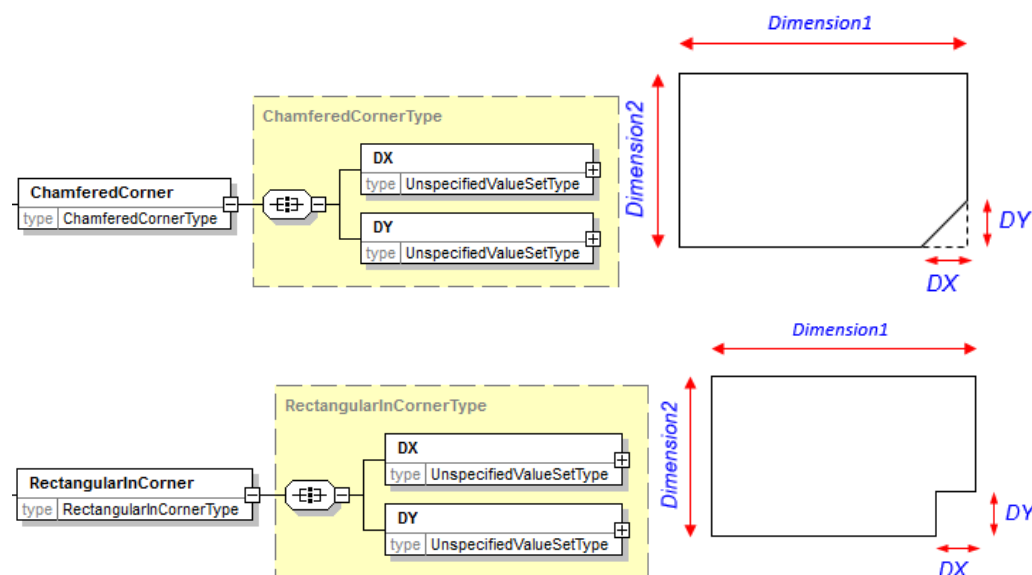
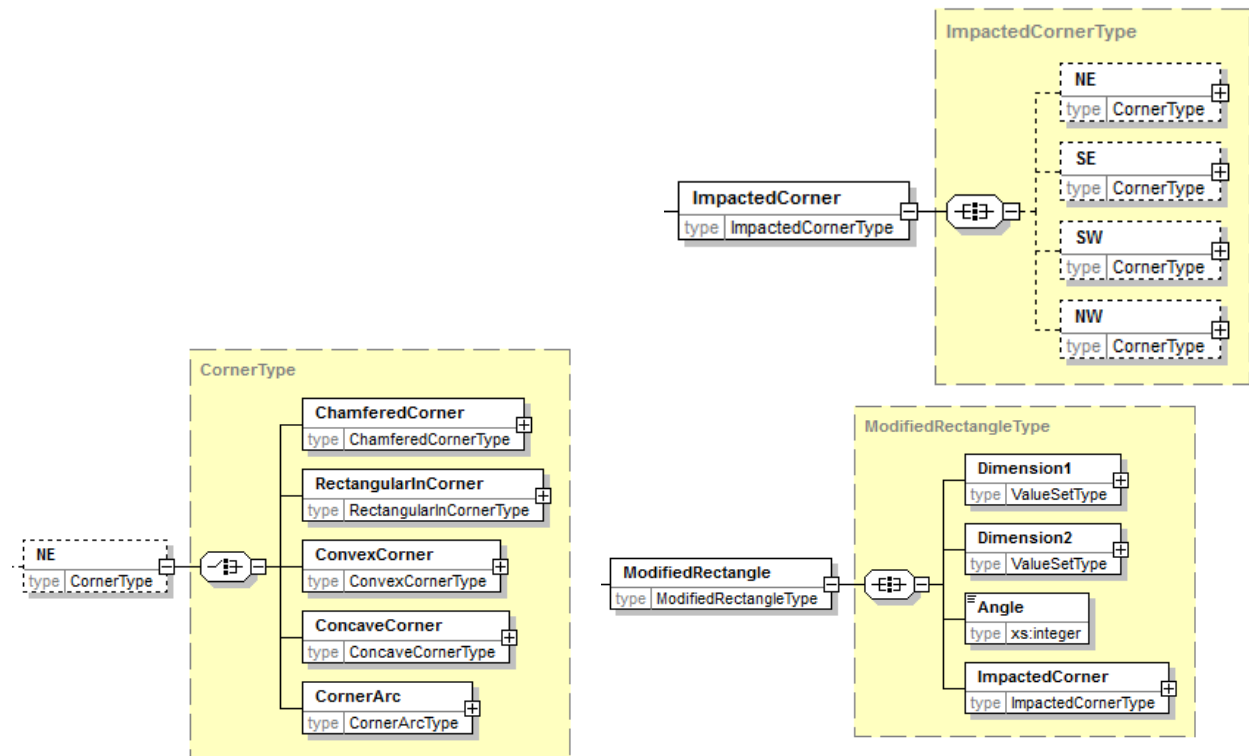
Irrespective of the orientation of the Tab, the **TabLength** is considered the direction away from the package body, while the **TabWidth** is considered the width of the tab parallel to the package body. The **TabOrientation** here is at 0 degrees, since the orientation follows the same

rules as the **Angle**.

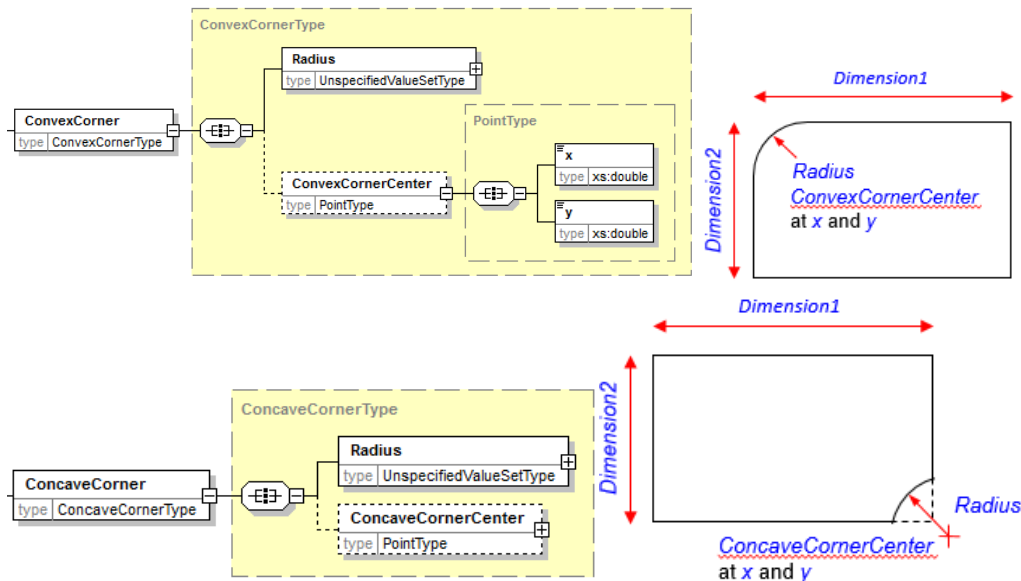
The **TabOrientation** is still captured with respect to the package body, In these 2 images, the **TabOrientation** is at 90° even if the package is rotated at an **Angle**.



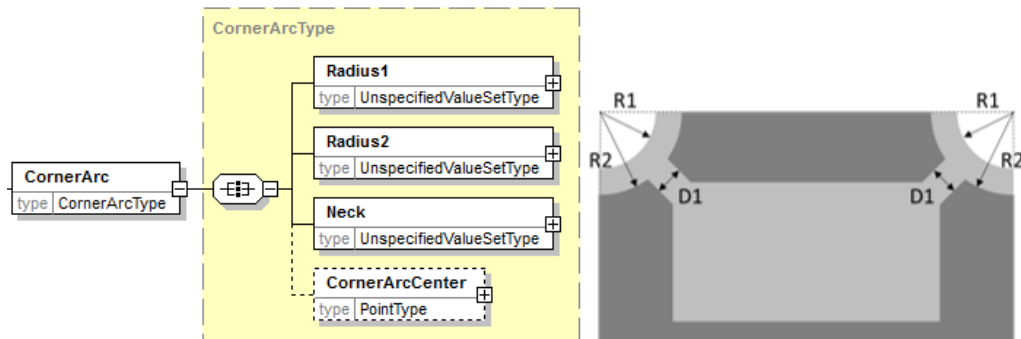
A.1.6 Modified Rectangle



A.1.6 Modified Rectangle (cont'd)



Note that the **ConcaveCornerCenter** point of **x** and **y** is not necessary to be at the same point as the corner of the **ModifiedRectangle**. If unspecified, then it is defaulted to the corner of the **ModifiedRectangle**.

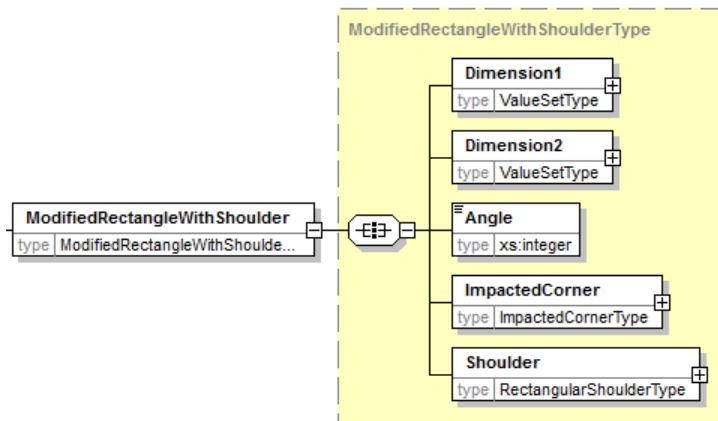


A **ModifiedRectangle** with **CornerArc** usually occurs when the Surface terminal is pulled back from the edge of the Package outline. In the majority of cases including those where the data is unspecified on the datasheet, we can assume that the center of the Arc from which (**R1**) **Radius1** and (**R2**) **Radius2** are defined, occurs at the linear intersection of the projection of the sides of the package body. This image shows that there are 2 **CornerArcs** connected to the same surface terminal in the **NW** and **NE** corners.

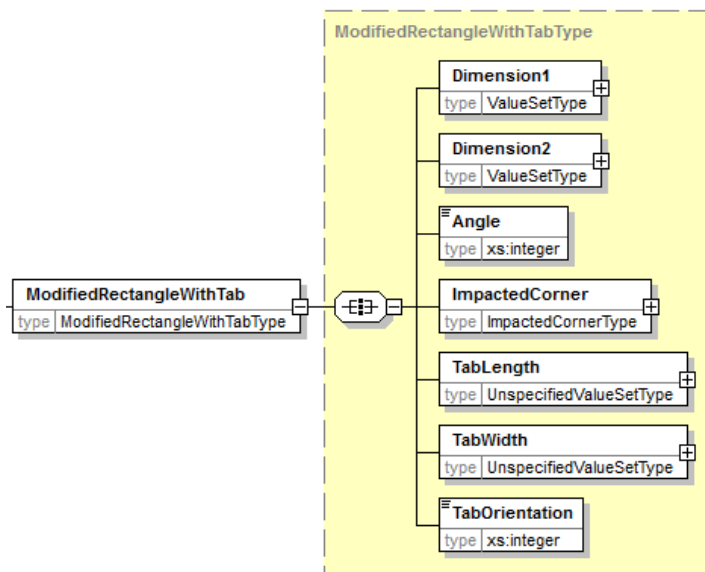
The trace **Neck** width (**D1**) connects the terminal shape to the arc along the line that would intersect with the center of the arc to the projected corner of the terminal shape.

Note that the **ConcaveArcCenter** point of **x** and **y** is not necessary to be at the same point as the corner of the **ModifiedRectangle**. If unspecified, then it is defaulted to the corner of the **ModifiedRectangle**.

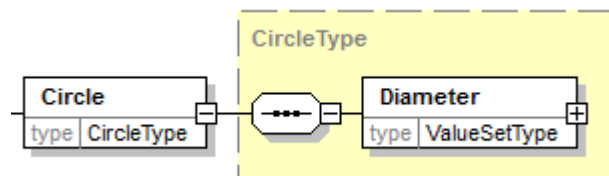
A.1.7 Modified Rectangle with Shoulder



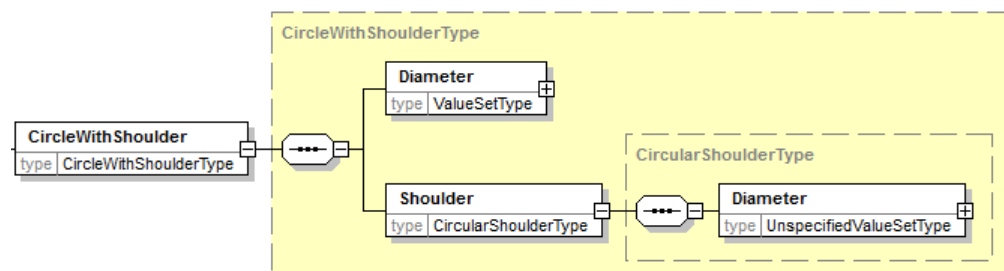
A.1.8 Modified Rectangle with Tab



A.1.9 Circle



A.1.10 Circle with Shoulder

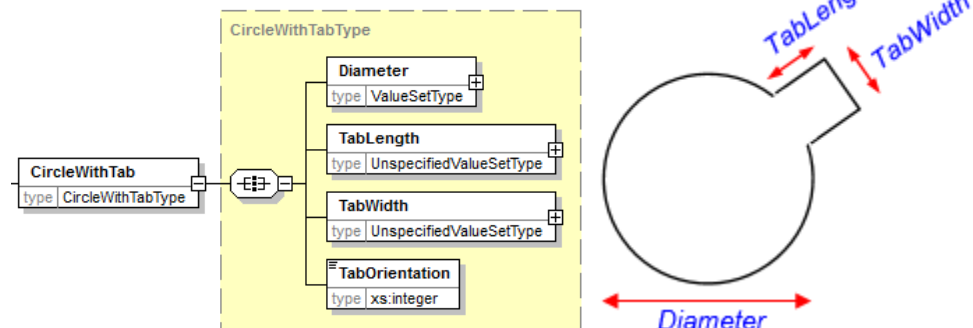


The *CircleWithShoulder/Diameter* (D1) is the diameter of the terminal that inserts into the printed board.

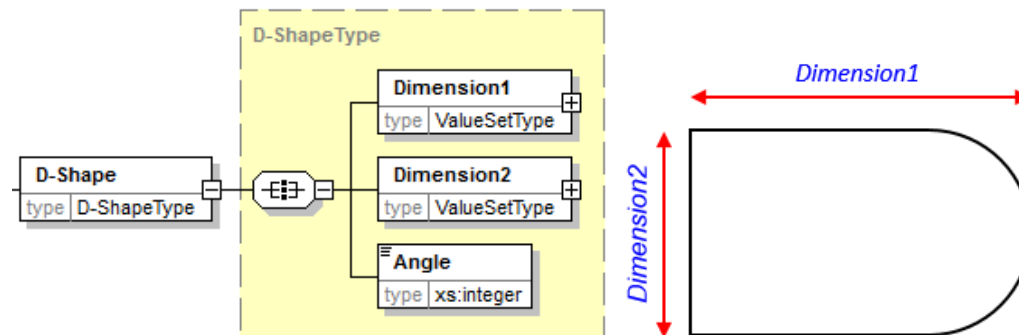
The *CircleWithShoulder/Shoulder/Diameter* (D2) is the diameter of the terminal shoulder that is typically not intended for insertion into the printed board.



A.1.11 Circle with Tab



A.1.12 D-Shape

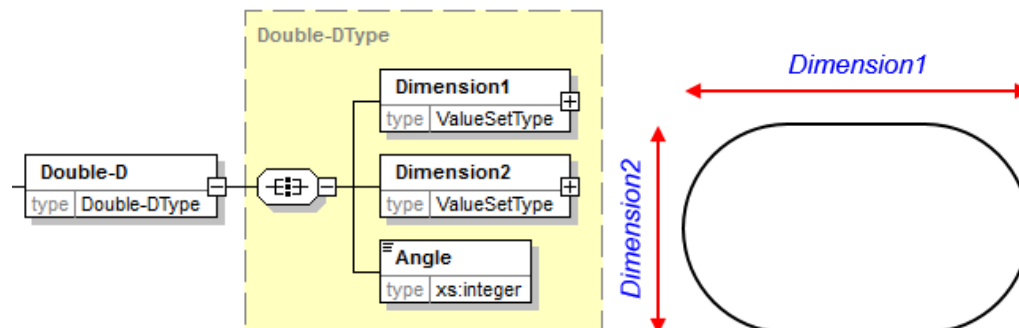


A *D-Shape* is where a semi-circle is attached to the end of a Rectangular shape. Note that radius is not required since it is 50% of *Dimension2*.

Dimension1 is the distance from the end of the rectangle to the outside edge of the semicircle.

The side with the curved end is defaulted towards the package center. If the curve is on a different side, then Modified Rectangle should be used.

