Digital Metrology Standards Consortium DMSC

Model-Based Characteristics v1.0

(MBC 1.0)

Persistent Identification and Related Digital Practices

MBC by DMSC

Designation:

DMSC MBC 1.0 – 2026

Approved American National Standard ANSI

Prepared and Published by:
Digital Metrology Standards Consortium, Inc. (DMSC)
3245 Latta Road, #16595
Rochester, NY 14616
Tel: (585) 451.5800
Email: Director@QIFStandards.org

www.DMSC-inc.org

Tables of Contents

Contents

Table of Contents iii
Contents iii
Figures xi
Tables xiii
Foreword xiv
DMSC Members xiv
MBC Working Group & Contributors xiv
DMSC Standards Committee xv
ANSI Statement xv
Model-Based Characteristics (MBC) subjects xv
Introduction xvi
1 Scope and Purpose 1
1.1 Contents of this document 1
1.2 Structure 1
1.3 Purpose 1
1.3.1 Model-Based Product Characteristic (PC) Tag 1
1.3.2 Machine-Readable Product Characteristic (PC) UUID 2
1.3.3 Bill of Characteristics (BOC) of PC Instances Example 3
1.3.4 Product Characteristic (PC) Designation with Criticality Classification Example 4
1.3.5 Product Characteristic (PC) Designation with all Augmentations Example 4
1.3.6 Product Characteristic (PC) and Augmentations Generation 5
1.4 Support within the Quality Information Framework (QIF) 6
1.5 Data Structure of the Model-Based Characteristics (MBC) 6
2 Normative References 8
3 Terms and Definitions 9
3.1 Annotation 9
3.2 Attribute 9
3.3 Bill of Characteristics (BoC) 9
3.4 Characteristic 9

3.5 Characteristic Augmentation 9
3.6 Characteristic, Critical – Critical Characteristic 9
3.7 Characteristic, Drawing-Based (DBC) – Drawing-Based Characteristic 9
3.8 Characteristic, Driven – Driven Characteristic 9
3.9 Characteristic, Model-Based (MBC) – Model-Based Characteristic 9
3.10 Characteristic, Operation (OC) – Operation Characteristic 9
3.11 Characteristic, Product (PC) – Product Characteristic 9
3.12 Characteristic, Service (SC) – Service Characteristic 9
3.13 Characteristic, Verification – Verification Characteristic 10
3.14 Criticality Classification (CC) 10
3.15 Derivative, Embellished – Embellished Derivative 10
3.16 Derivative, Equivalent – Equivalent Derivative 10
3.17 Designator 10
3.18 Designator, Characteristic – Characteristic Designator 10
3.19 Designator, Characteristic Augmentation – Characteristic Augmentation Designator 10
3.20 Designator, Criticality Classification - Criticality Classification Designator 10
3.21 Designator, Product Requirement Association - Product Requirement Association Designator 10
3.22 Designator, Verification Plan Requirement - Verification Plan Requirement Designator 10
3.23 Definition, Product – Product Definition 10
3.24 Definition, Process – Process Definition 10
3.25 Identifier, Product Characteristic Base - Product Characteristic Base Identifier 10
3.26 Identifier, Product Characteristic Extension - Product Characteristic Extension Identifier 10
3.27 Identifier, Product Characteristic Instance - Product Characteristic Instance Identifier 11
3.28 Identifier, Product Characteristic Reference - Product Characteristic Reference Identifier 11
3.29 Model 11
3.30 Model, 3D – 3D Model 11
3.31 Model, 3D CAD – 3D CAD Model 11
3.32 Model Digital – Digital Model 11
3.33 Model, Model-Based Definition (MBD) – Model-Based Definition 11
3.34 Numeric 11
3.35 Product and Manufacturing Information (PMI) 11
3.36 Product Requirement Association (PRA) 11
3.37 Quality Information Framework (QIF) 11

3.38 QIF Persistent Identifier (QPId) 11
3.39 Requirement, Customer – Customer Requirement 11
3.40 Requirement, Product – Product Requirement 11
3.41 Requirement, Verification – Verification Requirement 12
3.42 Tag 12
3.43 Tag, Characteristic – Characteristic Tag 12
3.44 Tag, Feature Reference – Feature Reference Tag 12
3.45 Tag, General – General Tag 12
3.46 Tag, Product Characteristic (PC) – Product Characteristic Tag 12
3.47 Tag, Product Characteristic Instance (PC Instance Tag) 12
3.48 Tag, Product Characteristic Reference (PC Reference Tag) 12
3.49 Universally Unique Identifier (UUID) 12
3.50 Verification 12
3.51 Verification Plan Requirement (VPR) 12
4 Characteristics 13
4.1 Types 13
4.1.1 Digital, Model-Based vs Electronic, Document-Based 13
4.1.2 Lifecycle Discipline 13
4.1.3 Augmentation Based Characteristics 14
4.2 System Requirements 15
4.2.1 Attributes 15
4.2.2 Associativity 15
4.2.3 Placement, Alignment & Annotation Planes 15
4.2.4 Display Management 16
4.2.5 Query 17
4.2.6 Behaviors 17
4.2.7 Designators 18
4.3 Augmentations 19
4.3.1 Criticality Classifications (CC) 19
4.3.2 Product Requirement Associations (PRA) 19
4.3.3 Verification Plan Requirement (VPR) 19
4.3.4 Augmentations Relative Order 19
5 Product Characteristics 20

5.1 Identification 20
5.1.1 Product Characteristic Identification Framework 20
5.2 Data Structure 22
5.3 Data Taxonomy 23
5.3.1 Product Characteristic Identification Framework Taxonomy 23
5.3.2 Product Characteristic Data Taxonomy 24
5.3.3 Product Characteristic with Augmentations 24
5.4 Data Objects 24
5.4.1 ProductCharacteristic 24
5.4.2 ProductCharacteristicDesignator 24
5.4.3 ProductCharacteristicDesignatorSymbol 25
5.4.4 ProductCharacteristicBaseIdentifier 25
5.4.5 ProductCharacteristicReferenceIdentifier 26
5.4.6 ProductCharacteristicInstanceIdentifier 27
5.4.7 ProductCharacteristicInstanceTag with Extension Identifiers 28
5.4.8 ProductCharacteristicInstanceUUID with Extension Identifiers 30
5.4.9 Product Characteristic Instance Identifier Format 30
5.4.10 PC Reference Tag with Augmentations 33
5.5 Annotation Applications 34
5.6 Single Product Characteristic with Single Application (Feature) 35
5.6.1 Geometric Tolerance 35
5.6.2 Dimensional Tolerance (limit, plus/minus, single Limit (MAX, MIN)) 35
5.6.3 Product Definition Notes 36
5.6.4 General Note 37
5.6.5 Individual General Notes 37
5.6.6 Supplemental Document General Note 38
5.6.7 Flag or Local Note 39
5.6.8 Supplement Note 40
5.6.9 Surface Texture 41
5.6.10 Thread Specification 42
5.6.11 Composite Tolerance as Separate PC Reference Tags 43
5.6.12 Individual Multiple Segment Feature Control Frame 44
5.6.13 Mixed Multiple Segment Feature Control Frame 44

5.6.14 Chamfer Dimensions....45
5.7 Single Product Characteristic with Defined Multi-Faceted Annotation....45
5.7.1 Surface Texture with Multiple Elements....46
5.7.2 Thread Spec with Multiple Elements....47
5.7.3 Welding Symbol with Multiple Elements....48
5.8 Single Product Characteristic with Derived Multi-Faceted Annotation....50
5.8.1 Hole with Depth Annotation....50
5.8.2 Counterbore with Depth Annotation....51
5.8.3 Multi-Counterbore Annotation....52
5.8.4 Hole with Countersink with Taper & Depth Annotation....53
5.8.5 Composite Tolerance as Single Feature Control Frame....54
5.8.6 Group General Note:....55
5.8.7 Chamfer of 45° with Tolerances Annotation Note....56
5.8.8 Per Unit Basis....58
5.8.9 Mixed Multi-Faceted Feature Control Frame....59
5.9 Single Product Characteristic with Multiple Feature References Applications....60
5.9.1 Dimensional Tolerance with Multiple Feature References....60
5.9.2 Geometric Tolerance with Multiple Feature References....64
5.9.3 Multiple Direct Leader Annotation....65
5.9.4 Tolerance All Around Specification....67
5.9.5 Tolerance All Over Specification....68
5.9.6 Unless Otherwise Specified (UOS) Tolerance....70
5.9.7 Tolerance Applied Between Points....71
5.9.8 Simple Bidirectional Position Pattern....72
5.9.9 Simple Position Pattern....74
5.9.10 Position of Counterbored Holes....80
5.10 Single Product Characteristic including Repetitive Group Applications....86
5.10.1 Repetitive Group of Multiple Features....86
5.10.2 Repetitive All-Around Profile Group....90
5.11 Single Product Characteristic with Mixed Multiple Extension Applications....94
5.11.1 Multi-Facet Annotation Applied to Multiple Features/Faces....94
5.12 Product Characteristics within an Assembly....96
5.12.1 Separable Assembly....96

5.12.2 Inseparable Assembly 101
5.13 Product Characteristics in Support Documents (Text-based support artifacts) 101
5.13.1 Example 101
5.13.2 Application: 101
5.13.3 BOC Example 102
5.14 Product Characteristics in System Information Models 102
5.14.1 Example 102
5.14.2 Application: 103
5.15 Product Characteristics on Surrogate Surfaces 103
5.16 Product Characteristics on Representative Test Artifacts 103
5.17 Product Characteristics on Embellished Derivatives 104
5.18 Application to Serialized Parts which enables Digital Twins 104
5.18.1 Enables Digital Twins 104
5.18.2 Serialized Part Examples 104
5.19 Non-Product Characteristics, General Tag 104
5.19.1 Example 105
6 Criticality Classifications (CC) Augmentation 106
6.1 Identification 106
6.2 Data Structure 106
6.3 Data Taxonomy 106
6.4 Data Objects 108
6.4.1 CriticalityClassification 108
6.4.2 CriticalityClassificationDesignator 108
6.4.3 CriticalityClassificationDesignatorSymbol 108
6.4.4 CriticalityClass 108
6.4.5 CriticalityLevel 108
6.4.6 CriticalityCategory 110
6.4.7 CriticalityCaveat 112
6.4.8 CriticalityClassificationText 112
7 Product Requirement Association (PRA) Augmentation 113
7.1 Identification 113
7.2 Data Structures 113
7.3 Data Taxonomy 113

7.4 Data Objects 114
7.4.1 ProductRequirementAssociation 114
7.4.2 ProductRequirementAssociationDesignator 114
7.4.3 ProductRequirementAssociationDesignatorSymbol 114
7.4.4 ProductRequirementAssociation Text 114
7.4.5 ProductRequirementAssociationUUID 115
7.4.6 ProductRequirementAssociationReferenceLink 115
8 Verification Plan Requirement (VPR) Augmentation 116
8.1 Identification 116
8.2 Data Structures 116
8.3 Data Taxonomy 117
8.4 Data Objects 117
8.4.1 VerificationPlanRequirement 117
8.4.2 VerificationPlanRequirementDesignator 117
8.4.3 VerificationPlanRequirementDesignatorSymbol 118
8.4.4 VerificationPlanRequirementText 118
8.4.5 VerificationMethod 118
8.4.6 VerificationContext 120
8.4.7 VerificationAlternative 121
8.4.8 SamplingPlan 121
9 MBC Structure Information Model 128
9.1 Product Characteristics with Augmentations Information Model 128
9.2 Product Characteristics Identification Framework Data Model 129
9.3 Product Characteristics Extension Identifier Data Model 130
9.4 Product Characteristic Data Model 131
9.5 Criticality Classification Augmentation Data Model 132
9.6 Product Requirement Association Augmentation Data Model 133
9.7 Verification Plan Requirement Augmentation Data Model 134
10 Informative Appendix 135
10.1 Annex A – Form & Proportion of Symbols 136
10.1.1 Preferred Form and Proportion 136
10.1.2 Compound Symbol Spacing 136
10.1.3 Symbol Alignment 136

10.2 Annex B – QIF Characteristic Designator Type 138
10.2.1 Characteristics 138
10.2.2 CharacteristicDefinitionBaseType / Characteristic Designator 139
10.2.3 CharacteristicNominalBaseType / Characteristic Designator 139
10.2.4 CharacteristicItemBaseType / Characteristic Designator 139
10.2.5 Characteristic Designator 139
10.2.6 Characteristic Designator Type 140
10.2.7 Characteristic Designator Type / Balloon 144
10.3 Annex C – Selection Criteria for a Characteristic Designator Symbol 145
10.4 Annex D – MBC Attributes Names for Applications 146
10.4.1 MBC_PC (Product Characteristic) 146
10.4.2 MBC_CC (Critical Classification) 146
10.4.3 MBC_PRA (Product Requirement Association) 146
10.4.4 MBC_VPR (Verification Plan Requirement) 146
10.4.5 MBC_GT (General Tag) 146
10.5 Annex E – Acronyms & Abbreviations 147
10.5.1 Characteristic Tags 147
10.5.2 PC Augmentations 147
10.5.3 Criticality Levels 147
10.5.4 Criticality Category 147
10.5.5 Verification Method 148
10.5.6 Verification Context 148
10.5.7 Verification Alternative 148
10.5.8 Sampling Plan 148
10.5.9 Statistical Control Limit Variables 149
10.5.10 Product Characteristic Extension Identifiers 149
10.5.11 Additional Acronyms 149
10.6 Annex F – System Modeling Language (SysML) Block Diagram Modeling 151
10.6.1 Block Diagrams 151
10.6.2 Relating Blocks 151
10.7 Annex G – Verification Method Types Detail 153
10.8 Annex H – Alternate Numerical Approach for Product Characteristic Tags 157
10.9 Annex I –Data Reporting Requirements (DRR) Augmentation 158

10.9.1 Identification 158
10.9.2 Data Structures 158
10.9.3 Data Taxonomy 158
10.9.4 Data Objects 158
10.10 Annex J– QIF 3.0 Mapping / Comparisons 167
10.11 Annex K – Control Limit Mapping to ISO 3534-2 and ISO 22514-3 Symbols 170
10.12 Annex L – Variations in implementation 172
10.13 Annex M – MBC Feedback Request 173

Figures

Figure 1: Visualization of a Model-Based Definition with Product Characteristic Tags 2
Figure 2: Visualization of a Model-Based Definition with Product Characteristic Tags & UUIDs 3
Figure 3: Example Product Characteristic (PC) Tag with Criticality Designator 4
Figure 4: PC Reference Tag with All Augmentations 5
Figure 5: Block Diagram - Product Characteristic with Augmentations Information Model 7
Figure 6: Block Diagram - Product Characteristic Identification Framework 21
Figure 7: Block Diagram - Product Characteristic Extension Identifiers 22
Figure 8: Block Diagram - Product Characteristic Data Model 23
Figure 9: PC Reference Tag Format Examples on Single Annotations 25
Figure 10: PC Reference Tags for Single Line, Multiple Verification Requirement Annotation 36
Figure 11: PC Reference Tags for Multi- Line, Multiple Verification Requirement Annotation 36
Figure 12: Individual General Note Example 37
Figure 13: Supplemental Document with Verification Requirements Example 38
Figure 14: Supplemental Document with Verification Requirements Application 39
Figure 15: PC Reference Tag Example on Marking Flag Note Annotations 40
Figure 16: Average Roughness Surface Texture 41
Figure 17: Thread Specification Applied to Multiple Features 42
Figure 18: Composite Position Tolerance with Segment Reference Tags 43
Figure 19: Two-Segment Feature Control Frame Tolerance 44
Figure 20: Mixed Multiple Segment Feature Control Frame 44
Figure 21: Multi-Faceted Surface Texture Symbol Example 46
Figure 22: Multi-Faceted Surface Texture Symbol Application 46
Figure 23: Multi-Faceted Thread Specification Designation Example 47
Figure 24: Multi-Facet Thread Specs Applied to Multiple Features 47
Figure 25: Welding Symbol Elements 49
Figure 26: Single PC Reference Tag for Single-line, Multiple Verification Requirement Annotation 51
Figure 27: Multi-Counterbore Annotation 52
Figure 28: Multi-Facet Dimensional Tolerance 53
Figure 29: Composite Position Tolerance with Single Reference Tag Example 55
Figure 30: Composite Position Tolerance with Single Reference Tag Application 55
Figure 31: Group General Note Example 55

Figure 32: Group General Note Application 56
Figure 33: 45° Chamfer Tolerance Note 57
Figure 34: Chamfer Tolerance 57
Figure 35: Form with Per Unit Basis 58
Figure 36: Form with Per Unit Basis with Reference and Instance Tags 59
Figure 37: Mixed Multiple Segment Feature Control Frame 59
Figure 38: Mixed Multiple Segment Feature Control Frame 60
Figure 39: Repetitive Feature Size Tolerance with Reference Tag 61
Figure 40: Repetitive Feature Size Tolerance with PC Instance Tags using Feature Reference Tag 61
Figure 41: Repetitive Feature Size Tolerance with Instance Tags using Feature Reference UUID 62
Figure 42: Repetitive Feature Size Tolerance with Instance Tags using Feature Sequencing 63
Figure 43: Repetitive Feature Geometric Tolerance 64
Figure 44: Multiple Direct Leader Geometric Tolerance Annotation 65
Figure 45: Multiple Direct Leader Flag Note Annotation 66
Figure 46: All-Around Profile Tolerance 67
Figure 47: All-Around Profile Tolerance with Instance Tags 68
Figure 48: All-Over Profile Tolerance 69
Figure 49: All-Over Profile Tolerance with Instance Tags 69
Figure 50: Profile Tolerance between Points Annotation 71
Figure 51: Profile Tolerance between Points Application 71
Figure 52: Bidirectional Positional Tolerance, Rectangular Coordinate Method with PC Reference Tags. 72
Figure 53: Bidirectional Positional Tolerance, Rectangular Coordinate Method with PC Instance Tags ... 73
Figure 54: Bidirectional Positional Tolerance, Polar Coordinate Method 74
Figure 55: Simple Position Pattern of Holes 75
Figure 56: Simple Position of Bolt-Hole Pattern with Feature Instance (informative) 76
Figure 57: Simple Position Pattern of Slots with PC Reference Tags 77
Figure 58: Simple Position Pattern of Slots with PC Instance Tags 78
Figure 59: Simple Composite Position of Pattern Feature of Size 79
Figure 60: Composite Position of Pattern Feature with Feature Instance (informative) 80
Figure 61: 4X Counterbored Holes with One Position Tolerance 81
Figure 62: 4X Counterbored Holes with Separate Position Tolerances 83
Figure 63: 4X Counterbores Related to Datum Holes Individually 85
Figure 64: Repetitive Group of Multiple Features Example 87
Figure 65: Repetitive Group of Multiple Features Application 88
Figure 66: Repetition of All-Around Profile Group Example 90
Figure 67: Repetition of All-Around Profile Example as Individual Profile Group Application 91
Figure 68: Query Highlighting of a Pocket Profile Group PC Instance Tags 91
Figure 69: Repetition of All-Around Profile Example as Multiple Faces Profile Group Application 92
Figure 70: Query Highlighting of a Face on Pocket Profile Group for PC Instance Tags 93
Figure 71: Multi-Faceted Annotation Applied to Multiple Features, PC Reference Tags 95
Figure 72: Multi-Faceted Annotation Applied to Multiple Features, PC Instance Tags 96
Figure 73: Lifting Block Assembly with Find Numbers 97
Figure 74: Hook Sub-Assembly with Part Identifiers 98
Figure 75: Lifting Block Assembly with Part Identifiers and Product Characteristic Tags 98

Figure 76: Requirements Diagram shown Requirements Satisfied identify by Product Characteristics.. 103
Figure 77: General Tags for Non-Product Characteristic Annotations 105
Figure 78: Block Diagram - Criticality Classifications Information Model 107
Figure 79: Block Diagram - Product Requirement Associations Information Model 113
Figure 80: Block Diagram - Verification Plan Requirement Information Model 116
Figure 81: Block Diagram - Product Characteristic with Augmentations Information Model..... 128
Figure 82: Block Diagram - Product Characteristic Identification Framework 129
Figure 83: Block Diagram - Product Characteristic Extension Identifiers 130
Figure 84: Block Diagram - Product Characteristic Information Model..... 131
Figure 85: Block Diagram - Criticality Classifications Information Model..... 132
Figure 86: Block Diagram - Product Requirement Associations Information Model 133
Figure 87: Block Diagram - Verification Plan Requirement Information Model..... 134

Tables

Table 1: PC Reference Tag Designator Presentation Examples 24
Table 2: Lifting Block Assembly Item List with Product Characteristic Tags 97
Table 3: Hook Sub-Assembly Item List with Product Characteristics 98
Table 4: Assembly Level BOC with PC Tags and PC UUIDs 99
Table 5: Subassembly Level BOC with PC Tags and PC UUIDs 100
Table 6: Record of Assembly Example with PC Tags and PC UUIDs 101
Table 7: BOC from Supplemental Product Definition Documents 102
Table 8: Textual Criticality Classification Designators 112

Foreword

The Digital Metrology Standards Consortium (DMSC, Inc.) is an American National Standards Institute (ANSI) accredited standards developing organization, as well as an A-Liaison to the International Organization for Standardization (ISO) Technical Committee (TC) 184. The mission of the DMSC is to identify urgently needed digital standards in the field of digital metrology, and to promote, foster, and encourage the development and interoperability of these standards, along with related and supporting standards that will benefit the manufacturing industry. More information about the DMSC can be found at www.dmsc-inc.org.