A.1.13 Double-D

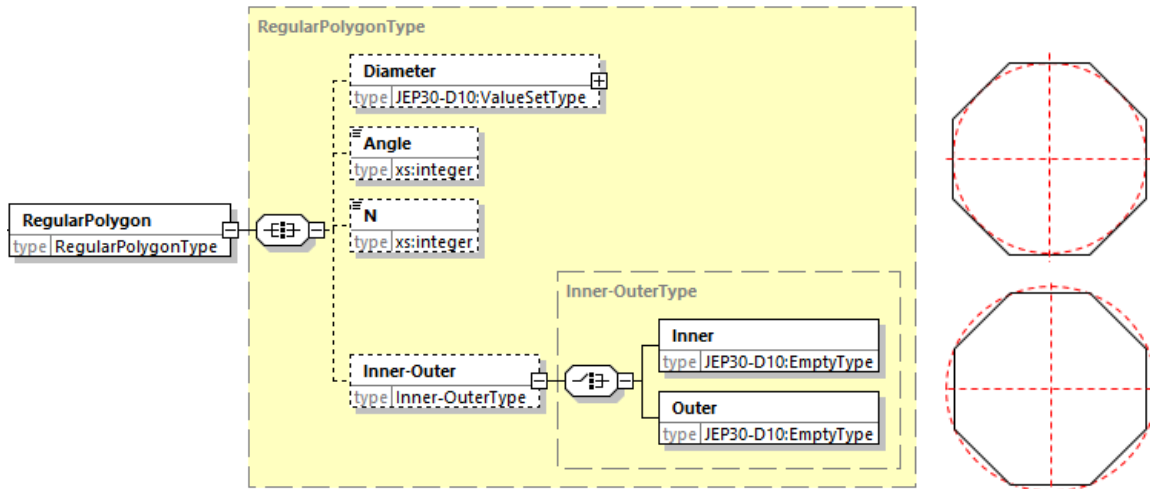


A *Double-D* shape is where a semi-circle is attached to the end of a Rectangular shape. Note that radius is not required since it is 50% of *Dimension2*.

Dimension1 is the distance from the outside edges of each of the semicircle at opposite ends of the rectangle.

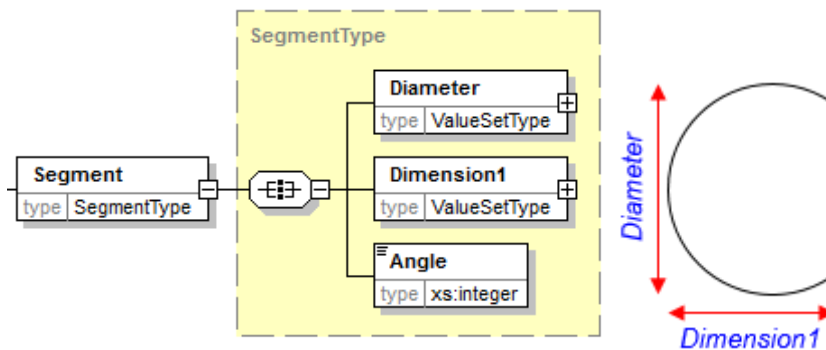
The curve sides occur on the shortest side of Dimension 1 or Dimension 2.

A.1.14 Regular Polygon



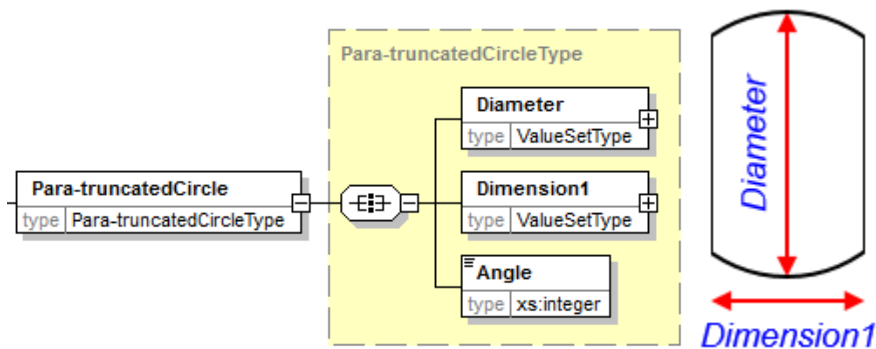
The most common Polygons required to represent package body shapes are **RegularPolygon**, that are equiangular (all its corner angles are equal), Cyclic (all its corners lie on a single circle) and equilateral (all its edges are the same length). These Polygons are also tangential (all sides are tangent to an inscribed circle and are convex (all angles are < 180 degrees)). Polygons dimensions can be captured by knowing the number of sides **N** and the **Diameter** of either the circumcircle (**Outer** circle that intersect with the corners of the polygon) or the inscribed circle (**Inner** circle which is tangential to all the sides within the polygon). All other forms of Polygons component shapes must be drawn via the **Contour** branch.

A.1.15 Segment



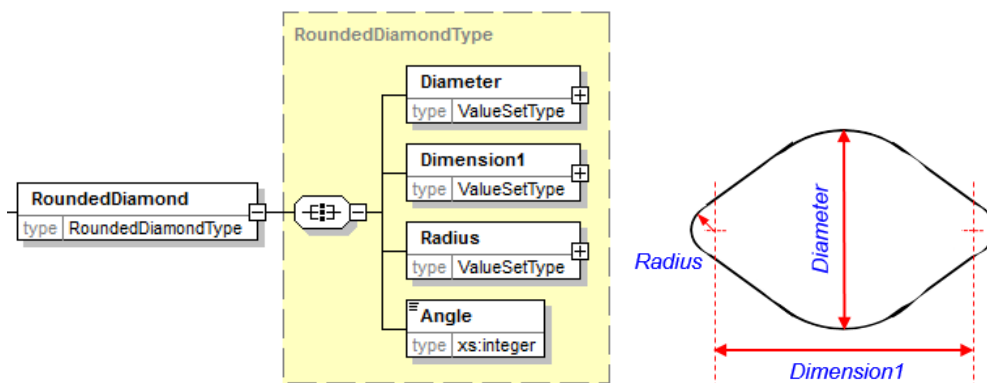
The **Angle** here defines the rotation in the counter clockwise direction of the straight edge from the 3 o'clock position. The image here shows 90°.

A.1.16 Para-truncated Circle



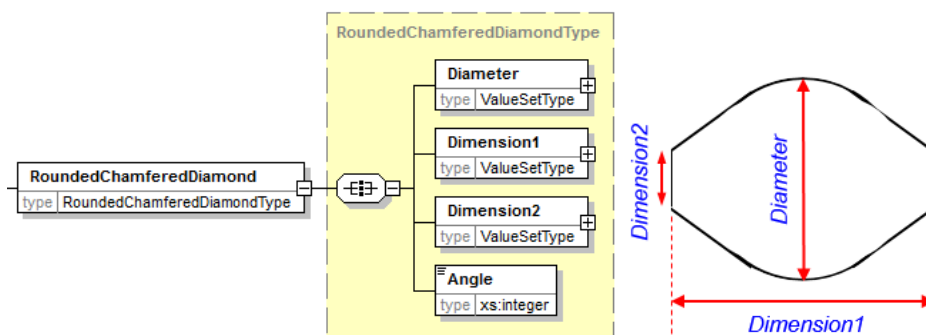
A *Para-truncatedCircle* is a circle with equal segments removed from opposite's sides.

A.1.17 Rounded Diamond



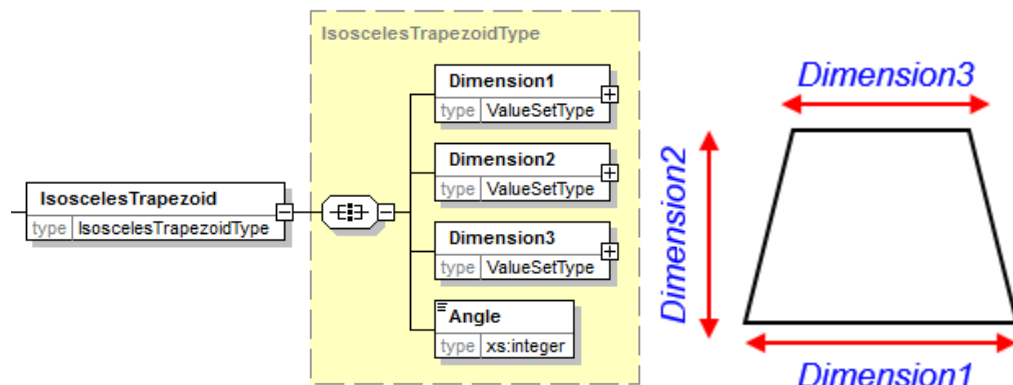
Dimension1 specifies the distance between the centers of the 2 end circles, while the *Radius* specifies the curvature of the 2 end circles. The *Diameter* specifies the dimension of the center circle. The 4 straight sides intersect tangentially with their respective two circles.

A.1.18 Rounded Chamfered Diamond

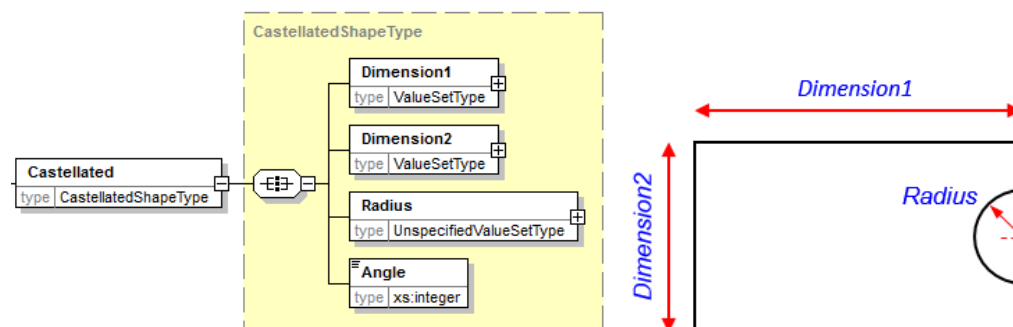


Dimension1 specifies the distance between the end straight edges, while *Diameter* specifies the dimension of the center circle. The straight sides have a length of *Dimension2*.

A.1.19 Isosceles Trapezoid

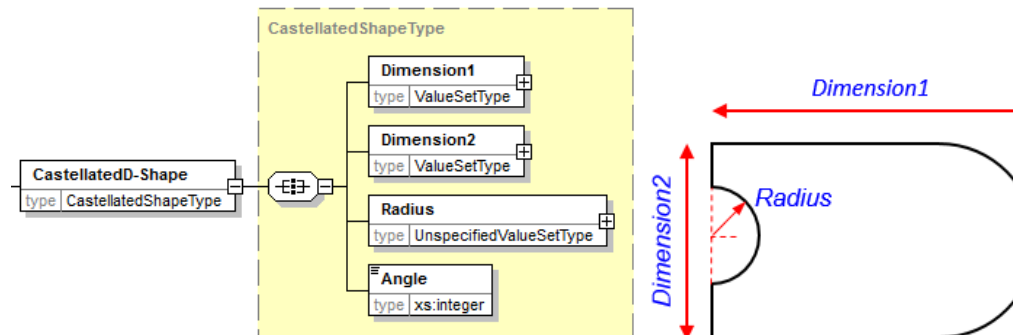


A.1.20 Castellated Shape

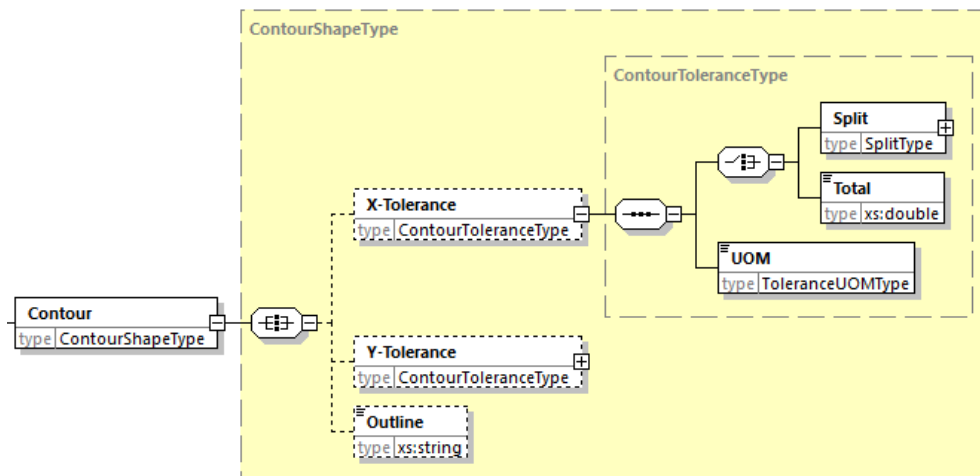


Castellated is shape with a recess in it. The **Radius** of this concave is less than the width of the shape side to which it is attached. The recess is also centered along that side of the shape.

A.1.21 Castellated D-Shape Shape



A.1.22 Contour



The **Outline** is a list of points representing the outline of the shape. In the XML it is defined as a string. This string must have a specific structure. The structure is $(X_1, Y_1, R_1), (X_2, Y_2, R_2), \dots (X_n, Y_n, R_n)$. The points are defined such that the $(0, 0)$ location of the resulting shape is the reference point of the shape. Contour shaped terminal contacts that are positioned randomly will be positioned using the $(0, 0)$ location of the terminal.

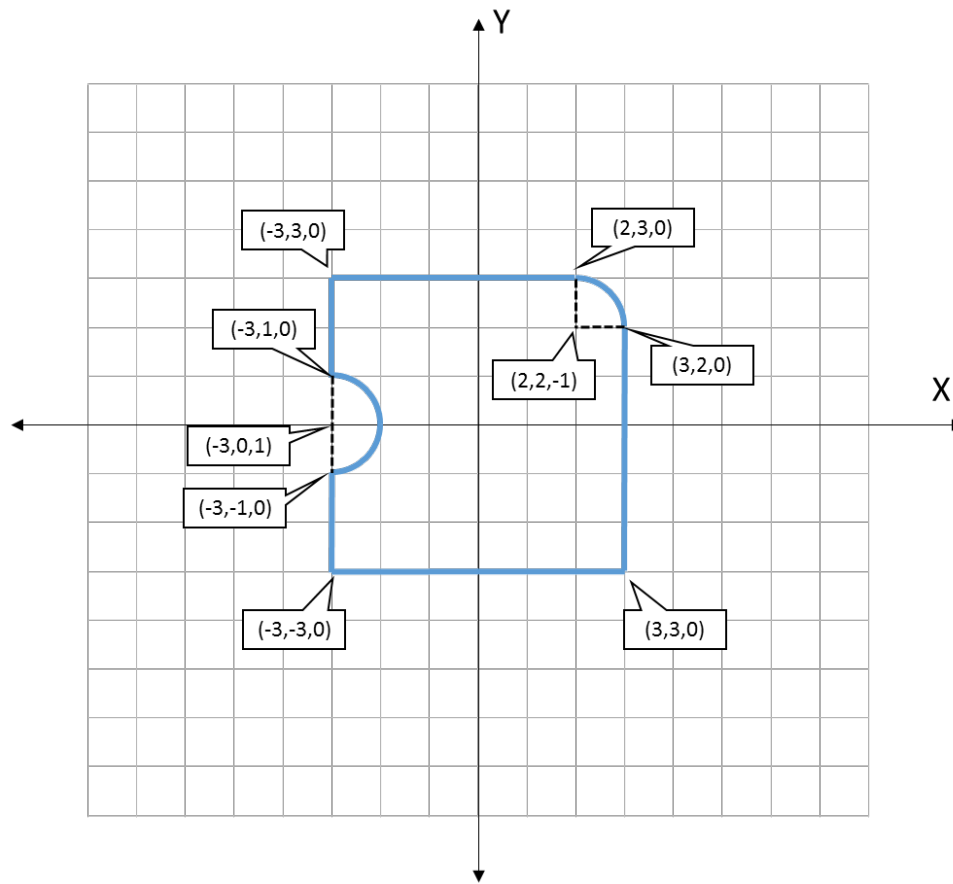
Non-arcs points are defined with an X, Y coordinate, The R value is always 0.

Arcs are defined with 3 points.

- The first point is the start point of the arc (R value is 0).
- The second point is the center point of the arc. The R value for this point is equal to the radius of the arc. If the R value is negative, a counter clockwise arc is defined. If the R value is positive, a clockwise arc is defined.
- The third point is the end point of the arc (R value is 0).

The last point does not need to be repeated in the case of Segments as all are closed shapes. You may begin with any point except a radius point. The points must be order properly from that starting point.