This Model-Based Characteristic (MBC) standard is a DMSC standard that was developed by a professional group of manufacturing metrologists, software developers, industrial users, and innovative thought leaders worldwide. Contributors that have contributed to the development of this standard include:

DMSC Members

- 2BMobile
- AAT3D
- Action Engineering
- Capvidia
- Deere & Company
- Elysium
- GemDT
- IIGDT
- HighQA
- Honeywell FM&T (KCNSC)
- Innovalia Metrology
- InnovMetic
- ITI
- Johns Hopkins API
- LK Metrology
- Lockheed Martin
- Los Alamos National Laboratory
- Metrologic
- Metrosage
- Mitutoyo
- NIST
- QVI
- Origin International
- Pratt & Whitney
- Purdue University
- Renishaw
- Samwell Group
- Schneider Electric
- Sigmetrix
- Smart Product Development
- Tec-Ease, Inc.
- TechAzul
- Thomas Kramer Consulting
- UNC-Charlotte
- Zeiss

MBC Working Group & Contributors

- Mark Nielsen, TechAzul (Chair)
- Curtis Brown, Honeywell FM&T (Vice-Chair)
- Ray Admire, QIF Solutions
- Rosemary Astheimer, NIST
- Ryan Bounds, Newport News Shipbuilding
- Kevin Braun, John Deere
- Larry Bergquist, Smart Product Dev.
- Daniel Campbell, Rubypoint
- Andrew Cheetham, Renishaw plc
- Jan De Nijs, Lockheed Martin
- Nathan Denver, L3Harris
- Zak Delphia, GE Aerospace
- Geoff Foulds, Origin International Inc.
- Steven Fulkerson, Self

- Sam Gambrell, Lockheed Martin
- Ryan Gelotte, John Hopkin, APL
- Joseph Graham, Sandia National Labs
- Blake Gorowsky, Pratt & Whitney
- Jeremy Hamilton, Deere & Co.
- Gary Heinly, Garycan Software LLC
- Simon Héroux, InnovMetric Logiciels
- Jennifer Herron, Action Engineering
- Duane Hess, Action Engineering
- Mario Hidalgo, Honeywell Aerospace
- Evan Kessick, Elysium Inc.
- Francois Klinkenberg, Herstal group
- Tom Kramer, Thomas Kramer Consulting
- Christopher Lalonde, Litens Auto Partnership
- Larry Maggiano, Mitutoyo America Corp.
- Michael McCellen, Collaboration Synergies
- Fred McMaier, Lockheed Martin
- John Tom Meeks, Boeing Co.
- Satoshi Nakamura, QVI Japan Inc.
- Andrew Pierce, GE Appliances
- Frank Popielas, Capvidia
- Allie Renard, GE Aerospace
- Ryan Scott, Belcan, LLC
- Jacob Sherwood, Pratt & Whitney
- Ray Stahl, 2BMobile
- Robert Stone, Origin International
- Jan Stothfang, B&W software
- Annalise Suzuki, Elysium
- Mark Thomas, DMSC
- Hermit Vega, Pratt & Whitney
- Benny Yap, Lam Research

DMSC Standards Committee

This document was written by the DMSC's Model-Based Characteristics (MBC) Working Group and given consensus approval via the DMSC's Standards Committee for future standardization. The DMSC Board of Directors have agreed to disseminate this document as a DMSC Standard. More information about DMSC's MBC effort can be found via www.qifstandards.org/about-dmsc/.

ANS Statement

The *Model-Based Characteristics v1.0 (MBC 1.0), Persistent Identification and Related Digital Practices* standard was approved by DMSC and published as **DMSC MBC v1.0–2024**. Following review and approval through ANSI's consensus-based standards process, it was approved as an **American National Standard (ANS)** and designated as **DMSC MBC v1.0–2026**.

Model-Based Characteristics (MBC) subjects

The Model-Based Characteristics (MBC) standard, version 1.0, consists of the following subject areas under the general title: *Model-Based Characteristics (MBC) – Persistent Identification and Related Digital Practices:*

- Introduction, Scope and Purpose
- Normative References
- Terms and Definitions
- Characteristics
- Product Characteristics (PC)
- Criticality Classification (CC) Augmentations
- Product Requirement Association (PRA) Augmentations
- Verification Plan Requirement (VPR) Augmentations
- MBC Structure Information Model
- Informative Appendix

Introduction

This Model-Based Characteristics (MBC) standard defines nomenclature, definitions, symbols, data structures, and practices for identifying, communicating, and exchanging model-based characteristics with various optional augmentations through both a logical data model and supporting documentation.

Specifically, this standard provides a common approach for tagging and uniquely identifying product characteristics that are of interest for the product realization process. Product characteristic tagging is primarily used for identifying a list of characteristics required to verify and accept product. Additionally, it can enable explicit referencing for product definition, change control, reporting on product non-conformance, and obfuscating confidential or classified product definition information. Moreover, this standard provides optional augmentations for communicating criticality classifications, product requirement associations, and verification plan requirements. Finally, this standard shall be applicable for both human-readable communication and machine-readable applications that can enable the digital thread.

Drawings and model-based definition models that communicate a product's form, fit, and function will contain product characteristics in the form of annotations such as dimensional tolerances, geometric tolerances, surface textures, flagged or local notes, general notes, and other specifications. The identification tagging of selected annotations as product characteristics can be enclosed within a specified shaped symbol to help communicate information. This standard intends to create a standardized symbology for identifying a Product Characteristic and any of its augmentations. Furthermore, this standard describes Product Characteristic behaviors within a model-based system.

This standard refers to Model-Based Definition (MBD) which is defined along with other commonly used terms in section 3 and Appendix E. While this standard is primarily written for Model-Based workflows, it also applies to 2D drawing workflows when used in conjunction with the QIF standard. QIF contains the data structures to capture the information defined in this standard and allows a user to abstract these concepts from the underlying product definition, whether it be a Model-Based Definition Model or a 2D Drawing.

1 Scope and Purpose

1.1 Contents of this document

The scope of this standard establishes a baseline for characteristics that are applied to product definition. This version of the standard specifically focuses on product characteristics (PC) and their optional augmentations of criticality classifications (CC), product requirement associations (PRA), and verification plan requirements (VPR). This standard defines the common nomenclature, definitions, symbols, data structures and practice, for the representation and communication of characteristics. Furthermore, this standard introduces a Product Characteristic identification framework that specifies both the use of locally unique, human-readable tag identifiers coupled with universally unique, machine-readable identifiers (UUIDs) for each characteristic.

This standard also describes model-based system behaviors for applications that produce and/or use product characteristics.

Finally, the intent of this standard is to complement and contribute to extending the Quality Information Framework (QIF) in the QIF's area of Characteristics and Characteristic Designators.

1.2 Structure

The structure of Model-Based Characteristics (MBC) consists of both a human-readable component and a machine-readable component. The human-readable component identifiers, also known as Product Characteristic Tags, shall be designated, and may be displayed within a graphical symbol (e.g., elongated hexagon) or as a common text string surrounded with designated characters (e.g., <PC007>). The machine-readable component identifiers, also known as Product Characteristic UUIDs, adhere to the ISO/IEC 9834-8 standard as a universal unique identifier. Both components have their own specific data structure, attributes, and relationships with other objects, which will enable software programs and developers to author and consume characteristics for downstream use and additional functions. A Product Characteristic shall be comprised of Product Characteristic identifiers, which may be supplemented with optional augmentations such as criticality classifications, product requirement associations, and/or verification plan requirements.

The data structure of the Product Characteristic and each of its optional augmentations are contained within this standard and are information modeled using OMG System Modeling Language (SysML) block diagrams.

1.3 Purpose

This standard's purpose is to address the identification, nomenclature and representation of product characteristics and their optional augmentations. The following summarizes important concepts being addressed in this standard.

1.3.1 Model-Based Product Characteristic (PC) Tag

The identification and use of PC Tags are a primary objective of this standard. Figure 1 shows an example of a model-based definition (MBD) model containing associated tolerance annotations with their Product Characteristic Tags. Specifically, this example shows that the

16mm diameter hole's annotation was assigned a PC Reference Tag <PC004> as indicated by the PC Reference Tag callout. This figure also illustrates the MBD model's capability of cross-highlighting four holes surfaces – in red (or other color) - upon selection of PC Reference Tag <PC004> as indicated by the "Query" callout. The PC Reference Tag display is optional, and its visibility may be controlled by the user or application.

Other annotations callouts that are not functionally a Product Characteristic may be designated as General Tags (GT) see section 5.19.