A.1.22 Contour (cont'd)



Example 1. The string value that will be present in *Outline* to represent this shape would be

- $(-3,-3,0),(3,-3,0),(3,2,0),(2,2,-1),(2,3,0),(-3,3,0),(-3,1,0),(-3,0,1),(-3,-1,0)$

A.1.23 Reference Terminal Contour

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalShape/ModifiedCorner |
| diagram | |
| type | ModifiedCornerType , ImpactedTerminalType , TerminalCenterType , Impact-to-TerminalGroupType , Apply-to-all-TerminalsType , SymmetryType , ImpactedCornerType , CornerType . |

A.1.24 Modified Corner

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalShape/ModifiedCorner |
| diagram | |
| type | ModifiedCornerType , ImpactedTerminalType , TerminalCenterType , Impact-to-TerminalGroupType , Apply-to-all-TerminalsType , SymmetryType , ImpactedCornerType , CornerType . |

A.1.24.1 Impacted Terminal

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalShape/ModifiedCorner/ImpactedTerminal |
| diagram | |
| type | ImpactedTerminalType, TerminalCenterType, JEP30-D10:PointXYType. |

A.1.24.2 Impact – to – Terminal Group

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalShape/ModifiedCorner/Impact-to-TerminalGroup |
| diagram | |
| type | Impact-to-TerminalGroupType, Apply-to-all-TerminalsType, SymmetryType, SymmetryRotationType, ReflectionType, |

A.1.24.2.1 Apply – to – all - Terminals

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalShape/ModifiedCorner/ShapeImpactedCorner |
| diagram | |
| type | Apply-to-all-TerminalsType , CornerImpact-to-StandardArrayType , JEP30-D10:EmptyType . |

A.1.24.2.2 Rotation

| | |
|---------|--|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalShape/ModifiedCorner/Impact-to-TerminalGroup/Symmetry/Rotation |
| diagram | |
| type | SymmetryRotationType , SymmetryRotationAxisType , SymmetryRotationCenterType , JEP30-D10:EmptyType . |

A.1.24.2.3 Reflection

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalShape/ModifiedCorner/Impact-to-TerminalGroup/Symmetry/Reflection |
| diagram | <p>The diagram illustrates the class hierarchy for ReflectionType. A dashed box labeled 'ReflectionType' contains three main components: 'MirrorPlane', 'InversionCenter', and 'ReflectionAxisType'. 'MirrorPlane' is a class with a type constraint 'type ReflectionAxisType'. 'InversionCenter' is a class with a type constraint 'type ReflectionInversionCenterType'. 'ReflectionAxisType' is a class with three subclasses: 'xyPlane', 'yzPlane', and 'xzPlane', each with a type constraint 'type JEP30-D10:EmptyType'. 'ReflectionInversionCenterType' is a class with two subclasses: 'TerminalCenter' and 'PackageBodyCenter', each with a type constraint 'type JEP30-D10:EmptyType'. The 'Reflection' class, located outside the 'ReflectionType' box, has a type constraint 'type ReflectionType' and is connected to the 'ReflectionType' box by a solid line with an open circle at the 'Reflection' end and a solid line with a filled circle at the 'ReflectionType' end.</p> |
| type | ReflectionType, ReflectionAxisType, JEP30-D10:EmptyType, ReflectionInversionCenterType. |

A.1.24.3 Shape Impacted Corner

| | |
|---------|---|
| path | PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalShape/ModifiedCorner/ShapeImpactedCorner |
| diagram | <p>The diagram illustrates the structure of the ShapeImpactedCorner class and its relationship to the ImpactedCornerType class. The ShapeImpactedCorner class is shown on the left, with a type field of ImpactedCornerType. It is connected to the ImpactedCornerType class via a composition relationship (indicated by a solid line with a filled diamond at the ShapeImpactedCorner end). The ImpactedCornerType class is shown on the right, with a type field of CornerType. It contains several subclasses: ChamferedCorner (type ChamferedCornerType), RectangularInCorner (type RectangularInCornerType), ConvexCorner (type ConvexCornerType), ConcaveCorner (type ConcaveCornerType), and CornerArc (type CornerArcType). Additionally, the ImpactedCornerType class contains three subclasses: NE (type CornerType), SE (type CornerType), and SW (type CornerType). The NE, SE, and SW subclasses are connected to the ImpactedCornerType class via composition relationships. The NE, SE, and SW subclasses are also connected to the ImpactedCornerType class via composition relationships.</p> |
| type | ImpactedCornerType, CornerType, ChamferedCornerType, RectangularInCornerType, ConvexCornerType, ConcaveCornerType, CornerArcType. |

A.1.24.4 Corner Type

| | |
|---------|---|
| path | <p>PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalShape/ModifiedCorner/ShapelImpactedCorner/NE</p> <p>PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalShape/ModifiedCorner/ShapelImpactedCorner/SE</p> <p>PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalShape/ModifiedCorner/ShapelImpactedCorner/SW</p> <p>PartModel/PackageSection/Package-Array/Package/TerminalGroups/TerminalGroup-Array/TerminalGroup/TerminalShape/ModifiedCorner/ShapelImpactedCorner/NW</p> |
| diagram | <pre> classDiagram class CornerType { type CornerType } class ChamferedCorner { type ChamferedCornerType } class RectangularInCorner { type RectangularInCornerType } class ConvexCorner { type ConvexCornerType } class ConcaveCorner { type ConcaveCornerType } class CornerArc { type CornerArcType } class ChamferedCornerType { DX JEP30-D10:UnspecifiedValueSetType DY JEP30-D10:UnspecifiedValueSetType } class RectangularInCornerType { DX JEP30-D10:UnspecifiedValueSetType DY JEP30-D10:UnspecifiedValueSetType } class ConvexCornerType { Radius JEP30-D10:UnspecifiedValueSetType ConvexCornerCenter JEP30-D10:PointXYType } class ConcaveCornerType { Radius JEP30-D10:UnspecifiedValueSetType ConcaveCornerCenter JEP30-D10:PointXYType } class CornerArcType { Radius1 JEP30-D10:UnspecifiedValueSetType Radius2 JEP30-D10:UnspecifiedValueSetType Neck JEP30-D10:UnspecifiedValueSetType CornerArcCenter JEP30-D10:PointXYType } CornerType -- > ChamferedCorner CornerType -- > RectangularInCorner CornerType -- > ConvexCorner CornerType -- > ConcaveCorner CornerType -- > CornerArc </pre> <p>The diagram illustrates the structure of the CornerType and its subclasses. The CornerType is a base class with a type attribute of type CornerType. It has five subclasses: ChamferedCorner, RectangularInCorner, ConvexCorner, ConcaveCorner, and CornerArc. Each subclass has its own set of attributes and is associated with a specific CornerType (e.g., ChamferedCornerType, RectangularInCornerType, etc.). The attributes for each subclass are as follows:</p> <ul style="list-style-type: none"> ChamferedCorner: DX (type: JEP30-D10:UnspecifiedValueSetType), DY (type: JEP30-D10:UnspecifiedValueSetType). RectangularInCorner: DX (type: JEP30-D10:UnspecifiedValueSetType), DY (type: JEP30-D10:UnspecifiedValueSetType). ConvexCorner: Radius (type: JEP30-D10:UnspecifiedValueSetType), ConvexCornerCenter (type: JEP30-D10:PointXYType). ConcaveCorner: Radius (type: JEP30-D10:UnspecifiedValueSetType), ConcaveCornerCenter (type: JEP30-D10:PointXYType). CornerArc: Radius1 (type: JEP30-D10:UnspecifiedValueSetType), Radius2 (type: JEP30-D10:UnspecifiedValueSetType), Neck (type: JEP30-D10:UnspecifiedValueSetType), CornerArcCenter (type: JEP30-D10:PointXYType). |
| type | <p>ImpactedCornerType, CornerType, ChamferedCornerType, RectangularInCornerType, ConvexCornerType, ConcaveCornerType, CornerArcType.</p> |

A.2 Terminal Dimensions

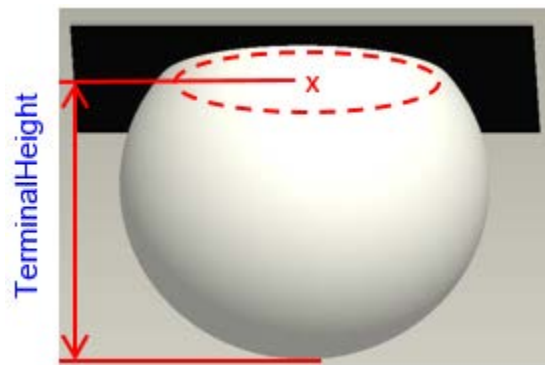
The following sections outlines the vertical dimensions that are required for each terminal.

A.2.1 Ball

The Height of the Ball is dependent upon the *Ball* type and is mandatory.

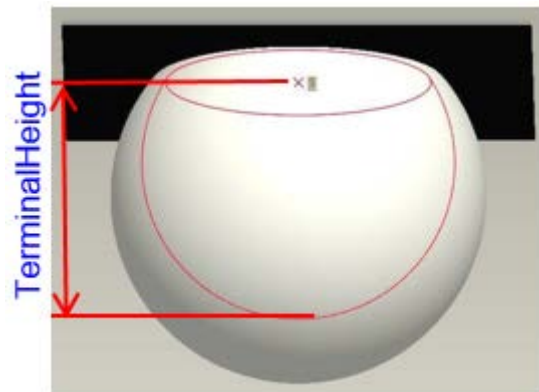
A.2.1.1 Collapsing Ball

For the *Ball/Collapsing* and for the *Ball/Bump*, the *TerminalHeight* is given as the dimension from the Seating Plane to the underside of the Package.



A.2.1.2 Non-collapsing Ball

For the *Ball/Non-collapsing*, the *TerminalHeight* is given as the dimension from the Seating Plane to the underside of the Package after the Package is soldered to the PCB Substrate. A Non-collapsing Ball has a high temperature ball within a ball, so that during reflow, only the outer ball reflows and the inner ball does not. This inner ball holds the package up from the printed board and prevents it from collapsing any further. This is the height that needs to be captured.

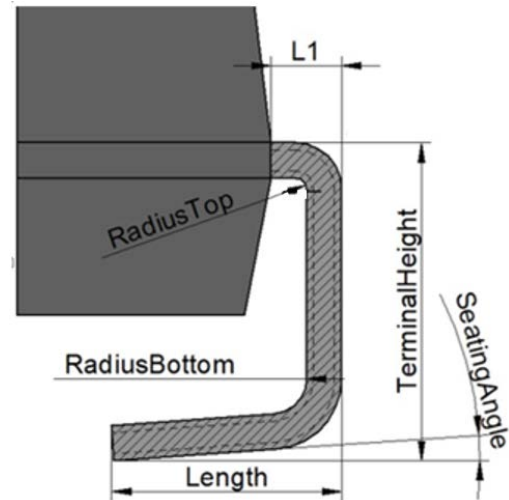


A.2.2 C-Bend

C-bend compliant terminal extend from the sides of the body, bend down, and form a flat contact area with the board under the package body. It is similar to a J-bend terminal except that the bottom of the J-terminal is flat, and not rounded up under the part body.

When terminals are in a Dual position on the device, then the dimension *Length* can be derived from alternative dimensions provided, such as

- 1) $(\text{Terminal Span} - \text{Terminal Spacing})/2$,
- 2) $\text{Terminal Span} - \text{Terminal Span Pitch}$,
- 3) etc.

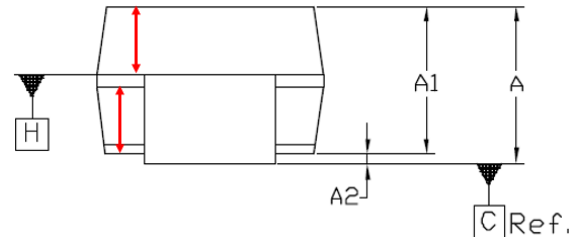


Similarly, the *PackageEdge-to-end-of-Terminal* can be derived from alternative dimensions provided if the terminals are in a Dual position, such as

- 1) $(\text{Terminal Span} - \text{Package dimension})/2$,
- 2) etc.

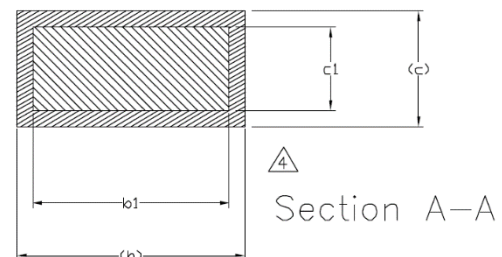
The *Width* (b) of the terminal is necessary to accurately calculate the width of the land pattern and is therefore mandatory to be captured.

TerminalHeight is the dimension from the Seating Plane to the highest point where the Terminal exits the Package Body as defined by the datum H in this drawing. The provision of this dimension will facilitate a more accurate representation of a 3D model created from this content. This data is mandatory to be provided, but if unavailable, it can be assumed that the portion of the package height above the terminal frame is also the same height of the package body below the terminal frame.



- 1) $\text{Terminal Height} = (\text{Package Body Height (A1)} - \text{Terminal Thickness})/2 + \text{Terminal Thickness} + \text{Standoff (A2)}$

TerminalThickness is defined as the thickness of the terminal (c). It is the same value on the horizontal portion as it is on the vertical portion. This is the dimension that drives the Toe of the Land Pattern in the land pattern calculation and is therefore mandatory. The Plating thickness is the value of the outer dimensions – base metal dimensions (i.e., b-b1 or c-c1 which should result in the same value).



The *RadiusTop* and *RadiusBottom* of the terminal is optional but preferred to accurately represent the 3D model of the part. When unspecified, it can be assumed to be equal to the terminal thickness.

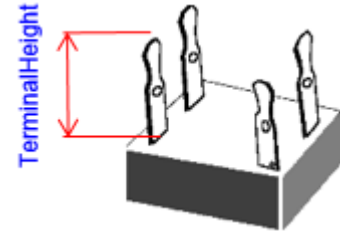
A.2.2 C-Bend (cont'd)

The *SeatingAngle* of the terminal is optional but preferred to accurately represent the 3D model of the part. When unspecified, it can be assumed to be 4 degrees, with a min of 0 degrees and a max of 8 degrees.

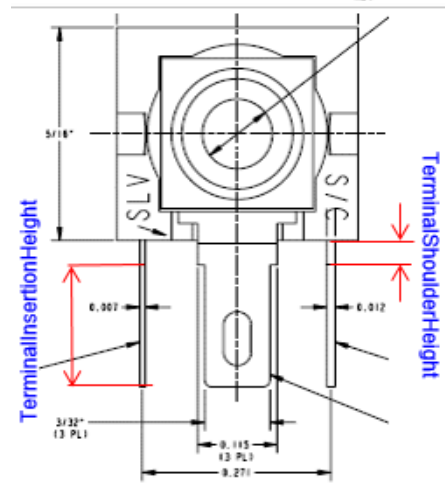
The *RiseAngle* of the terminal is optional but preferred to accurately represent the 3D model of the part. When unspecified, it can be assumed to be 90 degrees for a c-bend terminal.

A.2.3 Lug

Lugs typically are not soldered to the PCB. Instead they have mating connectors or wires connected to them. However some are soldered to the printed board and some penetrate the printed board to be connected on the other side. Therefore if the Lug is vertical off the upperside of the Package, then the *TerminalHeight* is captured as shown here.



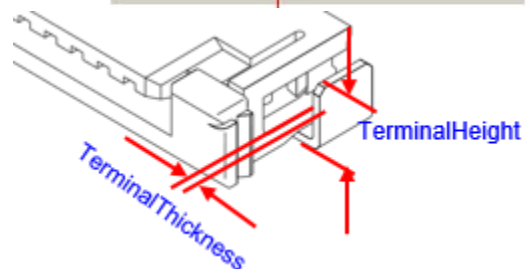
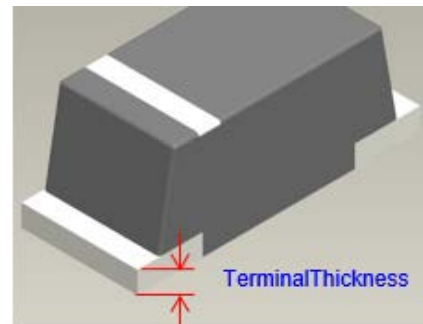
When the Terminals exit from the underside of the body, with the intention of penetrating the PCB, then the *TerminalInsertionLength* and the *TerminalShoulderHeight* are required. If either of the data is unavailable, then the User should state *Unspecified* for that respective dimension that is unavailable.



A.2.4 Flat Terminal

TerminalThickness is required for *Flat/Elevated*, *Flat/Hole*, *Flat/With-opening* terminals or for a normal *Flat* terminal.

The variant *Flat-L-bend* terminal however does have both the *TerminalThickness* plus the *TerminalHeight* requirements. The thickness of the Terminal will determine the side fillets outside the Body outline, whereas the height of the outer L-Bend will determine the Toe.

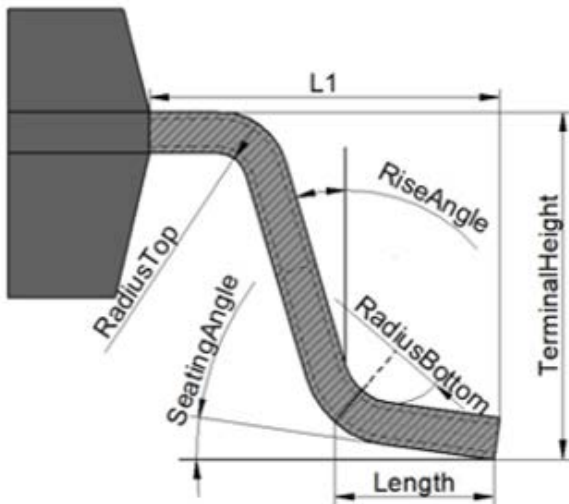


A.2.5 Gull-wing Terminal

Gull-wing compliant terminal extend from the sides of the body, bend down, and form a flat contact area with the board outside the package body. It is similar to a C-terminal except that the bottom of the J-terminal points away from the package center and not towards the package center..

As with the C-bend, when terminals are in a Dual position on the device, then the dimension *Length* can be derived from alternative dimensions provided, such as

- 1) $(\text{Terminal Span} - \text{Terminal Spacing})/2$,
- 2) $\text{Terminal Span} - \text{Terminal Span Pitch}$,
- 3) etc.

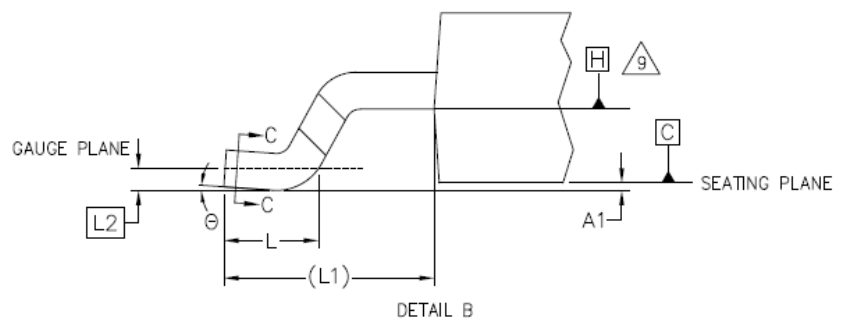
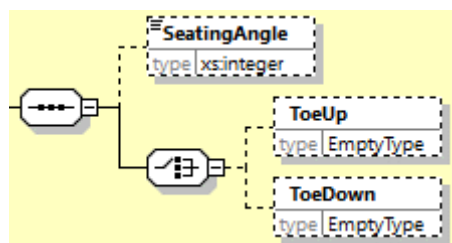


Similarly, the *PackageEdge-to-end-of-Terminal* can be derived from alternative dimensions provided if the terminals are in a Dual position, such as

- 1) $(\text{Terminal Span} - \text{Package dimension})/2$,
- 2) etc.

The elements *Width*, *TerminalHeight*, *TerminalThickness*, *RadiusTop*, *RadiusBottom*, *SeatingAngle*, and *RiseAngle*, follow the same definitions as that provided for the C-bend.

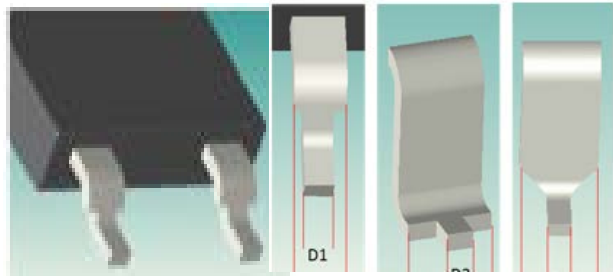
The above image shows that the Gull-wing terminal toe is pointing downwards, but an option is available for Gull-wing terminals whose Toe is pointing upwards.



The above concepts apply to all variants types of the Gull Wing Terminal with Modifications. There are several basis types of gull-wing modifications as defined herein.

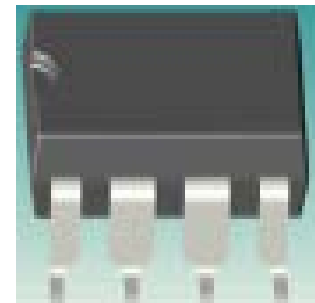
A.2.5 Gull-wing Terminal (cont'd)

When a Gull-wing has a shoulder, this is typically centered across the terminal contact area with the printed board. Notice that the variations can have the transition from the narrow portion of the terminal to the wider portion of the terminal at right angles or at a taper. The schema has the option to capture the details for either options. The schema provides 4 options to capture the dimensional details, namely:



- 1) From the end of terminal to the Start/End of the Shoulder taper transition,
- 2) From the Package edge to the Start/End of the Shoulder taper transition,
- 3) From the Seating Plane to the Start/End of the Shoulder taper transition,
- 4) From the top of the Terminal Exit from the Package to the Start/End of the Shoulder taper transition.

A DIP TH terminal can be prepped to be surface mounted as opposed to be TH mounted. In this situation, the element *TH-PreppedShoulder* should be selected.



For Inner and Outer configurations, we can assume that the edge of the terminal contact area aligns perfectly with the edge of the neck (or Shoulder). No Offset needs to be specified.

When the edge of the terminal contact area does not align perfectly with the edge of the neck (or Shoulder), then the *Offset* can be set under the *Configuration*.



Configuration = Inward



Configuration = Center

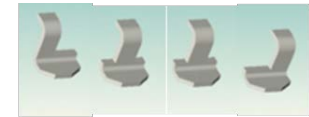
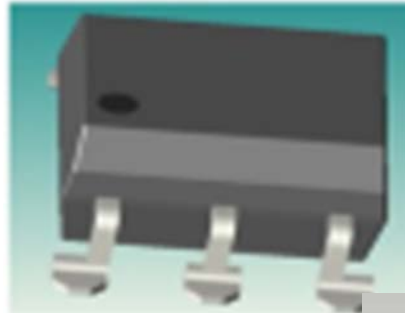


Configuration = Outward

A.2.5 Gull-wing Terminal (cont'd)

A Neck is very similar to a Shoulder except that the width of the neck is narrower than the width of the terminal contact area.

Typically, these Dimensions are “Unspecified Value Set type”, meaning that if the datasheet does not provide these values, we will set them to defaults.



The final gull-wing modification is a Dambar as shown here. The Dambar protrusion is a remnant from tie bar cutting.

The terminal width dimension “b” does not include dambar protrusion, but most manufacturers will specify the allowable dambar protrusion as not causing the terminal width to exceed the maximum terminal width dimension by a specified amount.

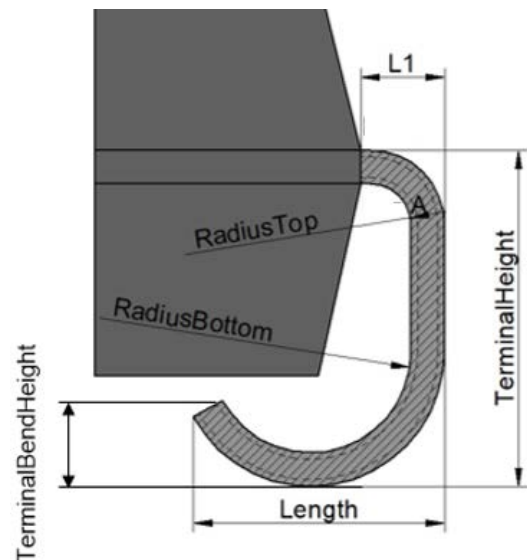


A.2.6 J-Bend

J-bend compliant terminal extend from the sides of the body, bend down, and form a curved contact area with the board under the package body. It is similar to a C-bend terminal except that the bottom of the J-terminal is curved and rounded up under the part body.

When terminals are in a Dual position on the device, then the dimension *Pitch* in the direction across the package body for the terminals in the dual position is normally provided, in addition to the terminal *Length*.

The elements *PackageEdge-to-end-of-Terminal*, *Width*, *TerminalHeight*, *TerminalThickness*, *RadiusTop*, *RadiusBottom*, and *RiseAngle*, follow the same definitions as that provided for the C-bend. *TerminalBendHeight* is useful for the purpose of generating more accurate 3D models and is therefore optional.



The applicable shoulder elements for the *ShoulderTransition/Tapered* shoulder are as follows:

- 1) *SeatingPlane-to-start-of-ShoulderTaperTransition*, and *SeatingPlane-to-start-of-ShoulderTaperTransition*,

A.2.6 J-Bend (cont'd)

or

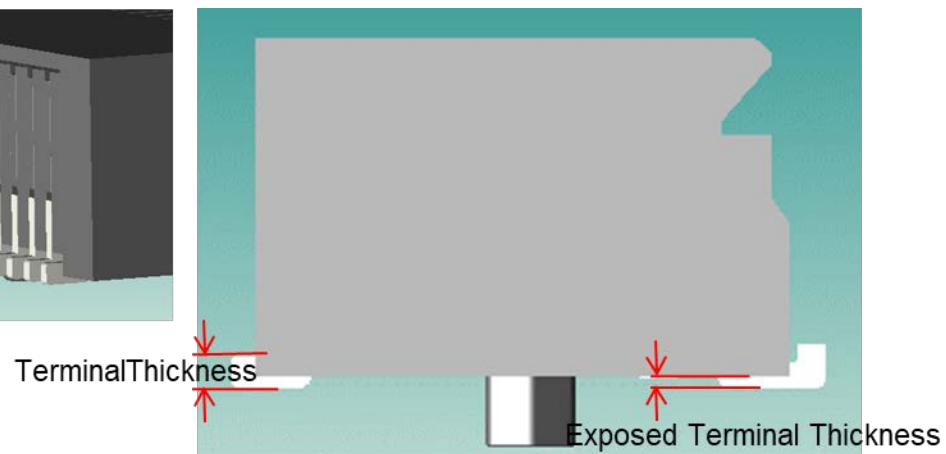
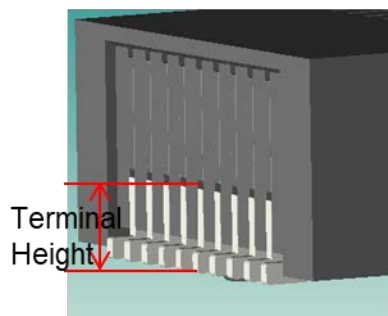
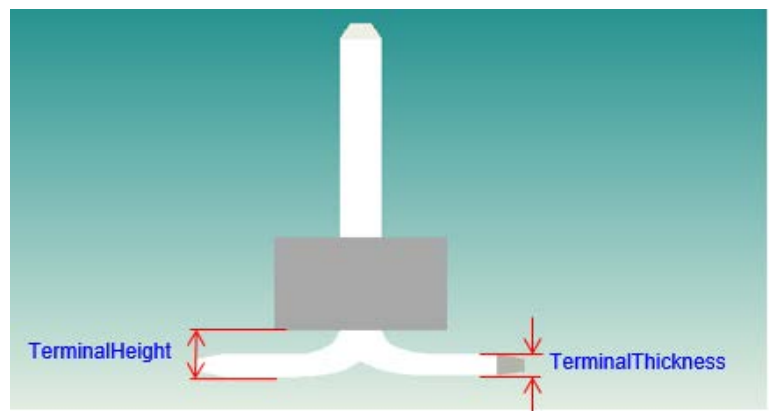
- 2) *Top-of-TerminalExit-from-Package-to-start-of-ShoulderTaperTransition*, and *Top-of-TerminalExit-from-Package-to-start-of-ShoulderTaperTransition*.

The applicable shoulder elements for the *ShoulderTransition/RightAngled* shoulder are the

- 1) *SeatingPlane-to-ShoulderTransition*, or *Top-of-TerminalExit-from-Package-to-ShoulderTransition*.

A.2.7 L-bend

The *L-bend* terminals exits vertically from the bottom of the package and then turns horizontal. Irrespective of whether, the terminal variant is *Inward*, *Outward*, *SideInward*, *SideOutward* or it is a normal *L-bend* in which the position is one of *Single*, *Dual*, *Triple* or *Quad*, the *TerminalThickness* drives the Toe calculation of the land pattern, while the *TerminalHeight* drives the heel portion of the calculation.

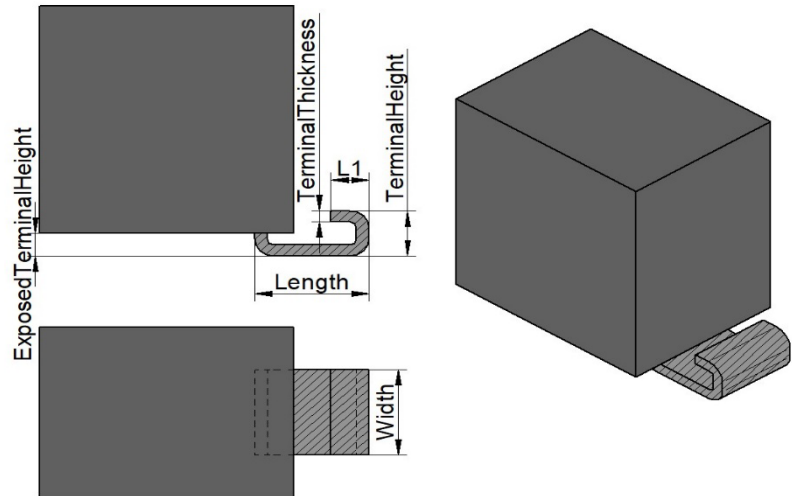


Sometimes as shown here in the 2nd image, the package body sits on the top surface of the horizontal portion of the L-Bend, in which case the *TerminalHeight* is reduced to the *TerminalThickness*.

A.2.7 L-Bend (cont'd)

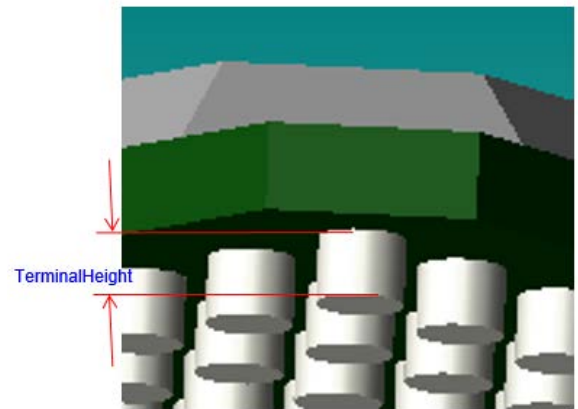
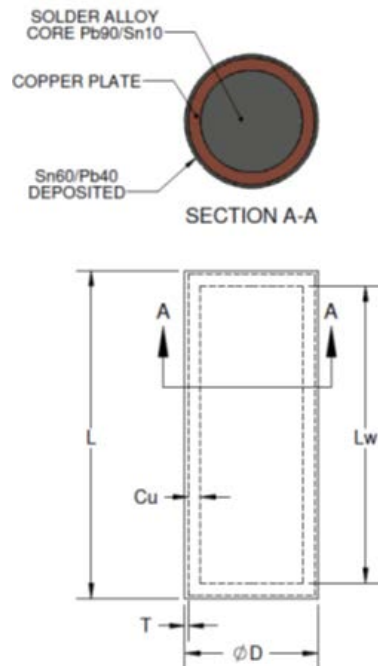
For Side-L-Bends, the terminal is partially immersed into the package body, in which case the *ExposedTerminalThickness* may be reduced to less than the *TerminalThickness*. In this case, the *ExposedTerminalThickness* is mandatory, while the *TerminalThickness* is optional.

LC-bends follows the same rule as the Side-L-Bend in which the *ExposedTerminalThickness* under the package body is the critical measurement.

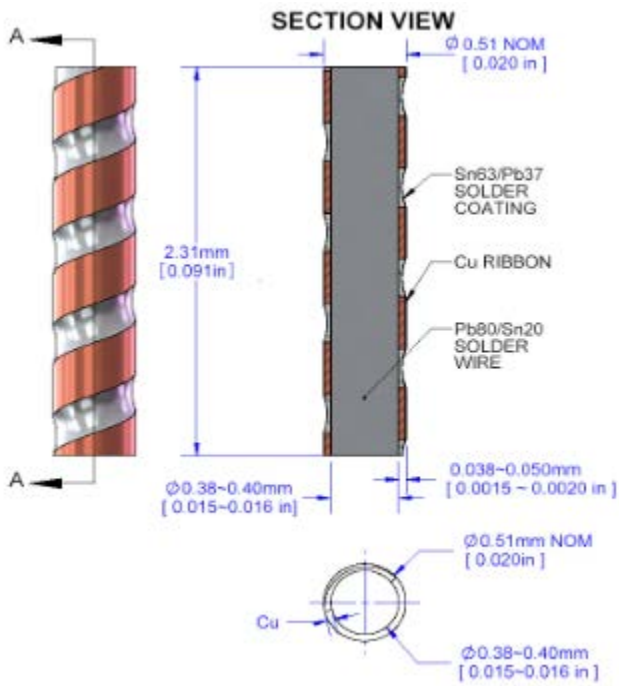


A.2.8 Column

The Column *TerminalHeight* is the same as the Standoff for the Part, provided that there are no bumps on the package body.



A.2.8 Column (cont'd)

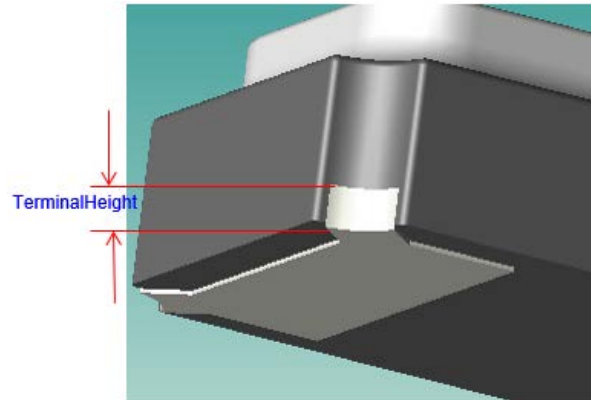


A.2.9 Surface Terminal

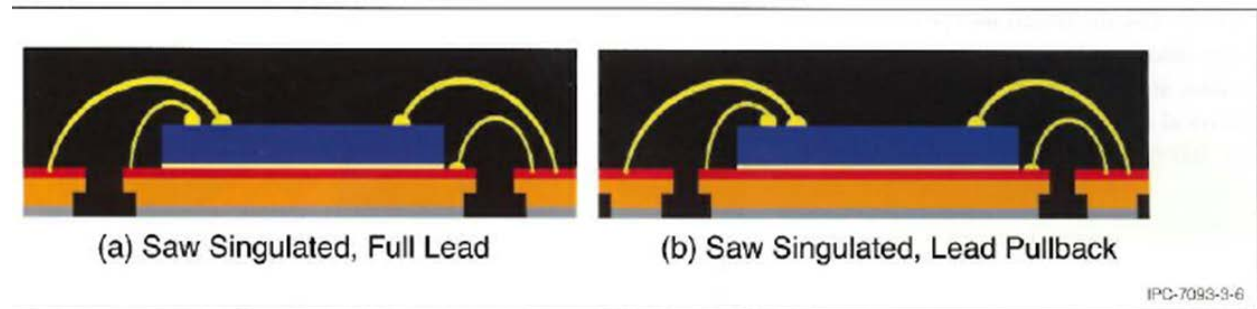
The majority of *SurfaceTerminals* are under the bottom of the Package where the Terminal Contact Area is Inside the package outline. Such Terminals have no Terminal Thickness or Terminal Height as these 3 images indicate.

However there are variants where the edge of the *SurfaceTerminal* is touching the edge of the Body and sometime wrap up the side of the body. In such cases, *TerminalHeight* shall be captured.

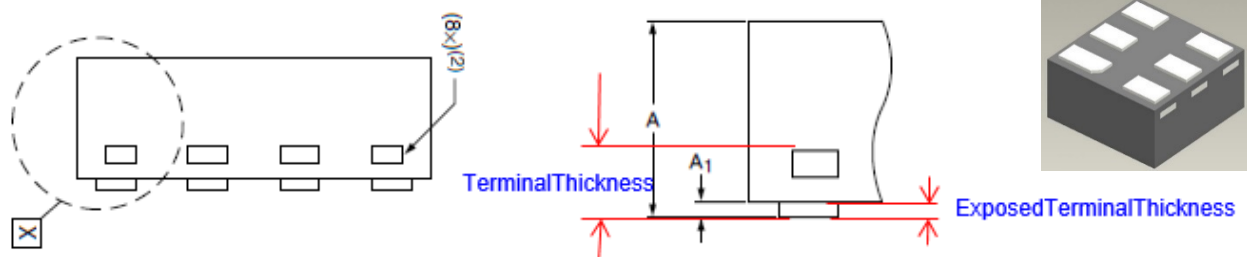
There are also some variants in which the side wrap is the same height as the package body. This is almost considered a *Wraparound* terminal, but if the terminal does not wrap around the top surface, then it is still a *SurfaceTerminal*. In these cases, the *TerminalHeight* is equal to the Body Height.