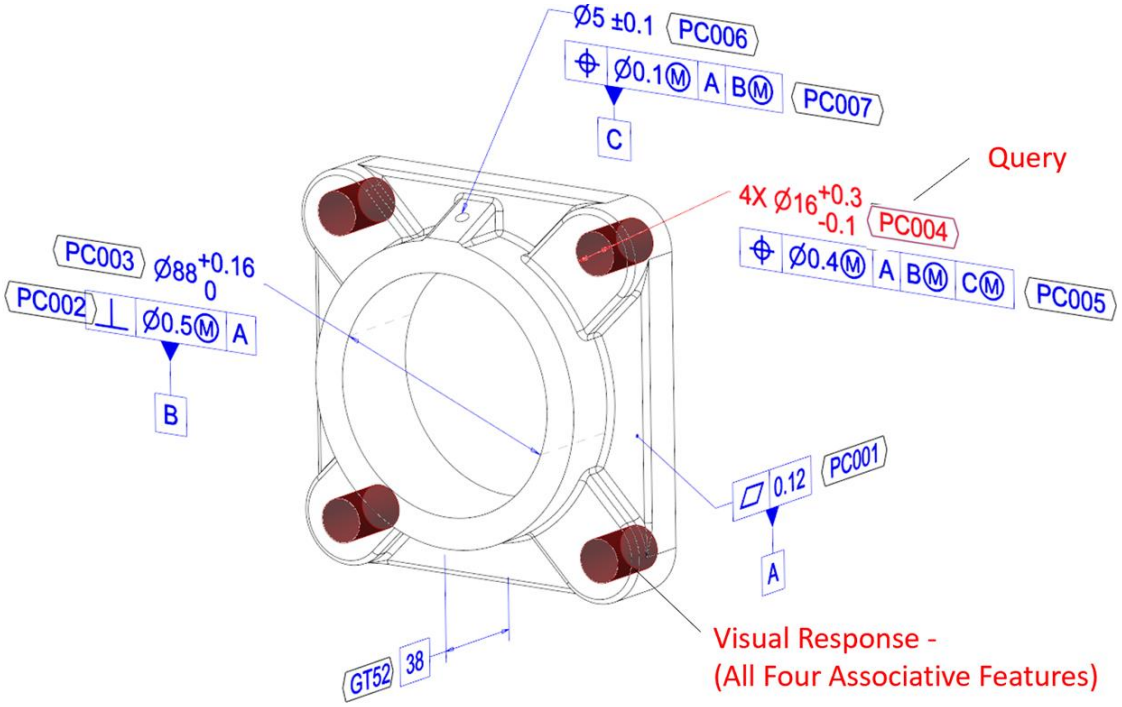


Figure 1: Visualization of a Model-Based Definition with Product Characteristic Tags

1.3.2 Machine-Readable Product Characteristic (PC) UUID

Assigning persistent machine-readable universally unique identifiers (UUID) to the human-readable Product Characteristic tags enables connection points that established a digital thread at the product characteristics definition level. Figure 2 extends the Figure 1 MBD model with PC Tags example by revealing the machine-readable product characteristics UUID for each human-readable Product Characteristic tag. For example, the PC UUID for PC Tag <PC004> is "DA8612FE-B1E4-423B-8191-B746E224C595" as indicated by the PC UUID callout. The PC UUID display is optional, and its visibility may be controlled by the user or application.

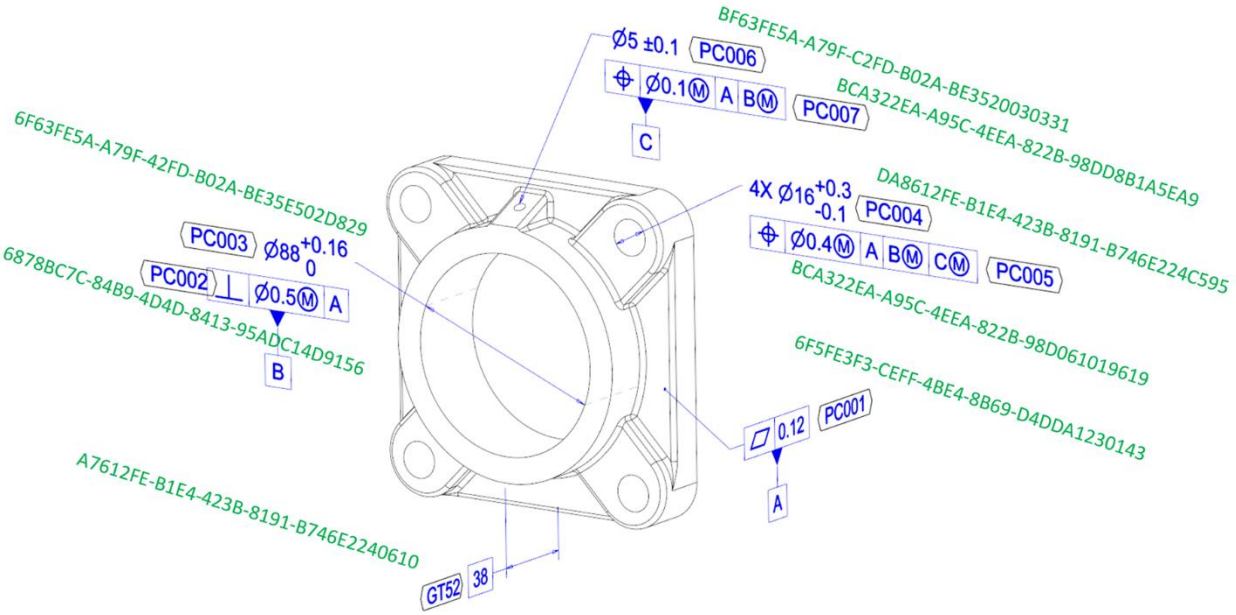


Figure 2: Visualization of a Model-Based Definition with Product Characteristic Tags & UUIDs

1.3.3 Bill of Characteristics (BOC) of PC Instances Example

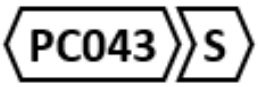
Manufacturing quality engineering often collects a product's verification requirements in the context of a Bill of Characteristics (BOC). This is a list of product characteristics, which identify what needs to be verified to assure an acceptable product. The table below shows an example of a bill of characteristics (BOC), which lists all of the product characteristics instances, their corresponding instance UUIDs, along with their descriptions as illustrated in Figure 2. Explanations of the contents of this BOC table will be discussed within this standard, but in general, the BOC table describes each PC instance. For the case of PC Reference Tag <PC001> on the flatness tolerance, there is only a single feature instance application, PC001. Furthermore, the UUID for the PC001 is 6F5FE3F3-CEFF-4BE4-8B69-D4DDA1230143. For the case of PC reference Tag <PC004>, the diameter tolerance is applied to multiple (i.e., four) features. The feature instance application of PC Reference Tag <PC004> will be uniquely identified per a method described in section 5.9.1, which could result into PC Instance Tags of PC004.1, PC004.2, PC004.3, and PC004.4. Likewise, the PC Instance UUIDs for each of the <PC004> instances would be DA8612FE-B1E4-423B-8191-B746E224C595.1, DA8612FE-B1E4-423B-8191-B746E224C595.2, DA8612FE-B1E4-423B-8191-B746E224C595.3, and DA8612FE-B1E4-423B-8191-B746E224C595.4.

For BOC table examples, as shown in the table below, the headings for the tables from left to right start with the "PC Tag (Base)" which represents the PC Instance Tag's base identifier per paragraph 5.4.4.1 and then the "Tag (Ext)" which represents the PC Instance Tag's optional Extension Identifiers per paragraph 5.4.7. Continuing, the "PC UUID (Base)" represents that PC UUID's instance identifier per paragraph 5.4.4.2, and then the "UUID (Ext)" represents the PC Instance UUID's optional extensions identifiers per paragraph 5.4.7. Finally, the "Description" is optional and defined by the user or obtained via queries from the model-based definition model that describes the product characteristic.

PC Tag (Base)	Tag (Ext)	PC UUID (Base)	UUID (Ext)	Description
PC001	N/A	6F5FE3F3-CEFF-4BE4-8B69-D4DDA1230143	N/A	Flatness 0.12 on Datum Feature A
PC002	N/A	6878BC7C-84B9-4D4D-8413-95ADC14D9156	N/A	Perpendicular w.r.t. A on Datum Feature B
PC003	N/A	6F63FE5A-A79F-42FD-B02A-BE35E502D829	N/A	Hole Size 88 +0.15 – 0.0 on Datum Feature B.
PC004	1	DA8612FE-B1E4-423B-8191-B746E224C595	1	Hole Size 16 +0.3 – 0.1 on Hole Feature 1
PC004	2	DA8612FE-B1E4-423B-8191-B746E224C595	2	Hole Size 16 +0.3 – 0.1 on Hole Feature 2
PC004	3	DA8612FE-B1E4-423B-8191-B746E224C595	3	Hole Size 16 +0.3 – 0.1 on Hole Feature 3
PC004	4	DA8612FE-B1E4-423B-8191-B746E224C595	4	Hole Size 16 +0.3 – 0.1 on Hole Feature 4
PC005	1	BCA322EA-A95C-4EEA-822B-98D061019619	1	Position 1.4 w.r.t. AB on Hole Feature 1
PC005	2	BCA322EA-A95C-4EEA-822B-98D061019619	2	Position 1.4 w.r.t. AB on Hole Feature 2
PC005	3	BCA322EA-A95C-4EEA-822B-98D061019619	3	Position 1.4 w.r.t. AB on Hole Feature 3
PC005	4	BCA322EA-A95C-4EEA-822B-98D061019619	4	Position 1.4 w.r.t. AB on Hole Feature 4

1.3.4 Product Characteristic (PC) Designation with Criticality Classification Example

The core for each model-based characteristic is its Product Characteristic designation. The PC Designation consists of a human-readable PC Reference Tag contained within an elongated hexagon symbol or textually between “less than” and “greater than” character. The PC Reference Tag is often followed by a criticality classification within an elongated chevron symbol or textually between two “greater-than” characters. The criticality classification designation contains a criticality level and/or category with an optional caveat. Figure 3 is an example of a PC Reference Tag <PC043> being augmented with a criticality classification >S>, both in symbolic form and textual form. Criticality Classifications are described in section 6.



<PC043>>S>

Figure 3: Example Product Characteristic (PC) Tag with Criticality Designator

1.3.5 Product Characteristic (PC) Designation with all Augmentations Example

The PC Reference Tag (PC Tag) shall be used as the core unique visual identifier and this standard extends the PC designation core with optional augmentations. Figure 4 uses PC Tag <PC041> with a product requirement association <REQ-ME-044< before the PC Tag, a criticality classification >CR: S.2> after the PC Tag, and a verification plan requirement /CMM: 100%/ at the end, both in symbolic form and textual form. The PC Reference Tag is described

in section 5. The Product Requirement Association augmentation is described in section 7. The Criticality Classifications are described in section 6. In addition, the Verification Plan Requirement is described in section 8. Note: there could be more than one augmentation of the same type.