SurfaceTerminal variants that are *Pullback*, have an exposed thickness of the terminal under the bottom surface of the package body.



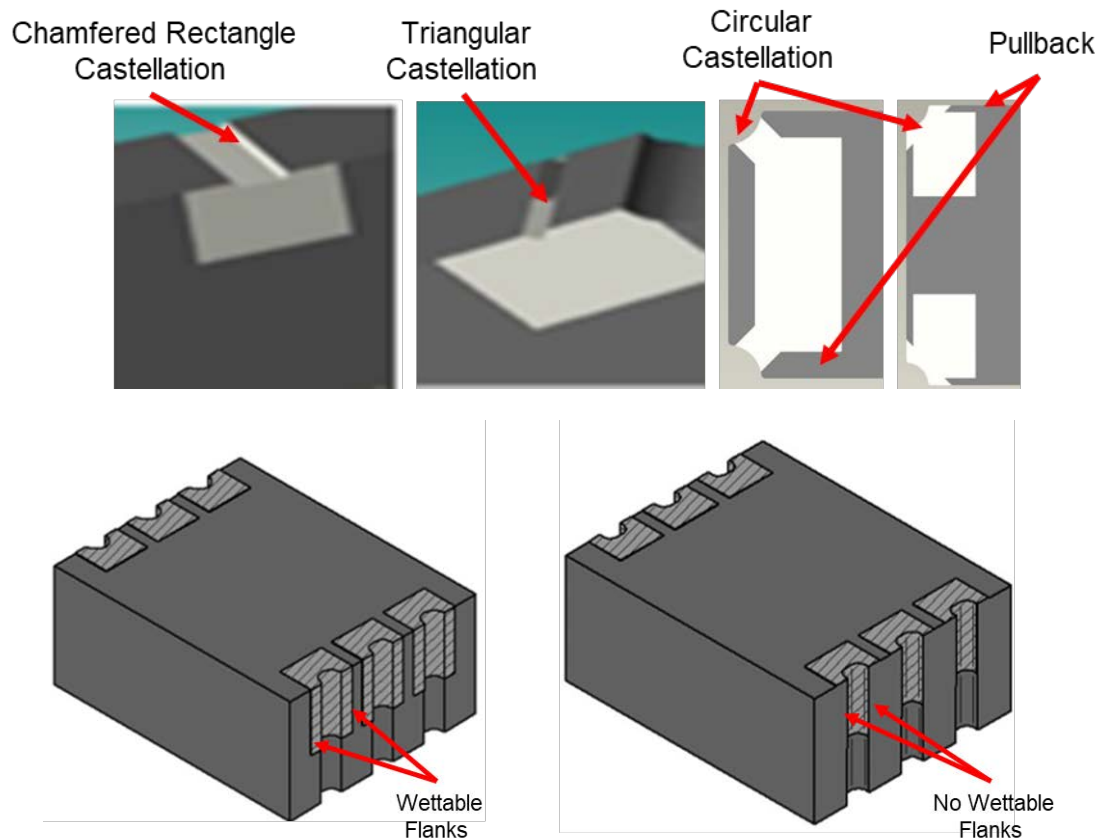
Only the *ExposedTerminalThickness* is required here. There is no value in collecting the *TerminalThickness* as defined below.



There are several different kinds of castellations such as

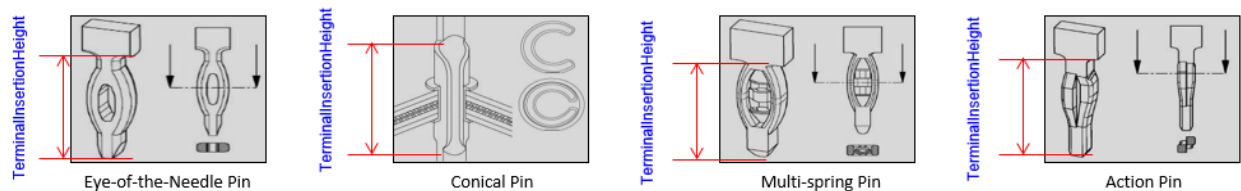
- 1) Circular Castellation
- 2) Triangular Castellation,
- 3) Chamfered Rectangle Castellation, and
- 4) Rounded Rectangle Castellation.

A.2.9 Surface Terminal (cont'd)



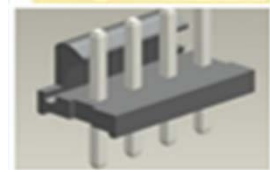
A.2.10 Pressfit

There are 4 types of *Pressfit/Compliant* terminals plus a Non-Compliant terminal. In all cases, the dimension to be captured is the length of the terminal from the end point to the point of maximum insertion of the terminal through the hole. This stopping point can be a shoulder on the pin, or it can be the package body. This is defined as the *TerminalInsertionHeight*.

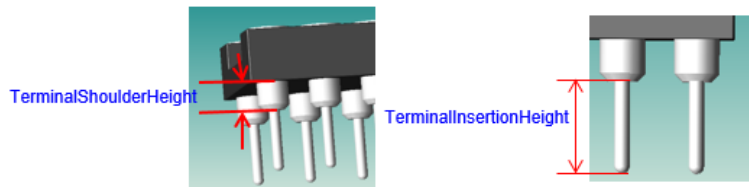


A.2.11 Pin

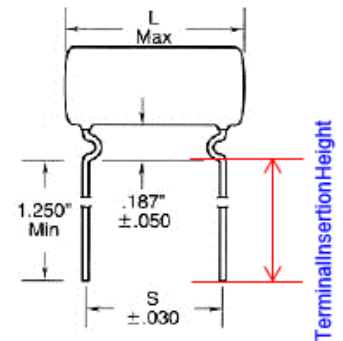
A *Pin* can be straight, *Kinked* or come with a *Shoulder*. There is also 3 additional variants, notably a *Press-inSolderable*, *Press-inNon-Solderable*, or a *SwageFasteningPin*.



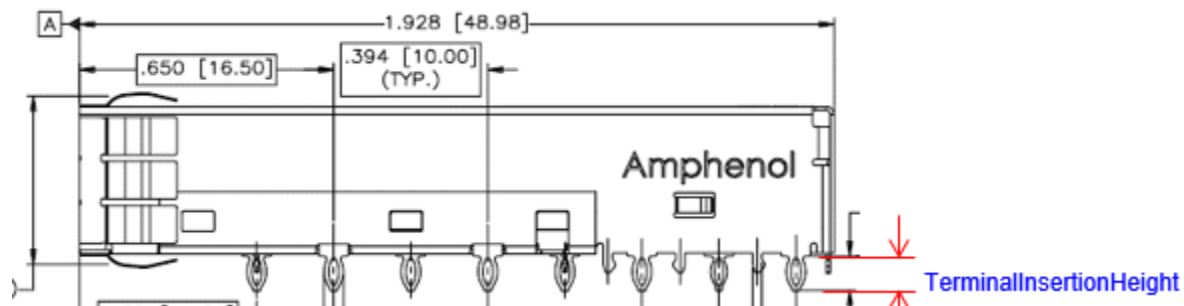
A *Pin* with a *Shoulder* will have a *TerminalInsertionHeight* that goes through the printed board. Plus a *TerminalShoulderHeight* that is typically the same as the Stand-off height.



The Pin with the variant *Kinked* acts in the same way as the Pin with the *Shoulder*. The dimension of the terminal that can penetrate the printed board (i.e. the section of the Pin below the Kink) is considered the *TerminalInsertionHeight*. The distance from the bottom of the “kink” to the underside of the package body is the considered the standoff.



For *Press-inSolderable* terminals, the *TerminalInsertionHeight* is considered the section of the terminal below the shoulder or below the Package body in the absence of a shoulder.

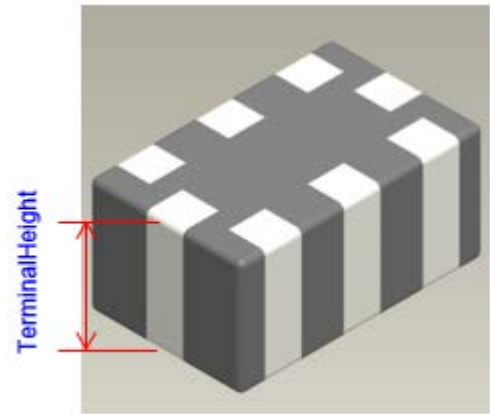


The same concept applies to the *Press-inNon-Solderable* terminals. The *TerminalInsertionHeight* is the length of the terminal that could penetrate the printed board.

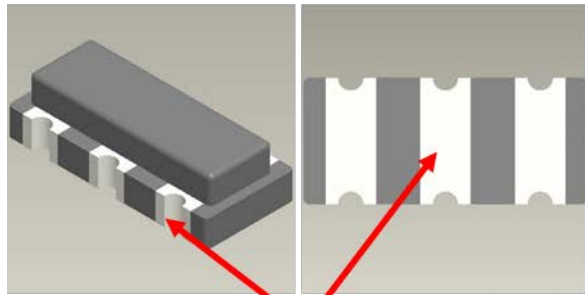
A.2.12 Wraparound

Wraparound terminals are like *SurfaceTerminals* in that the surface of the terminal is on the same plane as the surface of the package body, hence the term surface terminal. *TerminalThickness* may be “0” but the *TerminalHeight* is the height of the terminal when looking at the side elevation.

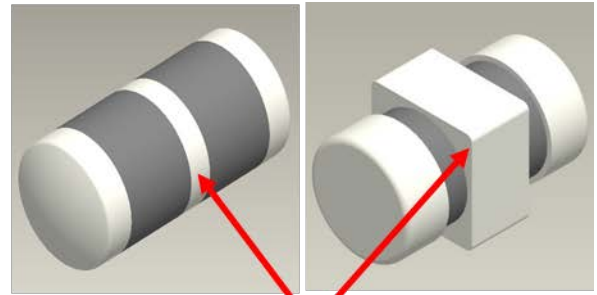
Some Parts can have a wraparound terminal that does not cover the same height as the Package Body. In this case the Terminal Height is still captured and is mandatory.



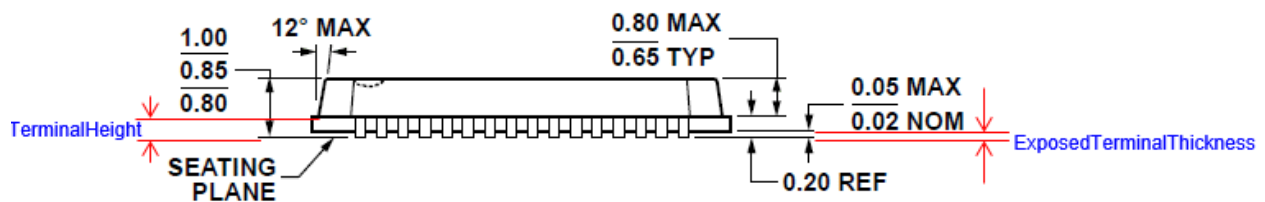
There are various options for Wraparound terminals. The castellation structure for *Wraparound* terminals is the same as for *Surface-terminals*.



Open Ring

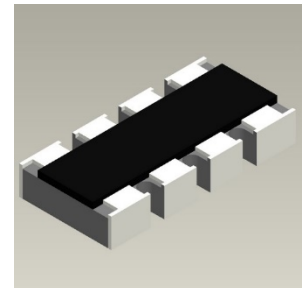


Ring



It is also possible that the surface of the wraparound terminal is not flush with the surface of the package body. This becomes particularly important on the underside of the part that connects with the land pattern. Since these terminals are embedded into the Package body material, only a portion of the terminal thickness is exposed. This *ExposedTerminalThickness* is a mandatory dimension to be collected.

Convex-S has the end Terminals larger than the terminal in between the ends.



A.2.13 S-Bend

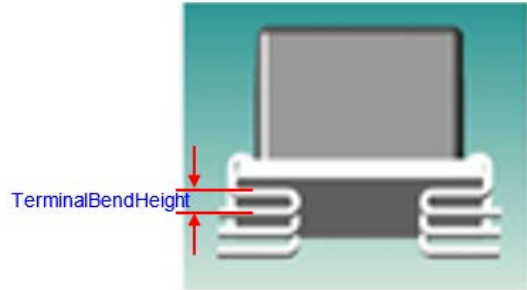
S-bend has a terminal contact area similar to a *C-bend*, or a *J-bend*.

The critical dimension for footprints is the *TerminalThickness* as shown here.

Similar to the J-bend, other dimensions can also be captured to facilitate more accurate 3D representation.

The *TerminalHeight* is useful only for the purpose of generating more accurate 3D models, and is therefore optional to be captured.

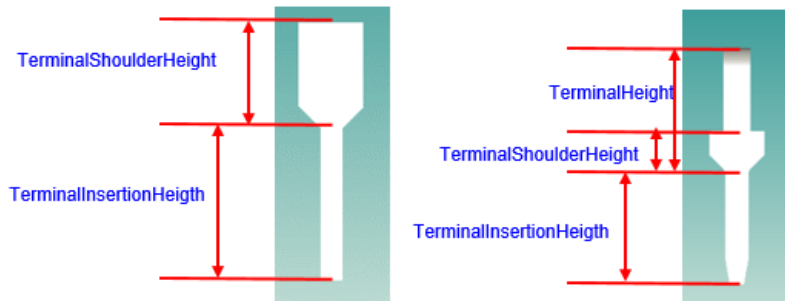
The *TerminalBendHeight* is also only useful only for the purpose of generating more accurate 3D models. It is also assumed that for S-bends, both the *TerminalHeight* and the *TerminalBendHeight* are well in excess of the max height that the solder will creep up the terminals, from which the projection to the land pattern for the toe or heel is defined. Therefore the *TerminalHeight* and the *TerminalBendHeight* are optional data to be captured.



A.2.14 Through-Hole

Through-hole terminals can exit from the side of the package or from the underside of the Package body.

As can be seen from the different types, the terminal shoulder can travel up all the way to reach the Package body. The *TerminalShoulderHeight* is the distance from the bottom of the Shoulder to the top of the terminal shoulder, irrespective of whether or not the Terminal exits from the side of the body or from the underside of the package body.

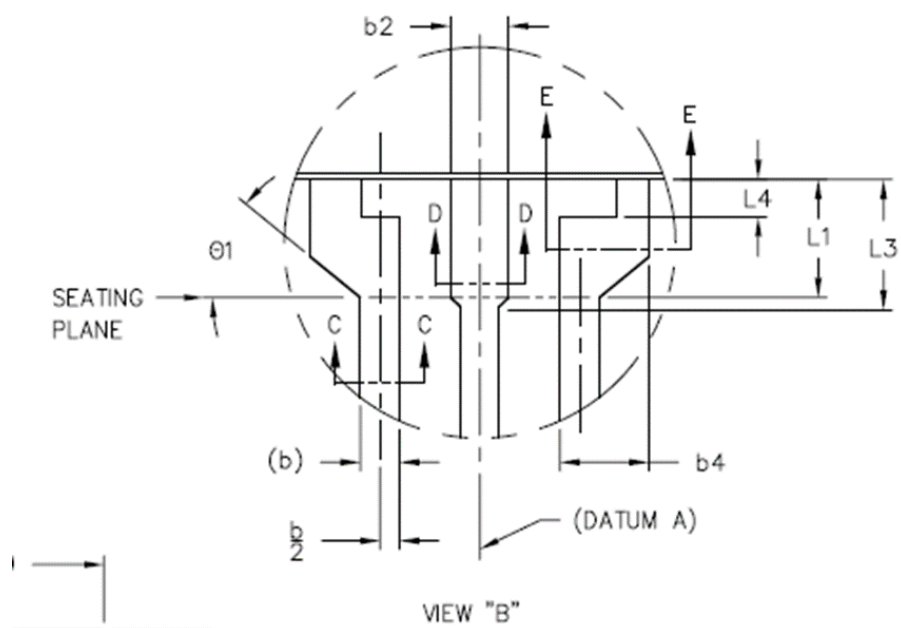
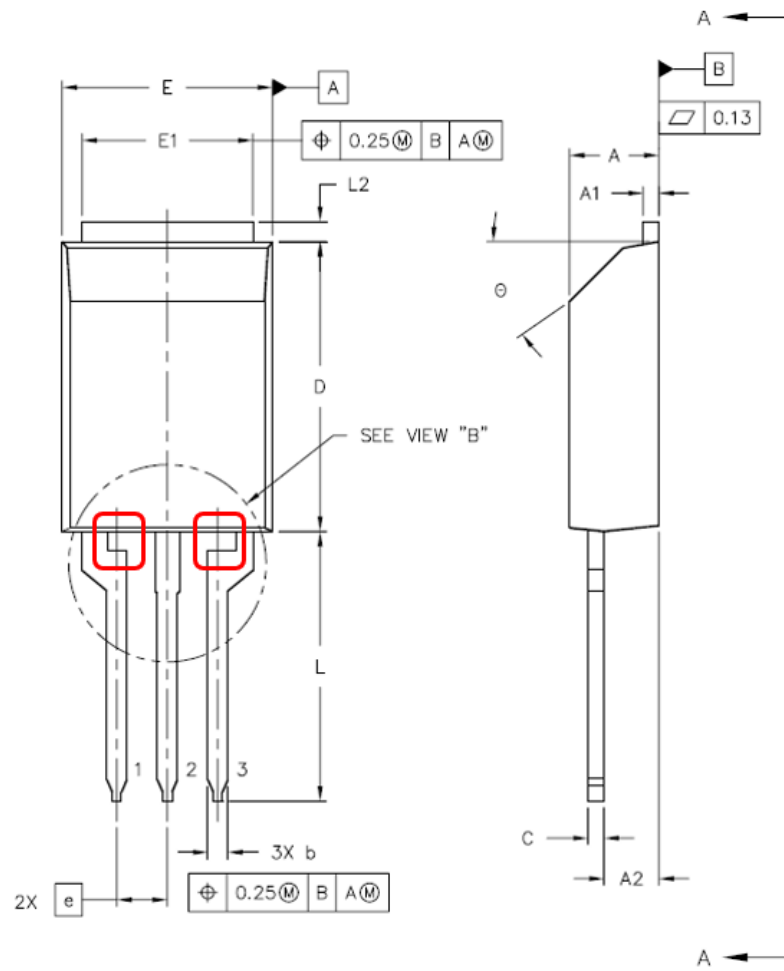


However in 2nd image, the shoulder does not reach the Package body. In this case the dimension from the bottom of the Shoulder to the top of the Terminal is called the *TerminalHeight*, whereas the distance from the bottom of the Shoulder to the top of the Shoulder remains the *TerminalShoulderHeight*.

In all types, the *TerminalInsertionHeight* is the distance from the bottom of the Shoulder to the end of the Terminal. This dimension is critical.

Through-Hole terminals such as those on Flange Mount devices can have cutouts at the top of the terminals. The details of these cutouts can be captured via the *TH-ShoulderCutout* section of the schema. An example of cutouts is shown below.

A.3.4 Through Hole (cont'd)

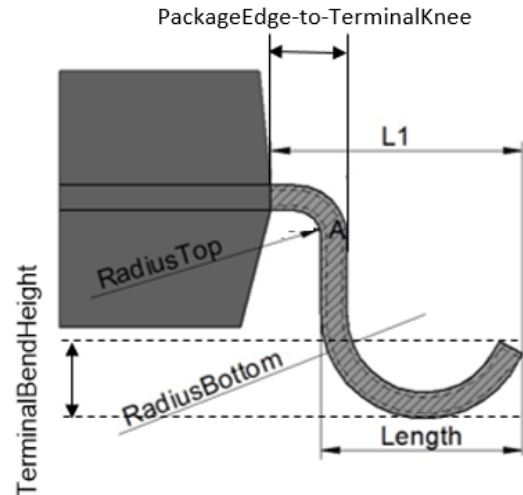


A.2.15 J-Inverted

J-inverted compliant terminal extends from the sides of the body, bend down, and form a curved contact area with the board outside the package body. It is similar to a J-bend terminal except that the bottom of the J-terminal is outside the package as opposed to under the package.

When terminals are in a Dual position on the device, then the dimension *Pitch* in the direction across the package body for the terminals in the dual position is normally provided, in addition to the terminal *Length*.

The element the *PackageEdge-to-TerminalKnee* measures the distance from the edge of the package to the intersection of the projection of the vertical portion of the terminal as defined by the rise angle. If the rise angle is 90 degrees, then this *PackageEdge-to-TerminalKnee* dimension equals the span $Pitch - Length + TerminalThickness \cdot 2$



The elements *Width*, *TerminalHeight*, *TerminalThickness*, *RadiusTop*, *RadiusBottom*, and *RiseAngle*, follow the same definitions as that provided for the C-bend. *TerminalBendHeight* is useful for the purpose of generating more accurate 3D models and is therefore optional.

The applicable shoulder elements for the *ShoulderTransition/Tapered* shoulder are as follows:

- 1) *SeatingPlane-to-start-of-ShoulderTaperTransition*, and *SeatingPlane-to-start-of-ShoulderTaperTransition*,
- or
- 2) *Top-of-TerminalExit-from-Package-to-start-of-ShoulderTaperTransition*, and *Top-of-TerminalExit-from-Package-to-start-of-ShoulderTaperTransition*.

The applicable shoulder elements for the *ShoulderTransition/RightAngled* shoulder are the

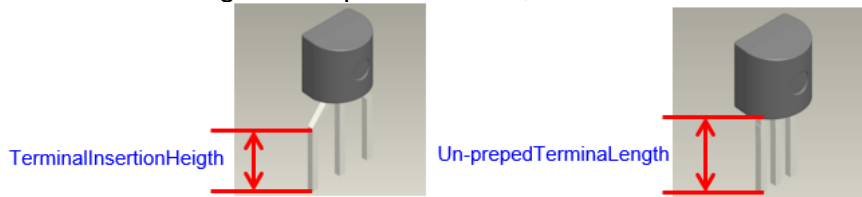
- 2) *SeatingPlane-to-ShoulderTransition*, or *Top-of-TerminalExit-from-Package-to-ShoulderTransition*.

A.2.16 Wire

There are several types of wires. For normal Wires, the **Un-preppedTerminalLength** is defined as the distance from the end of the Terminal to the point where it exits from the Package body.

This applies to all Wire Terminals that are not pre-formed.

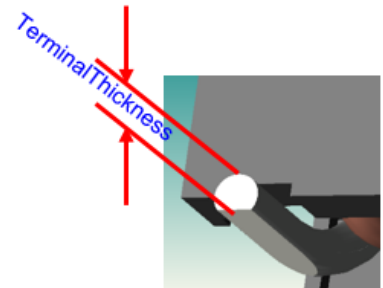
Where the Product Assembler preforms the Part to form either a **Horizontal** or **Vertical PackageBodyDirection** after mounting to the printed board, then the dimensions captured should reflect the prepped part. After prepping, the distance from the end of the Terminal to the Kink in the Terminal is defined as the **TerminalInsertionHeight**.



Un-preppedTerminalLength dimension is required to enable the various Terminal Insertion Spans to be calculated.



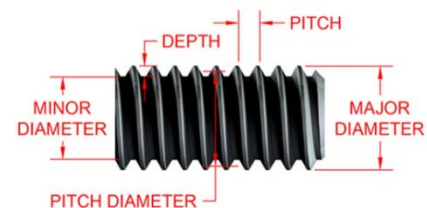
A **Wire** can also be surface mounted to the printed board, as in this image. Most **SM-Wires** are flattened so that the terminal does not roll off the pad, and this results in a lower vertical **TerminalThickness** dimension than the terminal diameter.



A.2.17 Screw

Major diameter (outside diameter) is the largest material diameter of the thread feature. Outside or external thread diameter means the diameter of a screw and includes a raised helix height around the thread.

Minor diameter can be defined as an imaginary cylinder that can touch the roots of the thread crests.



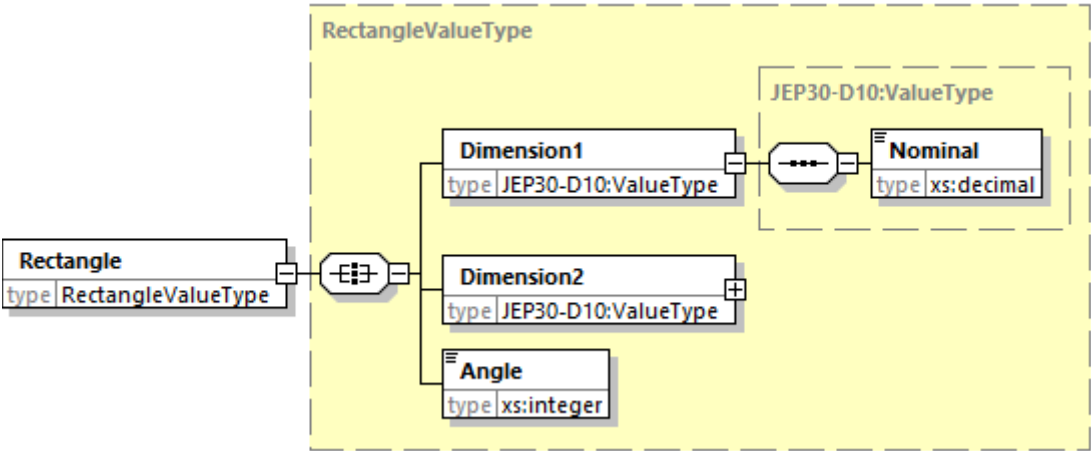
Effective or pitch diameter is the diameter of an imaginary co-axial cylinder that can intercept the surface of the thread. Pitch diameter can cross so that the intercept on a cylinder generator at the point where pitch diameter meets the opposite flanks of the thread groove. The thread groove is equal to half of the nominal pitch of the screw thread.

The Thread Pitch is a distance from a point on the same screw thread to a corresponding threshold.

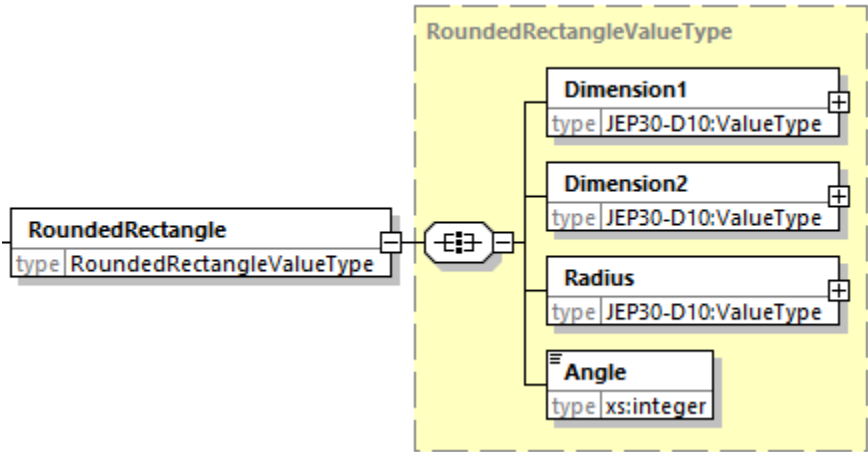
A.3 Land Pattern Shapes

The following sections outlines the recommended land pattern shapes. All shapes defined for design tools will only contain the Nominal value, since this is the target value that is defined in the software tools for which the design is intended.

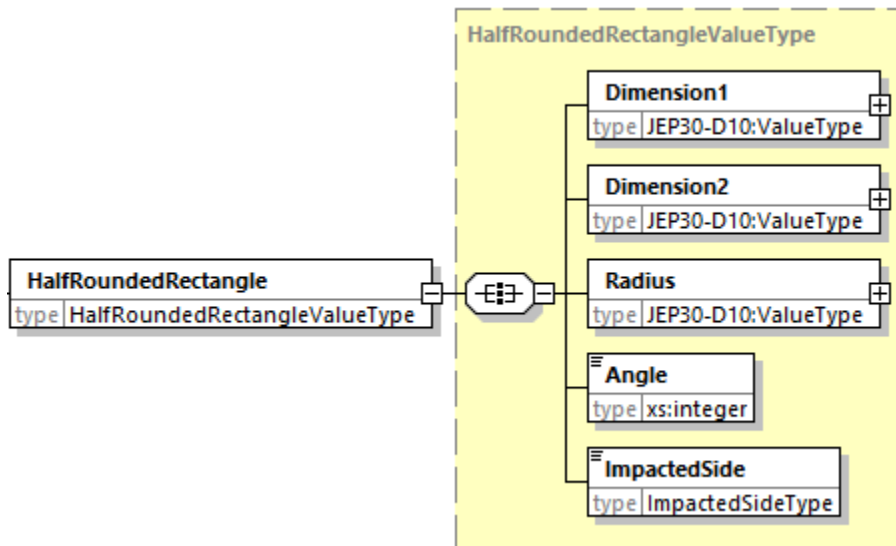
A.3.1 Rectangle



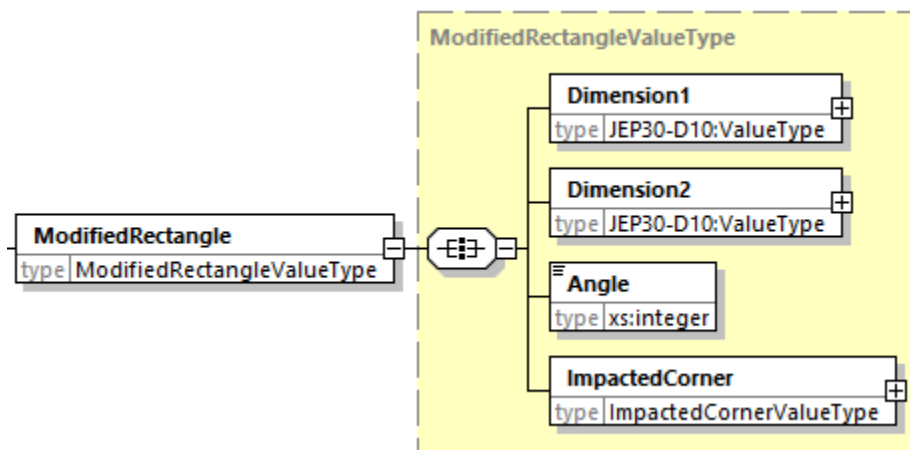
A.3.2 Rounded Rectangle



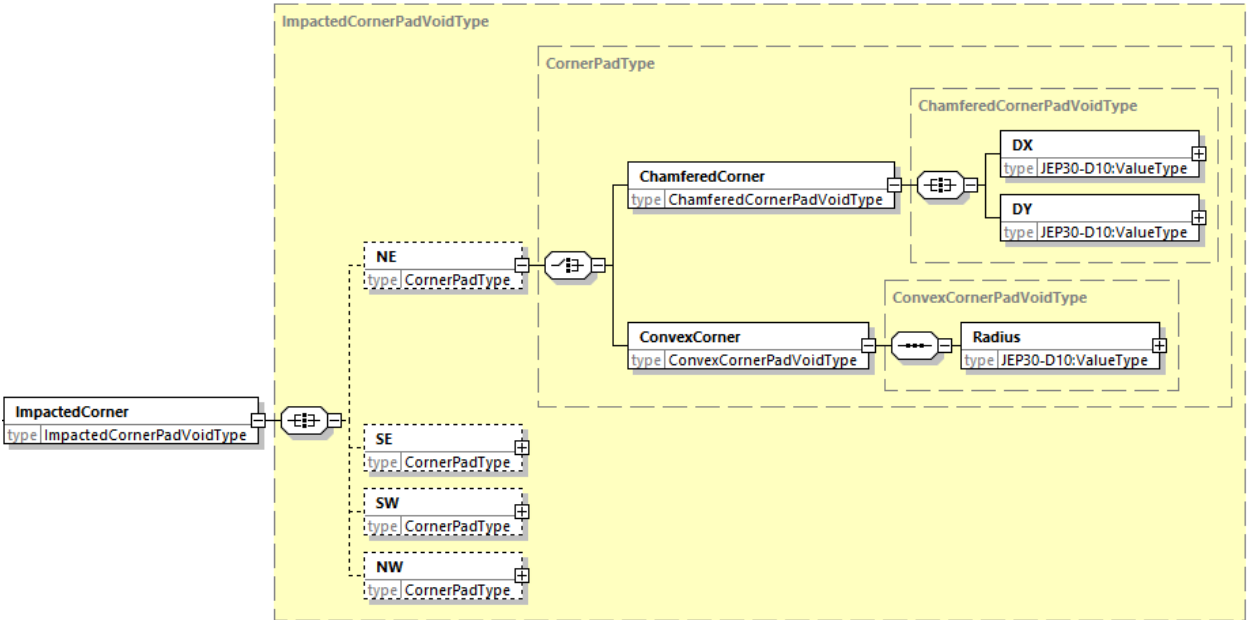
A.3.3 Half Rounded Rectangle



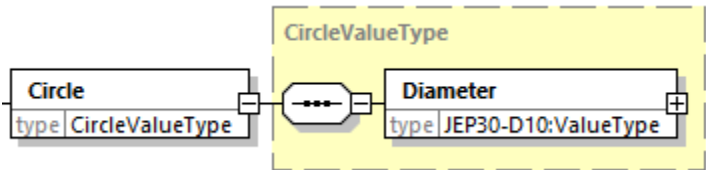
A.3.4 Modified Rectangle



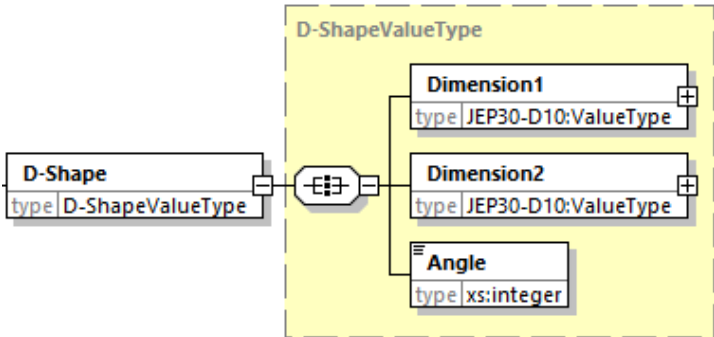
A.3.4 Modified Rectangle (cont'd)