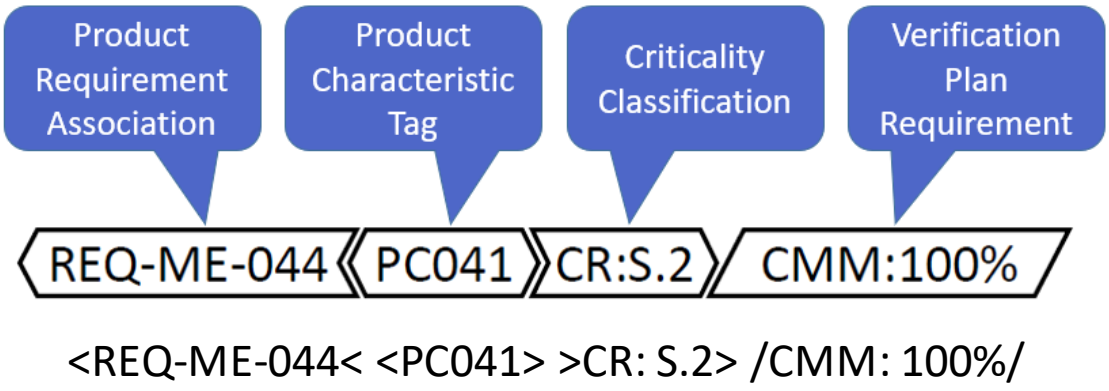


Figure 4: PC Reference Tag with All Augmentations

1.3.6 Product Characteristic (PC) and Augmentations Generation

PC Tags and their optional augmentations are typically created during the product realization process, specifically with the activities of product/design definition and process/operation definition.

PC Reference Tags should be created and assigned during product/design definition as it explicitly identifies annotations that require verification towards acceptance of a quality product. In addition, assigning PC Tags early in the product realization process allows the model-based enterprise to take advantage of PC Tags throughout the product realization process, including the enablement of precise quality feedback through digital PC identifiers, which enables the product's digital thread.

The assignment of a PC Tag's Criticality Classification augmentation allows the designer to convey important criticality information. Furthermore, the relationship of the Product Requirement Association to the PC Tag helps communicate the designer's purpose for the annotation.

Finally, the Verification Plan Requirement is tightly associated with the PC Tag and is seen as a process definition activity as it is typically performed by a quality process engineer after receipt of a design at a full completeness annotation state.

Therefore, the PC Tag with optional CC and/or PRA may be assigned during product/design definition activities, whereas a quality process engineer may embellish the authorized product definition with VPR information as part of their process definition activities. This process definition results in a new embellished derivative sourced from the authorized product definition.

1.4 Support within the Quality Information Framework (QIF)

The QIF standard defines an integrated set of information models, which enable the effective exchange of metrology data throughout the entire manufacturing quality measurement process – from product design to inspection planning to execution to analysis and reporting. The goal of the QIF specification is to facilitate interoperability of manufacturing quality data between system software components.

For reference, the ANSI/DMSC QIF v3.0 schema has data types for Characteristics, Characteristic Designator, Criticalities and Symbols. Fuller details for these QIF data types are shown in Appendix 10.2. Furthermore, related Terms and Definitions from QIF are shown in section 3.

1.5 Data Structure of the Model-Based Characteristics (MBC)

The machine-readable component of the MBC, which may not be visible to the user, will uniquely identify all the characteristic information in a machine-readable format.

Figure 5 shows a high-level view of the MBC information architecture. The architecture is documented as block diagrams according to the OMG Systems Modeling Language (SysML) standard. For a brief explanation of block diagrams using SysML, see informative Appendix 10.6.

The core of the MBC architecture is centered about a ProductCharacteristic. A Product Characteristic is a type of Characteristic. A ProductCharacteristic may have zero-to-many Augmentations. Augmentation types include CriticalityClassification, VerificationPlanRequirement, and ProductRequirementAssociation. The ProductCharacteristic and each of the Augmentations may be presented to the user via a subtype of a Symbol. Finally, the architecture references Symbol of different eSymbolShapeType value, and uses of UUID (universally unique identifiers) as a machine-readable identifier as a string value. The UUID should be generated per the ISO/IEC 9834-8 standard. Note that the block diagram names are typically a concatenation of the words in the “camel case” format that describe its function such as “ProductCharacteristic” is used for “product characteristic”.

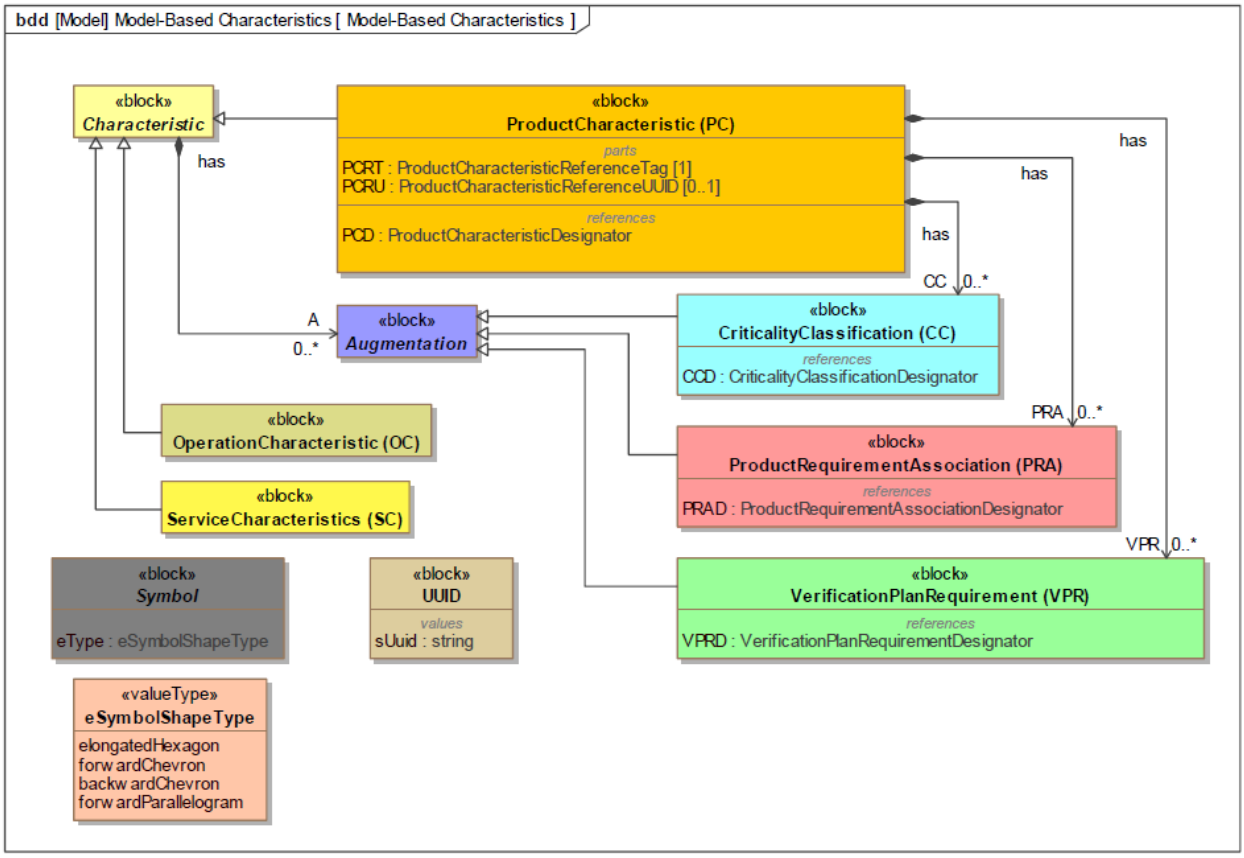


Figure 5: Block Diagram - Product Characteristic with Augmentations Information Model

Reference section 9 for the complete MBC logical data model. Review section 5 for details on Characteristics, section 6 for details on the CriticalityClassification augmentation, section 7 for details on the ProductRequirementAssociation augmentation, and section 8 for details on the VerificationPlanRequirement augmentation.

Furthermore, this standard defines the context of the designators and presents symbols for representing the ProductCharacteristic (PC) and its optional Augmentations. Finally, OperationCharacteristics (OC) and ServiceCharacteristics (SC) are listed for completeness and future development.

2 Normative References

- ANSI/DMSC QIF v3.0 -2018; Automation systems and integration — Quality information framework (QIF) - An integrated model for manufacturing quality information.
- ASME Y14.5-2018; *Dimensioning and Tolerancing - Engineering Drawing and Related Documentation Practices*
- ASME Y14.36; Surface Texture Symbols
- ASME Y14.41-2019; *Digital Product Definition Data Practices*
- ASME Y14.45-2021; *Measurement Data Reporting*
- ASME Y14.47-2019; *Model Organization Practices*
- ISO/IEC 9834-8 – 2014; Information technology — Procedures for the operation of object identifier registration authorities — Part 8: Generation of universally unique identifiers (UUIDs) and their use in object identifiers
- ISO 23952:2020; Automation systems and integration — Quality information framework (QIF) - An integrated model for manufacturing quality information.
- ISO 3534-2: 2006; Statistics — Vocabulary and symbols — Part 2: Applied statistics
- OMG Systems Modeling Language (OMG SysML™) Version 1.7 – August 2022

3 Terms and Definitions

For the purpose of this standard, the following terms and definitions are applicable.
For abbreviations and their definitions, see informative Appendix: Annex E – Acronyms & Abbreviations.

3.1 Annotation

an item of text or symbology conveying visible, non-shape information contained within a 3D model. (MBC v1.0)

3.2 Attribute

an item of non-visible information contained in, or related to, a model that is available upon interrogation of the model. (MBC v1.0)

3.3 Bill of Characteristics (BoC)

a list of all the characteristics applied to a product. (ANSI/DMSC QIF v3.0)

3.4 Characteristic

a trait, quality, or attribute on an element of a feature such as its size, location, form, or property, which may be a specification limit, a dimension with tolerance, a geometric tolerance within a feature control frame, or some other numerical or non-numerical control. (MBC v1.0)

3.5 Characteristic Augmentation

an item of specialized supplemental information associated with a characteristic such as Criticality Classification, Product Requirement Associations, or Verification Plan Requirements. (MBC v1.0)

3.6 Characteristic, Critical – Critical Characteristic

a Product Characteristic, which is supplemented by a criticality classification augmentation. (MBC v1.0)

3.7 Characteristic, Drawing-Based (DBC) – Drawing-Based Characteristic

a characteristic, which is presented on a graphical drawing. (MBC v1.0)

3.8 Characteristic, Driven – Driven Characteristic

a Product Characteristic, which is supplemented by a Product Requirement Association augmentation. (MBC v1.0)

3.9 Characteristic, Model-Based (MBC) – Model-Based Characteristic

a characteristic, which is represented within a model-based definition model. (MBC v1.0)

3.10 Characteristic, Operation (OC) – Operation Characteristic

a characteristic, which is created to identify an operation or process requirement for the product realization process of a product or of a feature of a product. (MBC v1.0)

3.11 Characteristic, Product (PC) – Product Characteristic

a characteristic, which is created to identify a verification requirement, such as a tolerance or specification, applied to a feature or of a product. (MBC v1.0)

3.12 Characteristic, Service (SC) – Service Characteristic

a characteristic, which is created to identify a service or maintenance requirement for the sustainment of a product or of a feature of a product. (MBC v1.0)

3.13 Characteristic, Verification – Verification Characteristic

a Product Characteristic, which is supplemented by a Verification Plan Requirement augmentation. (MBC v1.0)

3.14 Criticality Classification (CC)

a characteristic augmentation that optionally designates a criticality. (MBC v1.0)

3.15 Derivative, Embellished – Embellished Derivative

the condition of a derivative being embellished from its source with respect to an intended use. (MBC v1.0)

3.16 Derivative, Equivalent – Equivalent Derivative

the condition of a derivative being functionally equivalent to its source with respect to an intended use. (MBC v1.0)

3.17 Designator

a means to identify and set apart for a specific purpose. (MBC v1.0)

3.18 Designator, Characteristic – Characteristic Designator

a designator that labels a characteristic with its tag. (MBC v1.0)

3.19 Designator, Characteristic Augmentation – Characteristic Augmentation Designator

a designator that labels a characteristic augmentation from its attributes. (MBC v1.0)

3.20 Designator, Criticality Classification - Criticality Classification Designator

a designator that labels a Criticality Classification. (MBC v1.0)

3.21 Designator, Product Requirement Association - Product Requirement Association Designator

a designator for the Product Requirement Association. (MBC v1.0)

3.22 Designator, Verification Plan Requirement - Verification Plan Requirement Designator

a designator for the Verification Plan Requirement. (MBC v1.0)

3.23 Definition, Product – Product Definition

any information that defines the physical or functional characteristics of a product, to ensure design intent. (MBC v1.0)

3.24 Definition, Process – Process Definition

any information that defines a process that contributes to realizing (e.g., fabricating, accepting) a product. (MBC v1.0)

3.25 Identifier, Product Characteristic Base - Product Characteristic Base Identifier

an identifier, which is used as the basis for both a Product Characteristic Reference Identifier (i.e., Tag or UUID), and a Product Characteristic Instance Identifier. (MBC v1.0)

3.26 Identifier, Product Characteristic Extension - Product Characteristic Extension Identifier

an optional identifier that is concatenated with a Product Characteristic Base Identifier, which uniquely identifies a Product Characteristic Instance Identifier. (MBC v1.0)

3.27 Identifier, Product Characteristic Instance - Product Characteristic Instance Identifier

an identifier (i.e., Tag or UUID), of an instance of a Product Characteristic Reference Identifier, which combines a Product Characteristic Base Identifier with an optional Product Characteristic Extension Identifier. (MBC v1.0)

3.28 Identifier, Product Characteristic Reference - Product Characteristic Reference Identifier

an identifier (i.e., Tag or UUID), which references a verification requirement annotation as a Product Characteristic, which contains the Product Characteristic Base identifier. (MBC v1.0)

3.29 Model

a representation of a thing. (MBC v1.0)

3.30 Model, 3D – 3D Model

a digital model that is 3-dimensional. (MBC v1.0)

3.31 Model, 3D CAD – 3D CAD Model

a 3D model of a physical item that is generated using drafting and design software. (MBC v1.0)

3.32 Model Digital – Digital Model

a model that is digitally represented. (MBC v1.0)

3.33 Model, Model-Based Definition (MBD) – Model-Based Definition

a 3D model that requires semantic PMI and structured attributes. (MBC v1.0)

3.34 Numeric

an unsigned non-zero integer with optional leading zeroes

3.35 Product and Manufacturing Information (PMI)

the information, often presented as annotations, in a 3D model that support various lifecycle activities. (MBC v1.0)

3.36 Product Requirement Association (PRA)

a characteristic augmentation that optionally relates a product requirement to a characteristic. (MBC v1.0)

3.37 Quality Information Framework (QIF)

a standard, which supports digital thread concepts in engineering applications ranging from product design through manufacturing to quality inspection. (ANSI/DMSC QIF v3.0)

3.38 QIF Persistent Identifier (QPId)

an identifier used in QIF as a standard universally unique identifier (UUID) (ANSI/DMSC QIF v3.0)

3.39 Requirement, Customer – Customer Requirement

a requirement that addresses a customer's expectations for the functionality, performance, and quality of a product being developed. (MBC v1.0)

3.40 Requirement, Product – Product Requirement

a requirement that addresses a product's purpose, its features, functionalities, and behavior. (MBC v1.0)

3.41 Requirement, Verification – Verification Requirement

a requirement (e.g., tolerance, specification) applied to a part feature or product, which requires verification to assure product acceptance. (MBC v1.0)

3.42 Tag

a digital element identifier used as a human-readable identifier, unique within a local context. (MBC v1.0)

3.43 Tag, Characteristic – Characteristic Tag

a tag used for a characteristic. (MBC v1.0)

3.44 Tag, Feature Reference – Feature Reference Tag

a tag used to identify a geometric feature, face, or element. (MBC v1.0)

3.45 Tag, General – General Tag

a characteristic tag used for a non-product characteristic or general usage. (MBC v1.0)

3.46 Tag, Product Characteristic (PC) – Product Characteristic Tag

a characteristic tag used for a product characteristic. (MBC v1.0)

3.47 Tag, Product Characteristic Instance (PC Instance Tag)

a product characteristic tag used as an instance identifier. (MBC v1.0)

3.48 Tag, Product Characteristic Reference (PC Reference Tag)

a product characteristic tag used as a reference identifier. (MBC v1.0)

3.49 Universally Unique Identifier (UUID)

a digital element identifier used as a persistent machine-readable universally unique identifier. The UUID should be generated according to the ISO/IEC 9834-8 standard. (MBC v1.0)

3.50 Verification

the act or process of establishing the truth, accuracy, or reality of something (Merriam Webster)

3.51 Verification Plan Requirement (VPR)

a characteristic augmentation that optionally relates a verification plan requirement to the characteristic (MBC v1.0)

4 Characteristics

A characteristic may have multiple subtypes that are based upon its medium type (e.g., digital models vs electronic drawings), product-lifecycle discipline type (e.g., product definition, process definition, execution, service definition, and end-of-life) and/or by characteristic augmentation type (e.g., has a criticality classification, has a product requirement association, or has a verification plan requirement). Furthermore, the referencing of characteristic types may be cumulative; therefore, a characteristic may be a critical product characteristic and also a model-based product characteristic.

4.1 Types

4.1.1 Digital, Model-Based vs Electronic, Document-Based

Currently, a product definition can be communicated as a digital model-based definition model or as a set of more traditional 2-dimension drawings and/or specification-based product definition documents.

4.1.1.1 *Model-Based Characteristic (MBC)*

A characteristic that is represented in a model-based definition model. The characteristic shall have a characteristic reference identifier that is semantically connected to an annotation within the context of a 3D model. The characteristic reference identifier should have data structure attributes that both visually communicate a human-readable characteristic Reference Tag and digitally communicate a machine-readable characteristic base UUID. Furthermore, the characteristic reference identifier will contain a characteristic Base Identifier and may have one or more characteristic augmentations.

4.1.1.2 *Drawing-Based Characteristic (DBC)*

A characteristic that is presented on a graphical drawing (either paper or electronic document). The characteristic may have a human-readable Reference Tag within a graphical symbol shown in close proximity to a textual reference or drawing annotation. The characteristic Reference Tag will contain a characteristic Reference Tag and may have one or more characteristic augmentations.

4.1.2 Lifecycle Discipline

Some suggest that a product may transition through seven lifecycle phases. Those phases may be:

1) a product requirement phase, which identifies the **needs** of the product. This leads to
2) a product definition phase, which defines **what** product is to be made. Then
3) a process/operations definition phase, which defines **how** to make/accept a product. Next,
4) an execution phase, which **realizes** the product accordingly with
5) a final acceptance phase, which **sells** the product. Then
6) a service/maintenance definition phase, which **sustains** the product. Finally,
7) the dismantlement/ retirement phase, which **ends** the life of the product.

This leads to the identification of specific characteristic types or augmentations based upon when and how this information may be communicated within these phases. The following are some lifecycle phase specific characteristics.

4.1.2.1 *Product Characteristic (PC)*

A Product Characteristic is a characteristic initiated during the product definition activity. Typically, this is a verification requirement, such as tolerance or specification applied to a feature or product, which requires some method of verification. This verification plan shall be satisfied by a specific verification method with optional sampling plan. A Product Characteristic shall have a human-readable locally unique tag within a part context and should have a machine-readable universally unique identifier. A Product Characteristic may also be augmented with additional information. Characteristic augmentation information could be contained in the original product definition source or embellished later within the product lifecycle, resulting in an embellished derivative of the product definition.

4.1.2.2 *Operation Characteristic (OC) (future scope)*

An operation characteristic (aka process characteristic) is an operation or process requirement identified during the product realization process (e.g., manufacturing, testing) activity of a product or of a feature of a product. Typically, operation characteristics are specified in a supplemental product or process specification. An example of an operation characteristic is "The process specification requires the product to have an oven curing operation at 400°F for 30 minutes". The operation characteristic could be embellishments added to a product definition for when the part is being processed for manufacturing, inspection, and/or dismantlement activities, which results in an embellished derivative from the source product definition.

Informative Note: The contributors of this specification had originally identified this as a Process Characteristic, but its abbreviation "PC" would contribute to confusion with the Product Characteristic's abbreviation of "PC". Therefore, it was decided to consider this as a process Operation Characteristic that will allow for a unique abbreviation of "OC".

4.1.2.3 *Service Characteristic (SC) (future scope)*

A service characteristic is a service or maintenance requirement identified for the sustainment of a product or of a feature of the product. Typically, service characteristics are specified in a supplemental service manual. An example of a service characteristic is "This AMC 343 engine requires an oil change every 7,500 miles or every six months". The service characteristics could be embellishments added to a product definition for when the part is placed into service, this results in an embellished derivative from the source product definition.

4.1.3 Augmentation Based Characteristics

A Product Characteristic may be a stand-alone identifier or may have one or more augmentations. The following are types of characteristics based upon their augmentation.

4.1.3.1 *Critical Characteristic*

A product characteristic that contains a Criticality Classification augmentation

4.1.3.2 *Driven Characteristic*

A product characteristic that has been driven by a Product Requirement Association augmentation.

4.1.3.3 *Verification Characteristic*

A product characteristic that is supplemented by a Verification Plan Requirement augmentation.

4.2 System Requirements

This section establishes requirements for the model-based product characteristics application system's attributes, associativity, placement, display management, query, and behaviors of product characteristics data.