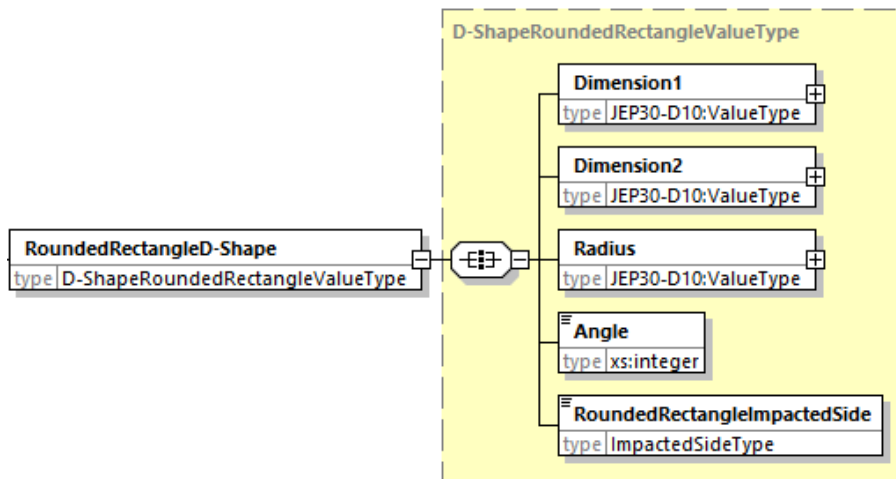
A.3.5 Circle



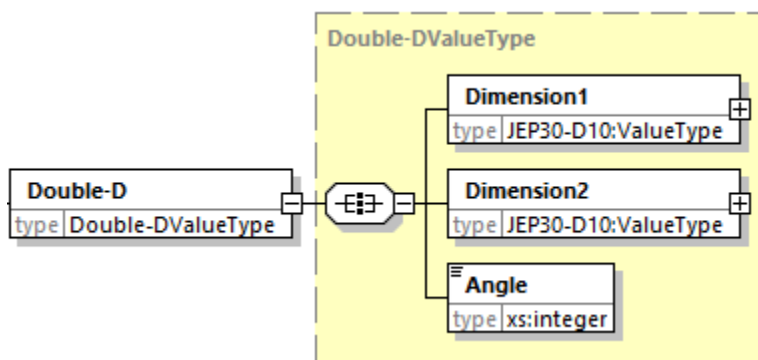
A.3.6 D-Shape



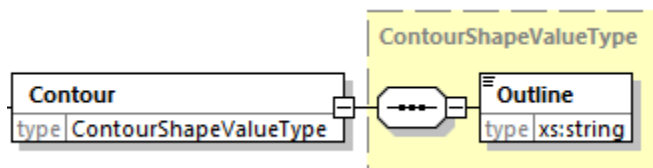
A.3.7 Rounded Rectangle D-Shape



A.3.8 Double-D

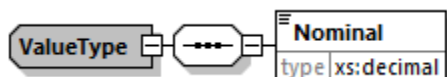


A.3.9 Contour



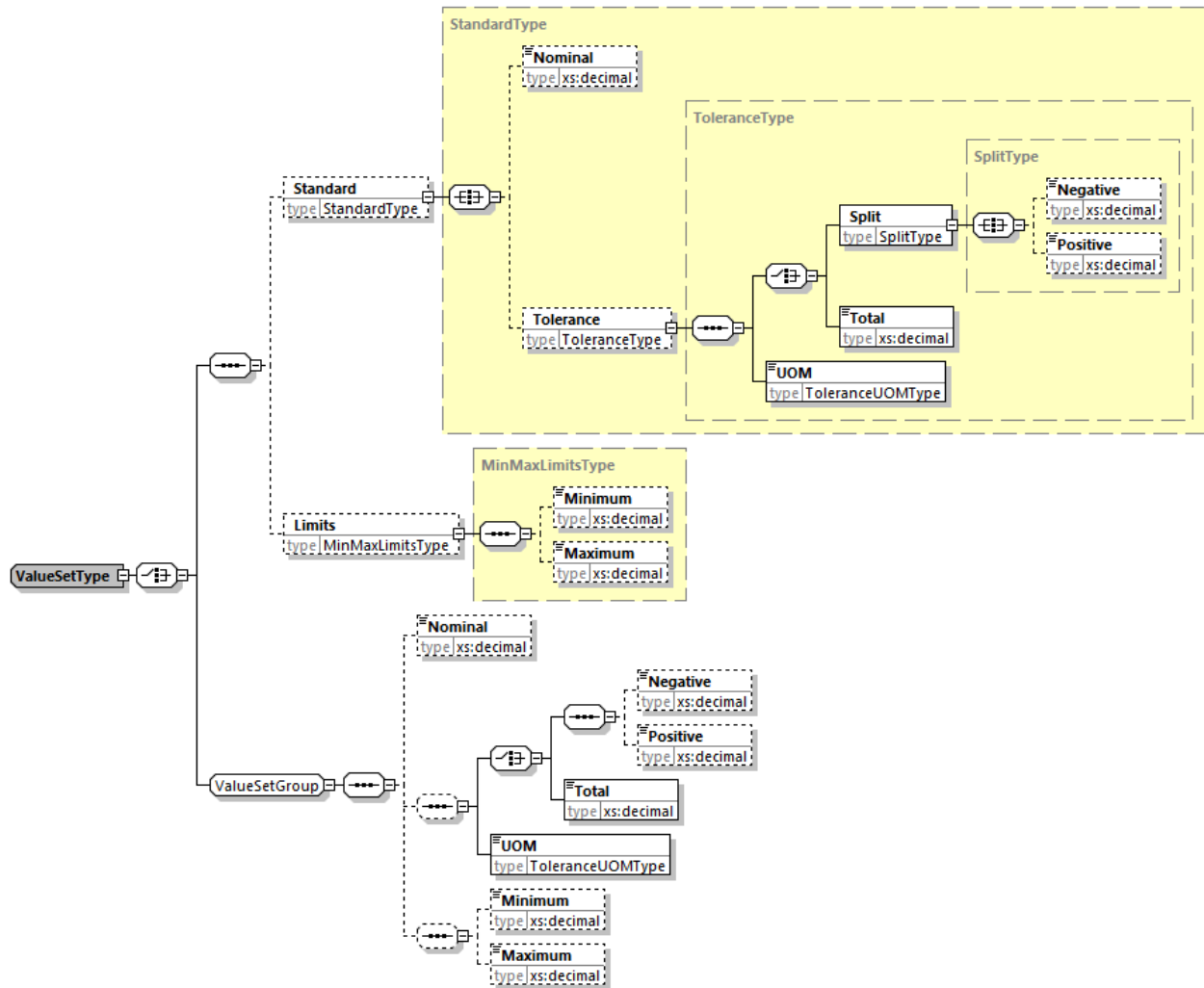
A.4 Value Set Types

A.4.1 Value Type



Recommended land pattern shapes, only requires the nominal to be defined and not any of the tolerances nor the min and max values

A.4.2 Value Set Type

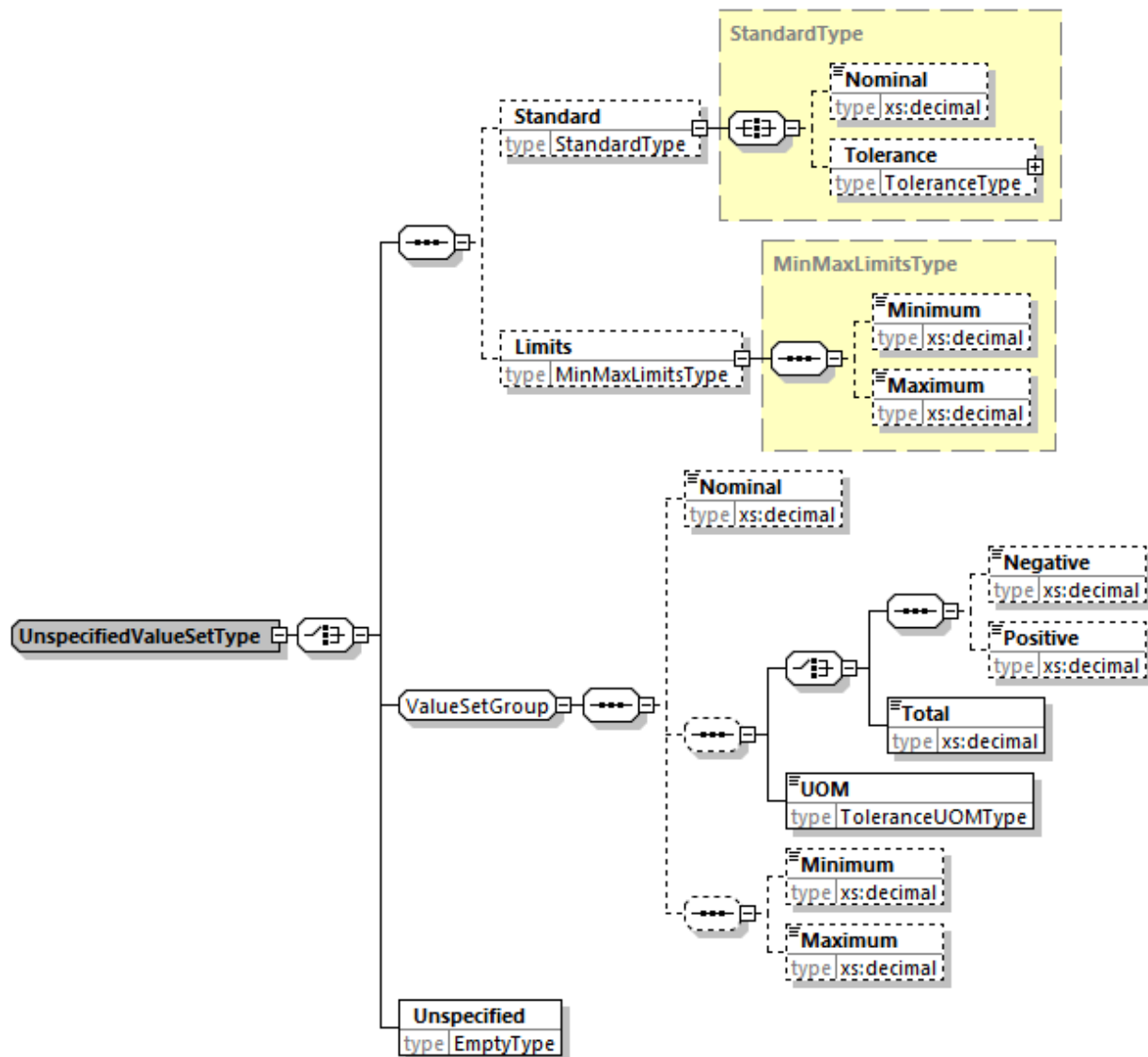


The following set of data can be captured for each dimension via the *ValueSetType*:

- *Nominal*,
- *Negative* and *Positive* Tolerance, or *Total* Tolerance. The *UOM* (*ToleranceUOMType*), defines whether the tolerance value is a percentage of the Nominal or in absolute values, according to the *DimensionUOM* selected.
- *Minimum*,
- *Maximum*.

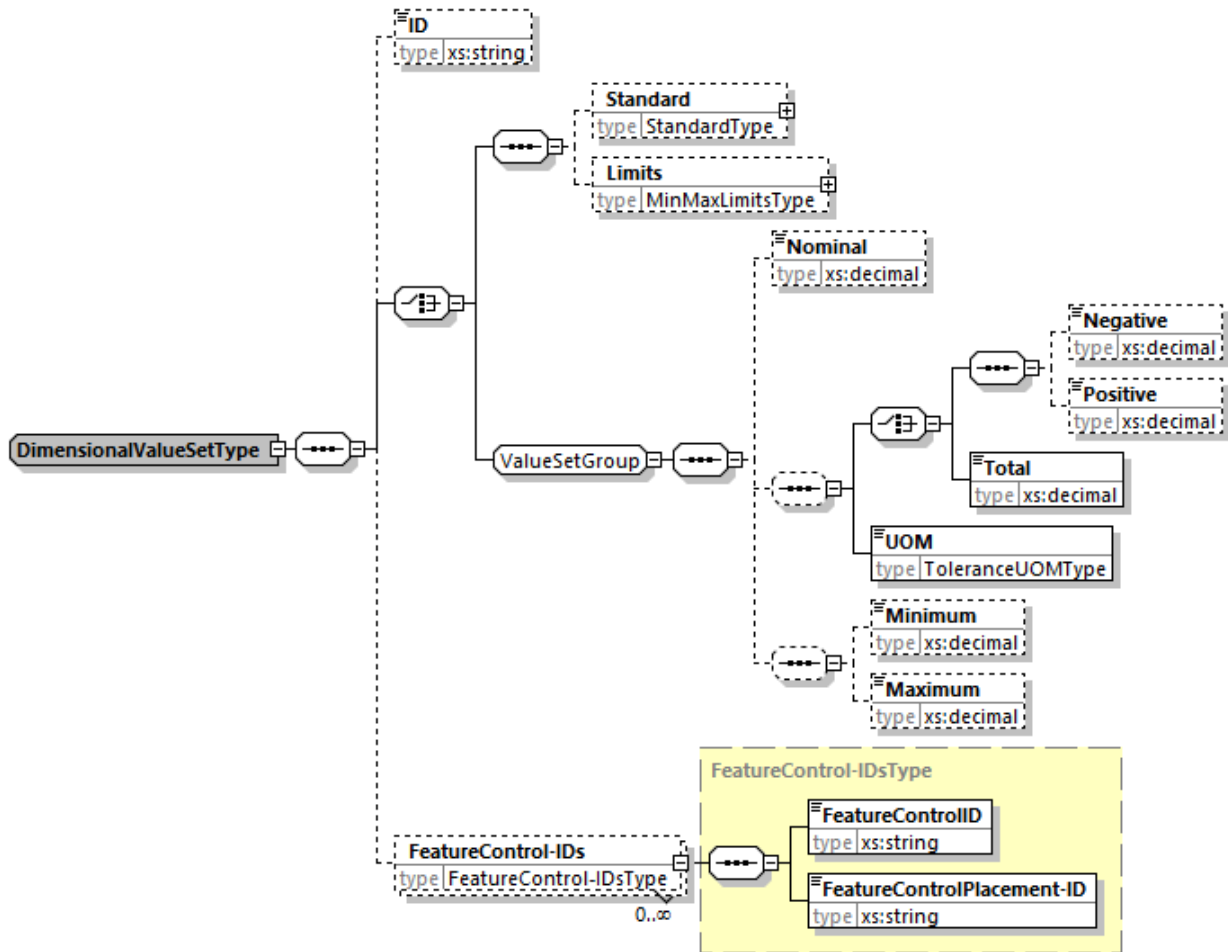
It is highly recommended to utilize the *ValueSetGroup* as opposed to the upper *Standard* and *Limit* structure, as this reduces the size of the xml file and flattens the hierarchy within the xml file. The existing *Standard* and *Limit* structures are maintained within the schema for backward compatibility.

A.4.3 Unspecified Value Set Type



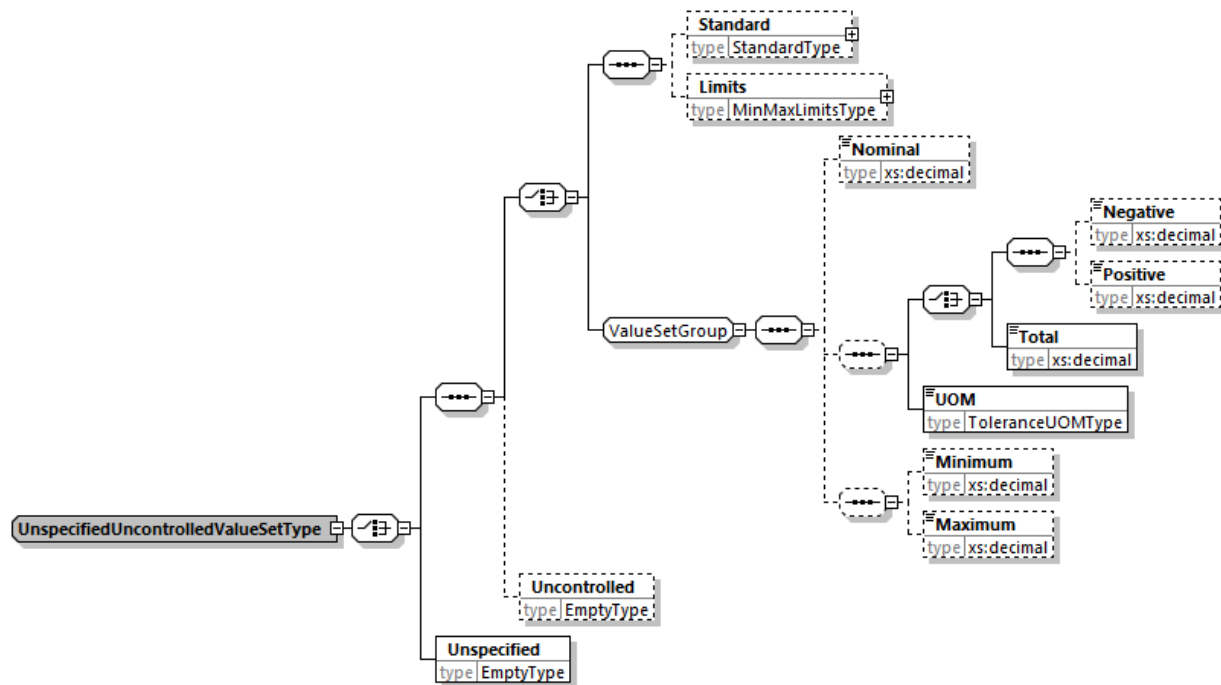
This is the same as the *ValueSetType* with the exception that *Unspecified* can be chosen in the event that that particular dimension is not specified. This is only applied to non-critical dimensions.