4.2.1 Attributes

Attributes are used to capture additional information for a characteristic or augmentation that is not shown as annotation on or along with the product characteristic's annotation. Attributes shall be available on demand. Attributes may be presented using text description, forms, or other techniques.

4.2.2 Associativity

The system shall associate each PC Reference Tag with its primary associative annotation, Product Characteristic augmentation designator(s), and its PC Reference UUID.

4.2.2.1 *PC Reference Tag to Annotation Group*

The system shall associate a PC Reference Tag with its primary associative annotation(s).

4.2.2.2 *Product Characteristic Augmentation to PC Reference Tag*

The system shall associate the Product Characteristic augmentation designator(s) of a characteristic with its associative PC Reference Tag.

4.2.2.3 *PC Reference Tag to PC Reference UUID*

The system shall have a bi-directional associativity between a PC Reference Tag and its PC Reference UUID.

4.2.2.4 *Persistence of PC Reference UUID with PC Reference Tag*

The system shall maintain the persistence of the PC Reference UUID with the Product Reference Tag.

4.2.3 Placement, Alignment & Annotation Planes

The application system shall place the human-readable PC Reference Tag in relation to its primary associative annotation. Furthermore, any optional augmentation(s) shall be visibly positioned near its PC Reference Tag and shall be on the same plane as its primary associative annotation.

4.2.3.1 *PC Reference Tag Placement*

The PC Reference Tag and its augmentations should be placed per the following:

- The PC Reference Tag and any PC augmentation(s) shall not overlap another annotation within its saved state view.
- The PC Reference Tag and any PC augmentation(s) shall not overlap the part within its saved state view.

4.2.3.2 *Manipulation of PC Reference Tag*

The manipulation of the annotation shall move the PC Reference Tag, any PC augmentations, and the annotation collectively as a group.

4.2.3.3 *Placement of PC Reference Tag*

The human-readable PC Reference Tag designator:

- Shall be visibly placed near its associative annotation(s).
- Shall be aligned (e.g., top, bottom, before, after, or corner) with its associative annotation(s).
- Should be in line (i.e., before or after) with its associative annotation(s).
- Shall be on the same annotation plane as its primary associative annotation.

4.2.3.4 *Placement of PC Augmentation Designator*

The symbolic placement of the human-readable PC augmentation designator:

- Shall be visibly placed near its PC Reference Tag.
- Should be aligned with its PC Reference Tag.
- Shall be on the same annotation plane as its PC Reference Tag.

4.2.3.5 *PC Instance Tag Placement*

The display of PC Instance Tags may need to overlap another annotation or the part.

4.2.4 Display Management

Display management shall include the ability to enable or disable the display (show/hide) of all PC Reference Tags, augmentation designators by type, or other identifiers.

4.2.4.1 *Toggle PC Reference Tag*

The system shall have the ability to toggle (show/hide) the display of the active PC Reference Tag on the model. This display shall work in conjunction with the annotation display of the associated entity and shall allow display only when the associated annotation is visible.

4.2.4.2 *Toggle PC Augmentation Designator*

Displaying of augmentations shall be grouped and controlled separately. The system should have the ability to toggle (show/hide) the display of an active PC augmentation(s) such as CriticalityClassification, ProductRequirementAssociation, and VerificationPlanRequirement.

4.2.4.2.1 Toggle Product Characteristic Designator CC Augmentation

The system should have the ability to toggle (show/hide) the display of an active PC Reference Tag's CriticalityClassification augmentation.

4.2.4.2.2 Toggle Product Characteristic Designator PRA Augmentation

The system should have the ability to toggle (show/hide) the display of an active PC Reference Tag's ProductRequirementAssociation augmentation.

4.2.4.2.3 Toggle Product Characteristic Designator VPR Augmentation

The system should have the ability to toggle (show/hide) the display of an active PC Reference Tag's VerificationPlanRequirement augmentation.

4.2.4.3 *Toggle All PC Reference Tag's PC Instance Tags*

The system should have the ability to toggle (show/hide) the display of all PC Instance Tags of an active PC Reference Tag.

4.2.4.4 *Toggle Individual PC Reference Tag's PC Instance Tag*

The system should have the ability to toggle (show/hide) the display of individual Instance Tags of an active Product Characteristics Designator.

4.2.4.5 *Toggle Individual Product Characteristic Reference UUID*

The system shall have the ability to toggle (show/hide) the display of an active PC Reference UUID.

4.2.5 Query

The ability to query the Product Characteristic information shall be available.

4.2.5.1 *Graphical Display of Associated Annotations*

The system shall have the ability to graphically display the PC Reference Tag's associated annotation(s). The associated annotations for a PC Reference Tag shall be highlighted, or otherwise distinguished from other entities on the display, on demand.

4.2.5.2 *Query of Last Reference Tag*

The system shall have the ability to determine and report the numerically last active PC Reference Tag within the product's bill of characteristics, on demand.

4.2.5.3 *Query for Missing Reference Tags*

The system shall have the ability to determine and report of any deleted active PC Reference Tag from a numeric range (e.g., first to last) PC Reference Tags within the product's bill of characteristics, on demand.

4.2.5.4 *Query for Individual PC Reference UUID*

The system shall have the ability to report the UUID of the product characteristic, on demand.

4.2.6 Behaviors

The system shall have the ability to assign, check, lock, modify, and persist PC Reference Tags or PC Reference UUIDs. However, the string of characters making up a given UUID may not be changed. Any new UUIDs shall be created by a competent UUID generator.

4.2.6.1 *Assign PC Reference UUID to PC Reference Tag*

The system shall have the ability to assign a PC Reference UUID to a PC Reference Tag.

4.2.6.2 *Check for Uniqueness of all PC Reference Tags*

The system shall have the ability to check all active PC Reference Tags for uniqueness within the product's bill of characteristics. That is, there are no duplicate identifiers.

4.2.6.3 *Check for Uniqueness of all PC Reference UUIDs*

The system shall have the ability to check all active PC Reference UUIDs for uniqueness within the product's bill of characteristics. That is, there are no duplicate identifiers.

4.2.6.4 *Lock of PC Reference Tags at a Revision (Release)*

The system shall have the ability to lock all active PC Reference Tags from modification, typically after a revision promotion.

4.2.6.5 *Lock of PC Reference UUIDs at a Revision (Release)*

The system shall have the ability to lock all active PC Reference UUIDs from modification, typically after a revision promotion.

4.2.6.6 *Modify Individual PC Reference Tag*

The system shall have a controlled ability to modify a PC Reference Tag within the product's bill of characteristics for the purpose of correctness or to assure local uniqueness.

4.2.6.7 *Modify Individual PC Reference UUID*

The system shall have a controlled ability to modify a Reference UUID within the product's bill of characteristics for the purpose of correctness or to assure uniqueness.

4.2.6.8 *Persistence of PC Reference Tags after Release*

The system shall assure the persistence of a PC Reference Tag even after removal. Once removed, the PC Reference Tag is attributed as deleted/removed and may only be used if that PC Reference Tag is resurrected as an equivalent product characteristic.

4.2.6.9 *Persistence of PC Reference UUIDs after Release*

The system shall assure the persistence of a PC Reference UUID even after removal. Once removed, the PC Reference UUID is attributed as deleted/removed and may only be used if that PC Reference UUID is resurrected as an equivalent product characteristic.

4.2.7 Designators

Within the context of this standard, designators are used to set apart and label product characteristics and augmentations. The space " " character may be used within the designator to aid in human-readability, and therefore, spaces, unless within a quoted string "*string*", have no contextual meaning for parsing or interpretation. A space cannot interrupt an explicitly defined tag (e.g., PC007) or standard enumeration value (e.g., CMM, CR). For example, the following designators are interpreted as the same and shall be parsed as the same.

<PC007>>CR:S>/CMM:100%/

<PC007> >CR: S> / CMM: 100% /

Some designators allow for a free-from string, and any free-form string requires beginning and ending double quotes. Furthermore, any Unicode character, except for another double quote, may be contained within the free-form string between double quotes.

"free-form string!"

4.3 Augmentations

An augmentation is an item of specialized supplemental information associated with a characteristic. A characteristic may have one-to-many augmentations.

4.3.1 Criticality Classifications (CC)

A criticality classification is an augmentation applied to a product characteristic. Often when a Product Characteristic has been augmented by one or more criticality classifications, it is referenced as a Critical Product Characteristic.

4.3.2 Product Requirement Associations (PRA)

A product requirement association is an augmentation applied to a product characteristic. Often when a Product Characteristic has been augmented by one or more product requirement associations, it is referenced as a Driven Product Characteristic, as the Product Characteristic has been driven into existence by a product requirement association.

4.3.3 Verification Plan Requirement (VPR)

A verification plan requirement is an augmentation applied to a product characteristic. Often when a Product Characteristic has been augmented by a verification requirement, it is referenced as a Verification Characteristic.

4.3.4 Augmentations Relative Order

The symbolic placement of various augmentations shall be in-line and shall be consistently ordered with respect to the Product Characteristic Designator Symbol. Augmentation Designator Symbols are categorized as a Pre-Augmentation Symbol or as a Post-Augmentation Symbol. Pre-Augmentation symbols shall be placed in-line to the left of the PC Reference Tag symbol, whereas the post-Augmentation symbols shall be placed in-line to the right of the PC Reference Tag symbol.

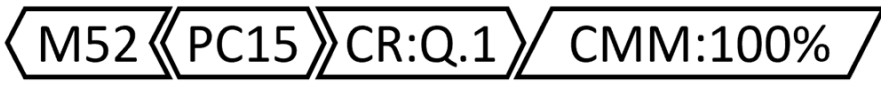
Pre-Augmentations are Product Requirement Associations whereas the post-Augmentations are the Criticality Classifications and then the Verification Plan Requirement.

The order of optional augmentation with respect to the Product Characteristic designator follows the format shown below:

<*PRA*< <PC Tag> >CC> /VPR/

An example of multiple optional augmentations within the context of a Product Characteristic designator is shown below, both textually and symbolically. This example reads that a Product Characteristic Designator of <PC15> has a Criticality Classification Designator of >CR:Q.1>, which designates a Criticality Level of “CR” for “Critical”, a Criticality Category of “Q” for “Quality”, and a Criticality Caveat of “1” that has some organizational meaning to the criticality’s usage. Next, the Product Characteristic has a forward-looking Product Requirement Association to a requirement designated as <M52<, which designates a Product Requirement with an identification of “M52”. Meaning, that Product Requirement “M52” is being at least partially satisfied by this associated Product Characteristic. Finally, a Verification Plan Requirement was associated with the Product Characteristic and was designated as

/CMM: 100%/. Based upon this designation, the Product Characteristic should be verified by a "CMM" Verification Method with a Sampling Plan of "100% sampling".