A.4.4 Dimensional Value Set Type



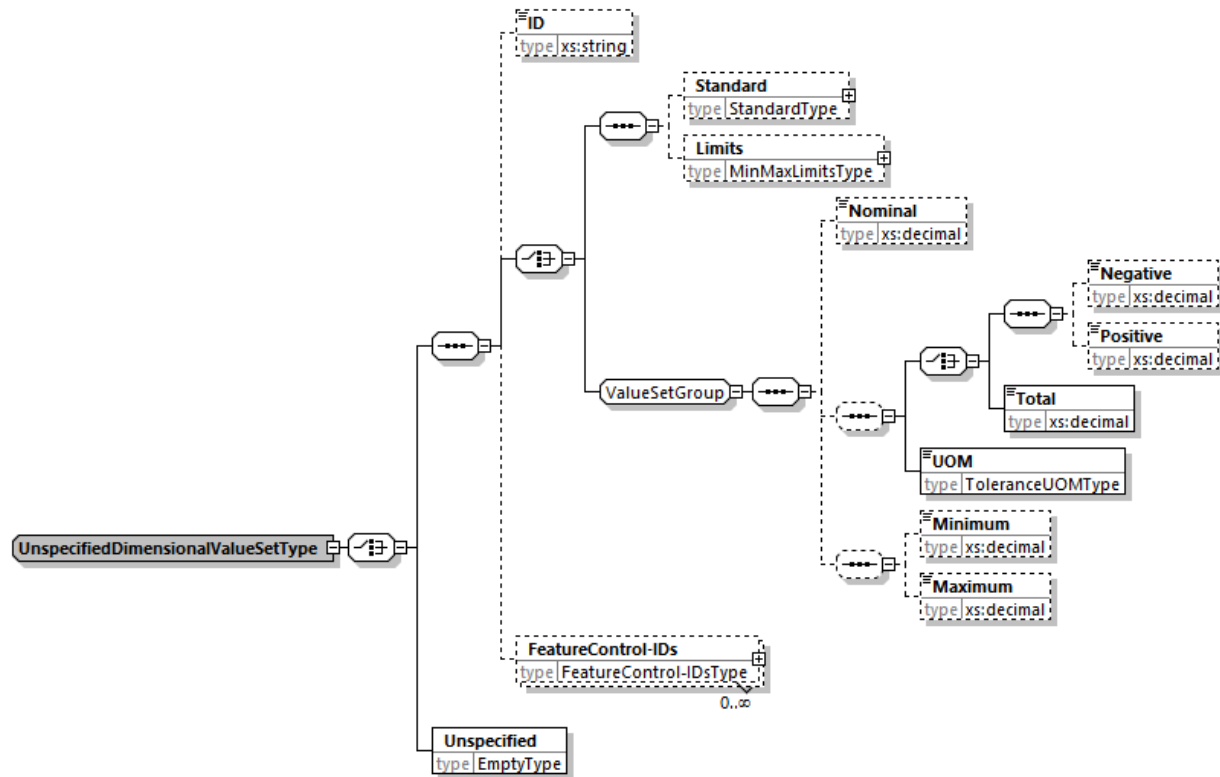
This is the same as the [ValueSetType](#) with the exception that [FeatureControl-IDs](#) is an added structure to support GD&T.

A.4.5 Unspecified and Uncontrolled Value Set Type



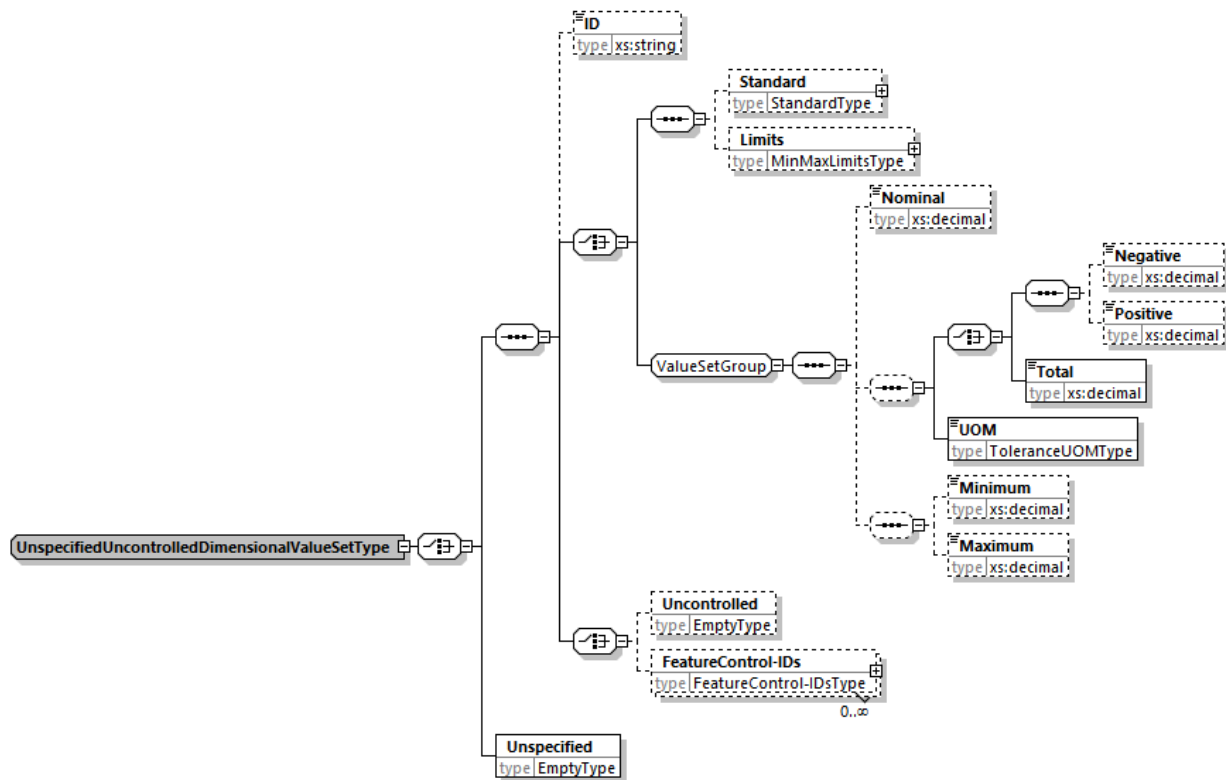
This is the same as the *ValueSetType* with the exception that even when dimensions are provided, that these dimensions are un-controlled. Often, these dimensions are omitted from the drawing, hence *Unspecified* can be chosen in the event that that particular dimension is not specified. This is only applied to non-critical dimensions.

A.4.6 Unspecified Dimensional Value Set Type



This is the same as the [DimensionalValueSetType](#) with the exception that *Unspecified* can be chosen in the event that that particular dimension is not specified. This often occurs on terminal shoulders where some situations these critical shoulders definitions require GD&T references, and in other in other situations, the shoulder is not critical and can be left unspecified.

A.4.7 Unspecified and Uncontrolled Value Set Type



This is the same as the [UnspecifiedDimensionalValueSetType](#) with the exception of the addition of the *Uncontrolled* element as a choice to the *FeatureControl-IDs*. Often, these dimensions are omitted from the drawing, hence *Unspecified* can be chosen in the event that that particular dimension is not specified. This is only applied to non-critical dimensions.

Annex B (informative) Differences between JEP30-P100 and its predecessors

This table briefly describes most of the changes made to entries that appear in this standard, JEP30-P100, compared to its predecessor; Punctuation changes may or may not be included.

| | | |
|----------------|----------------|-----------------------|
| Initial Issue: | Date: May 2018 | Item Number: 11.2-938 |
|----------------|----------------|-----------------------|

Change Record History

| | | |
|---|------------------|------------------------|
| Issue: A | Date: March 2023 | Item Number: 11.2-839S |
| Section 2.1 Applicable Documents JEDEC – Added in reference to new document SPP-010 | | |
| Section 2.1 Applicable Documents ASME – Added in reference to new document ASME Y14.5-2009 | | |
| Section 4.4 Package-Array – Updated Package Type to include <i>GDAndT</i> section. | | |
| Updated all diagrams in all sections that had elements with no type to now have an JEP30-D10:EmptyType. | | |
| Section 4.5 Package Terminal Position – Added Internal position to align with JESD30J Also changed MultiPosition to MixedPosition. | | |
| Section 4.6.1 Microelectronics Assembly: Added new extended package outline option for “Chiplets” | | |
| Section 4.6.2 Post Mount: Changed “Sequence” to “Choice” | | |
| Removed Section 4.6.3 <i>Switch</i> , since the extended package outline section for Switch is moved to Part Classification under JEP30-E101 | | |
| Section 4.7.13 Through-Hole: Added missing diagram for Through-Hole | | |
| Section 4.14 Keepout Region – Make Component Optional | | |
| Section 4.19 Terminal Function – Added Thermal as an additional function. | | |
| Sections 4.20.2, 4.21.2.2 and 4.22.3.2 Circular Array – Updated <i>RotateWithCircle</i> and <i>RotationAngle</i> types | | |
| Section 4.22.3.3 Pad-or-Hole Status is updated to enable the center and Index to be unbounded. | | |
| Section 4.25 Updated Table 8 - Terminal Group to Package Body Relationship Dimension Variations to replace “Lower” and “Upper” with “Back” and “Front in Y1-Y27 Offsets. Also added Offsets for the Z-direction | | |
| Section 4.26 Added in the <i>Z-Direction</i> to <i>TerminalSpan</i> and <i>TerminalSpacing</i> into <i>TerminalGroupToTerminalGroupRelationships-Array</i> | | |
| Section 4.27 Changed <i>TerminalDetails-Array</i> from a Choice to a Sequence | | |
| Section 4.27.1 Terminal Details - Added " <i>PolarTerminalIndex</i> " to " <i>TerminalDetailsType</i> " and to " <i>FirstTerminalLocationType</i> " in Section 4.27.2.1 | | |
| Section 4.27.1.1 – Added new section for Terminal Center to include polar coordinates | | |
| Section 4.27.1.2 – Added " <i>Excluded</i> " to " <i>TerminalStatus</i> " | | |
| Section 4.27.2.1 First Terminal Location | | |
| Section 4.27.2.2 Terminal Numbering Pattern | | |
| Section 4.27.2.3 Added Grid Terminal Numbering Pattern | | |

Annex B (cont'd)

| |
|---|
| Section 4.27.2.4 Terminal Details Exceptions - Added " <i>Excluded</i> " to " <i>TerminalStatus</i> " |
| Section 4.27.2.4.1 Terminal Index - Added " <i>PolarTerminalIndex</i> " to " <i>AssociatedTerminalStatus</i> " |
| Section 4.27.2.4.2 Terminal Number – Added " <i>FromNumber</i> " and " <i>ToNumber</i> " to " <i>TerminalNumberType</i> " |
| Section 4.28 Updated Via Array |
| Section 4.29 Added a new section for 4.29 Geometric Dimensioning and Tolerancing |
| Section 4.30 Added a new section for Physical Model |
| Annex A Section A.1 – Update Contour example to fix one of the coordinates |
| |
| Revised Table of Contents |

| | | |
|---|----------------|------------------------|
| Issue: B | Date: Jun 2023 | Item Number: 11.2-1032 |
| | | |
| Section 2.1 Applicable Documents: Update reference to JESD30K | | |
| Section 2.3 JEDEC/IPC: Added new reference to J-Std-609 Standard | | |
| Section 4.4 Package: Added <i>AssemblyTechnologyType</i> , <i>CTE-ArrayType</i> , and <i>YoungsModulus-ArrayType</i> to <i>PackageType</i> . | | |
| Section 4.5 Package Terminal Position: Updated Package Terminal Position / Upper position to be consistent with Bottom position to support Chiplets | | |
| Section 4.5.8 Upper: Added new section for upper terminal position | | |
| Section 4.7.4 Gull-wing: Renamed extended terminal code from Shoulder to Modified to align with JESD30K | | |
| Section 4.11 Package Shape: Add in a new primitive shape <i>RectangleConcave</i> | | |
| Section 4.15.1 Terminal Position: Updated Terminal Group Terminal Position / Upper position to be consistent with Bottom position to support Chiplets | | |
| Added new section 4.16 for Assembly Technology | | |
| Added new section 4.17 for CTE - Array | | |
| Added new section 4.17 for Young Modulus | | |
| Section 4.18.1 Terminal Group – Array: Added Terminal Material and CTE-Array to Terminal Group Type | | |
| Section 4.18.3.1 Terminal Detail: Added ID key under terminal Details array for reference by the Fiducial Array | | |
| Section 4.19 Fiducial: Added new section for Fiducial Marking | | |
| Section 4.20 : Changed GD&T Datum-to-Element Map to unbounded under GD&T Datum-to-Element Map - Array | | |
| Section 4.20.2.1 Recommended Pad Or Hole Shape: Moved this structure from under Terminal Groups to an array under Recommended Footprints | | |
| Section 4.20.8 Keepout Region -Array: Moved this structure from under Package to under Recommended Footprints | | |
| | | |

Annex B (cont'd)

| | | |
|--|---------------------|------------------------|
| Issue: C | Date: November 2023 | Item Number: 11.2-1040 |
| Section 2.1 Applicable Documents: Update reference to JESD30L | | |
| Section 4.5.2 Dual: Added “Bottom-to-Upper” as a new Dual position to support Compressed Mount Technology terminal types., | | |
| Section 4.5.2.2 Dual Bottom-to-Upper: Added new structure for Bottom-to-Upper position | | |
| Section 4.7 Package Terminal Code: Updated diagram to include “Compressed Mount Technology” as a new Terminal type. | | |
| Section 4.17.1.2 Terminal: Added Compressed Mount Technology as a new terminal type | | |
| | | |

| | | |
|---|---------------------|------------------------|
| Issue: D | Date: February 2024 | Item Number: 11.2-1053 |
| Description of Change | | |
| Section 4.3 Linking the MPN to a specific Package Family Data set: Update section to include reference from Part Number to Die | | |
| Section 4.3.2 Linking the Manufacturing Part Number to Physical Model Content: Corrected the PhysicalModelKey assignment to PhysicalModel/ID, | | |
| Added new Section 4.3.3 “Linking the Manufacturing Part Number to Die Content” | | |
| Section 5 Package Section - Package: Updated diagram to include the addition of Die-Array | | |
| Updated images in sections 5.6, 5.8, and A.4.2 thru A.4.7 to include ValueSetGroup | | |
| Updated image in section 5.7.1 to include Outline-Array | | |
| Section 5.13.3 Terminal Detail – Array: “Terminal Number Pattern” is made unbounded | | |
| Section 5.13.3.2 Terminal Number Pattern: Added Prefix and Suffix under Terminal Number Pattern | | |
| Section 5.13.3.2.2 Sequential: Added Prefix, Start and Suffix under both Numerical and Alphabetical Sequence. | | |
| Section 5.13.3.2.3 Grid: Updated image. | | |
| Section 5.13.3.2.1 First Terminal Location: Updated elements in “Location Relative-to-Package Center” type from optional to mandatory under choice. | | |
| Section 5.13.3.2.2 Sequential: Updated Numerical and Alphabetical Sequence | | |
| Added new Section 6 “Package Section – Die-Array” | | |

Annex B (cont'd)

| | | |
|--|-------------------|------------------------|
| Issue: E | Date: August 2024 | Item Number: 11.2-1059 |
| Description of Change | | |
| Section 4.1, and section 4.2: Update sections to align with modifications performed at the JEP30 parent structure | | |
| Section 5 Package Section – Package: Updated name of the JESD30 document. Added Package Status. Changed xs:integer with Min of 1 to | | |
| Changed xs:integer with Min of 1 to “Min Integer Of One Type” throughout the document | | |
| Section 5.13.1 Terminal Group – Array: Make Terminal Location unbounded. Added Pattern Group section. | | |
| Section 5.13.1.5 Terminal Location: Updated Terminal location to make it unbounded. | | |
| Section 5.13.1.5.1 Standard Array: Make Angle optional | | |
| Section 5.13.1.6: Added new section for Pattern Groups | | |
| Section 5.13.1.7.22.3: Updated image for Void Status | | |
| Section 5.13.3.2.1: Update image to include choice of Terminal Pattern ID or Pattern Group ID. | | |
| Section 5.13.3.2.4.1: Update Image to include Terminal Location ID, and Pattern Group ID | | |
| Section 5.13.3.2.4.2: Updated image to make various branches unbounded | | |
| Section 5.16 Recommended Footprint – Array: Updated section with Footprint Name, Footprint Extended Name, Polarity. Made Soldermask and Pastemask Layer arrays optional. | | |
| Section 5.16.2 Recommended Pad Or Hole Shape – Array: Added new element for Pad Group To Pad Group Relationship | | |
| Section 5.16.2.1 Recommended Pad Or Hole Shape: Make Location unbounded. Added Pattern Group. | | |
| Section 5.16.2.1.1.1.1.1 Standard Array: Make dx, dy, Angle and VoidGroupLowerLeftVoidCenter optional. | | |
| Section 5.16.2.1.3 Location: Updated Location image to make it unbounded. | | |
| Section 5.16.2.1.3.1 Standard Array: Updated image to make dx, dy, Angle and LandGroupLowerLeftPad-or-HoleCenter optional | | |
| Section 5.16.2.1.4: Added new section for Pattern Groups | | |
| Section 5.16.2.1.5: Added new section for Land Pattern Span | | |
| Section 5.16.2.1.6: Added new section for Land Pattern Spacing | | |
| Section 5.16.2.2: Added new section for Pad Group To Pad Group Relationship | | |
| Section 5.16.7.1 Restrictive layer: Update element name to InnerLayer | | |
| Section 5.16.9 Soldermask Layer – Array: Make Soldermask Layer – Array optional | | |
| Section 5.16.10 Pastemask Layer – Array: Revised section to align with previous sections | | |
| Section 6 Package Section – Die-Array: Added Process Technology | | |

This page intentionally left blank.



Standard Improvement Form**JEDEC Standard No. JEP30-P100E**

The purpose of this form is to provide the Technical Committees of JEDEC with input from the industry regarding usage of the subject standard. Individuals or companies are invited to submit comments to JEDEC. All comments will be collected and dispersed to the appropriate committee(s).

If you can provide input, please complete this form and return to:

JEDEC
Attn: Publications Department
3103 North 10th Street
Suite 240 South
Arlington, VA 22201-2107

Email: angies@jedec.org

1. I recommend changes to the following:

☐ Requirement, clause number _____

☐ Test method number _____ Clause number _____

The referenced clause number has proven to be:

☐ Unclear ☐ Too Rigid ☐ In Error

☐ Other _____

2. Recommendations for correction:

| |
|--|
| |
| |
| |
| |

3. Other suggestions for document improvement:

| |
|--|
| |
| |
| |
| |

Submitted by

Name: _____

Phone: _____

Company: _____

E-mail: _____

Address: _____

City/State/Zip: _____

Date _____