<M52<<PC15>>CR:Q.1>/CMM:100%/

5 Product Characteristics

A Product Characteristic identifies a verification requirement that typically applies a tolerance or specification to a feature or product.

5.1 Identification

The identification of a Product Characteristic has a framework of identifications shown in Figure 6. This framework addresses both a human-readable context and a machine-readable context.

5.1.1 Product Characteristic Identification Framework

The information model, shown in Figure 6, for a Product Characteristic is based upon the four-identifier element types.

- ProductCharacteristicBaseIdentifier
- ProductCharacteristicReferenceIdentifier
- ProductCharacteristicInstanceIdentifier
- ProductCharacteristicExtensionIdentifier

The ProductCharacteristicBaseIdentifier, ProductCharacteristicReferenceIdentifier, and ProductCharacteristicInstanceIdentifier are abstract classes, which each have two subtype identifiers that support a human-readable tag, and a machine-readable universally unique identifier (UUID).

The subclasses derived from the ProductCharacteristicBaseIdentifier are the ProductCharacteristicBaseTag and the ProductCharacteristicBaseUUID. Likewise, the subclasses derived from the ProductCharacteristicReferenceIdentifier are the PC Reference Tag and the ProductCharacteristicReferenceUUID. Moreover, the subclasses derived from the ProductCharacteristicInstanceIdentifier are the ProductCharacteristicInstanceTag and the ProductCharacteristicInstanceUUID.

The PC Reference Tag is composed from a ProductCharacteristicBaseTag and the ProductCharacteristicReferenceUUID is composed from a ProductCharacteristicBaseUUID.

The PC Reference Tag has one-to-many ProductCharacteristicInstanceTag; likewise, the ProductCharacteristicReferenceUUID has one-to-many ProductCharacteristicInstanceUUID.

The ProductCharacteristicInstanceTag is made up from a ProductCharacteristicBaseTag and optionally a ProductCharacteristicExtensionIdentifier; likewise, the ProductCharacteristicInstanceUUID is made up from a ProductCharacteristicBaseUUID and optionally a ProductCharacteristicExtensionIdentifier.

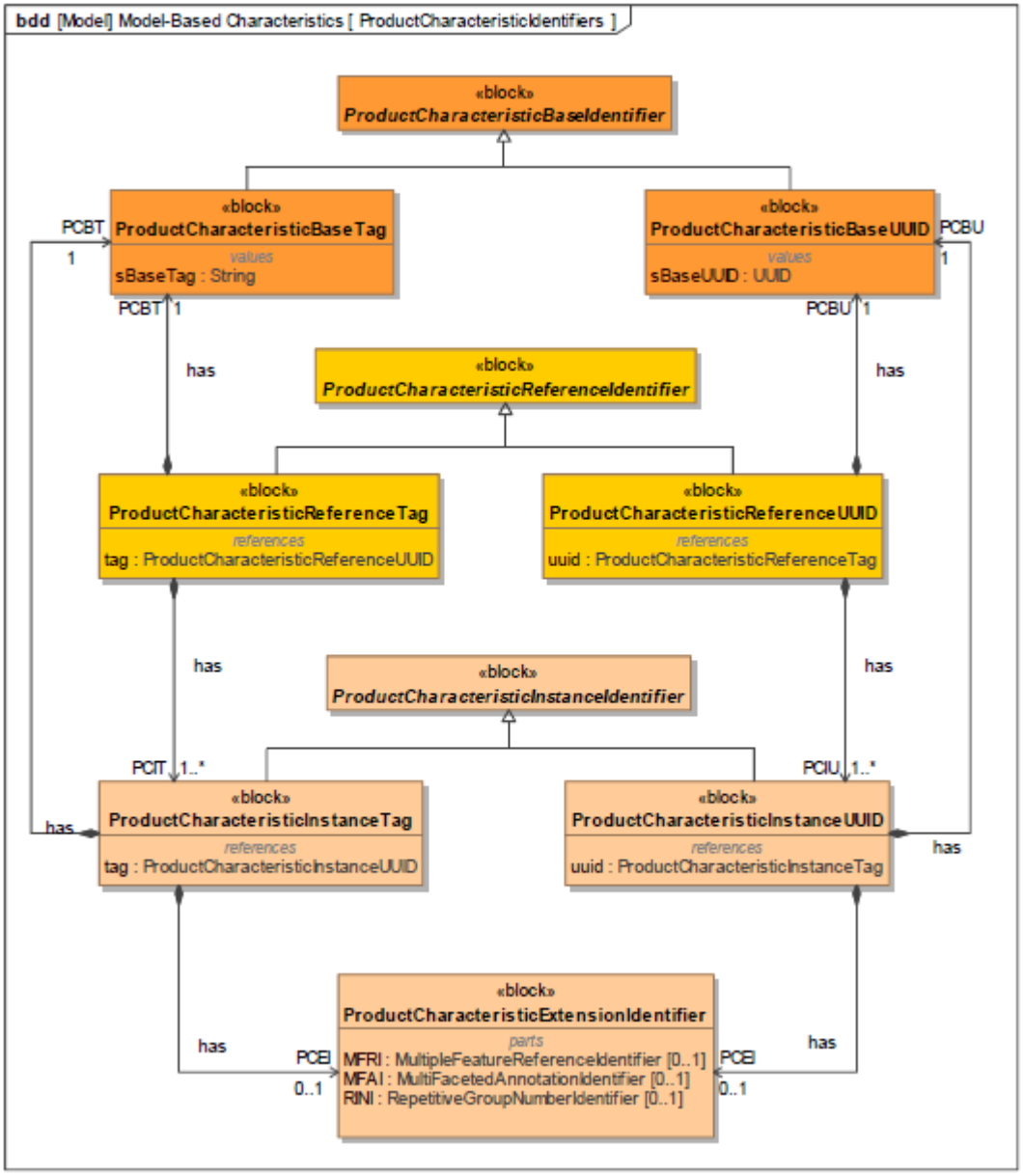


Figure 6: Block Diagram - Product Characteristic Identification Framework

The ProductCharacteristicExtensionIdentifier data model is shown in Figure 7. A ProductCharacteristicExtensionIdentifier uniquely extends the identification of a ProductCharacteristicInstanceIdentifier for use-cases that involve three different zero-or-one Extension Identifiers. See section 5.4.7 for format and examples. Extension Identifiers include:

- MultiFacetedAnnotationIdentifier (MFAI)
- MultipleFeatureReferenceIdentifier (MFRI)
- RepetitiveGroupNumberIdentifier (RGNI)

The Extension Identifiers concatenate with the PC Base Identifier to assure that the PC Instance Identifier uniquely identifies the instance of the PC.

A MultipleFeatureReferenceIdentifier is an abstract class for three different ways to identify features:

- FeatureReferenceTag,
- FeatureReferenceUUID, or
- FeatureReferenceSequenceNumber.

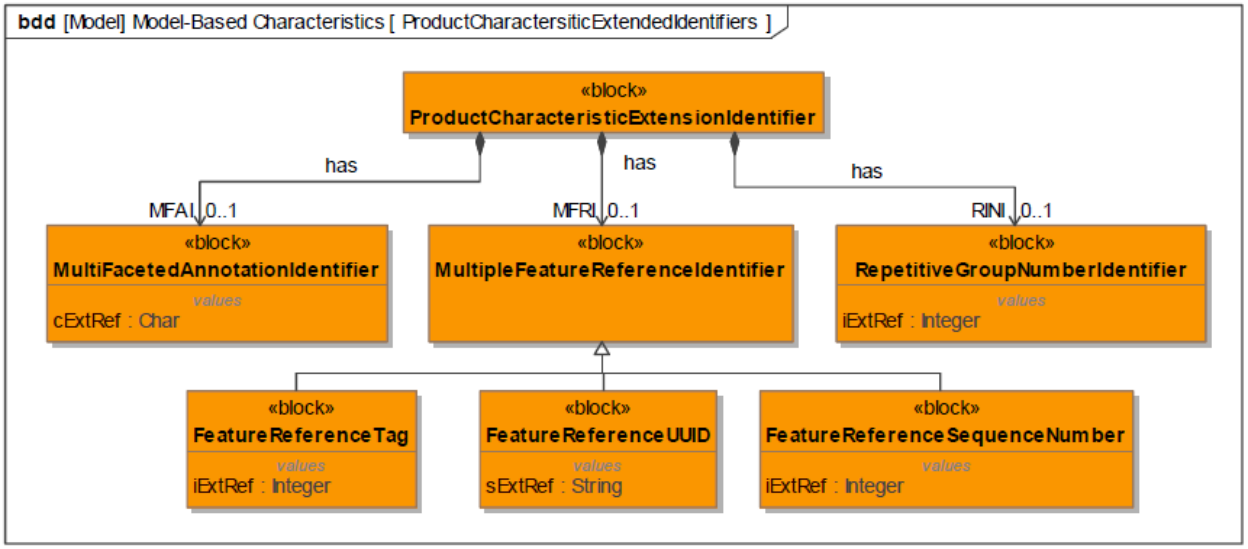


Figure 7: Block Diagram - Product Characteristic Extension Identifiers

Next, each occurrence of a Product Characteristic Reference will be applied to one or more Product Characteristic Instance. Therefore, each Product Characteristic occurrence will have a Product Characteristic Reference Identifier and then each occurrence will have multiple Product Characteristic Instance Identifiers. As a result, both the Reference Identifier and the Instance Identifier shall use the same common Base Identifier whereas the Instance Identifier will be concatenated with Product Characteristics Instance Extension Identifiers. This framework of identifiers is discussed further in this section.

Finally, the human-readable Product Characteristic Designator is used for displaying. It is built from the PC Reference Tag and may be shown within a ProductCharacteristicDesignatorSymbol.

5.2 Data Structure

The core information model of a ProductCharacteristic is shown in Figure 8. A ProductCharacteristic is a type of Characteristic. Each ProductCharacteristic occurrence shall have a human-readable ProductCharacteristicReferenceTag and have an optional machine-readable ProductCharacteristicReferenceUUID. Then a ProductCharacteristicReferenceTag shall have one-to-many ProductCharacteristicInstanceTag, likewise, a ProductCharacteristicReferenceUUID may have one-to-many ProductCharacteristicInstanceUUID. Additional data objects and their relationships are described more completely in subsequent sections.

Figure 5 shows how each ProductCharacteristic may have zero-to-many augmentations such as criticality classifications, verification plan requirements, and/or product requirement associations. Each augmentation is described in subsequent sections of this standard.

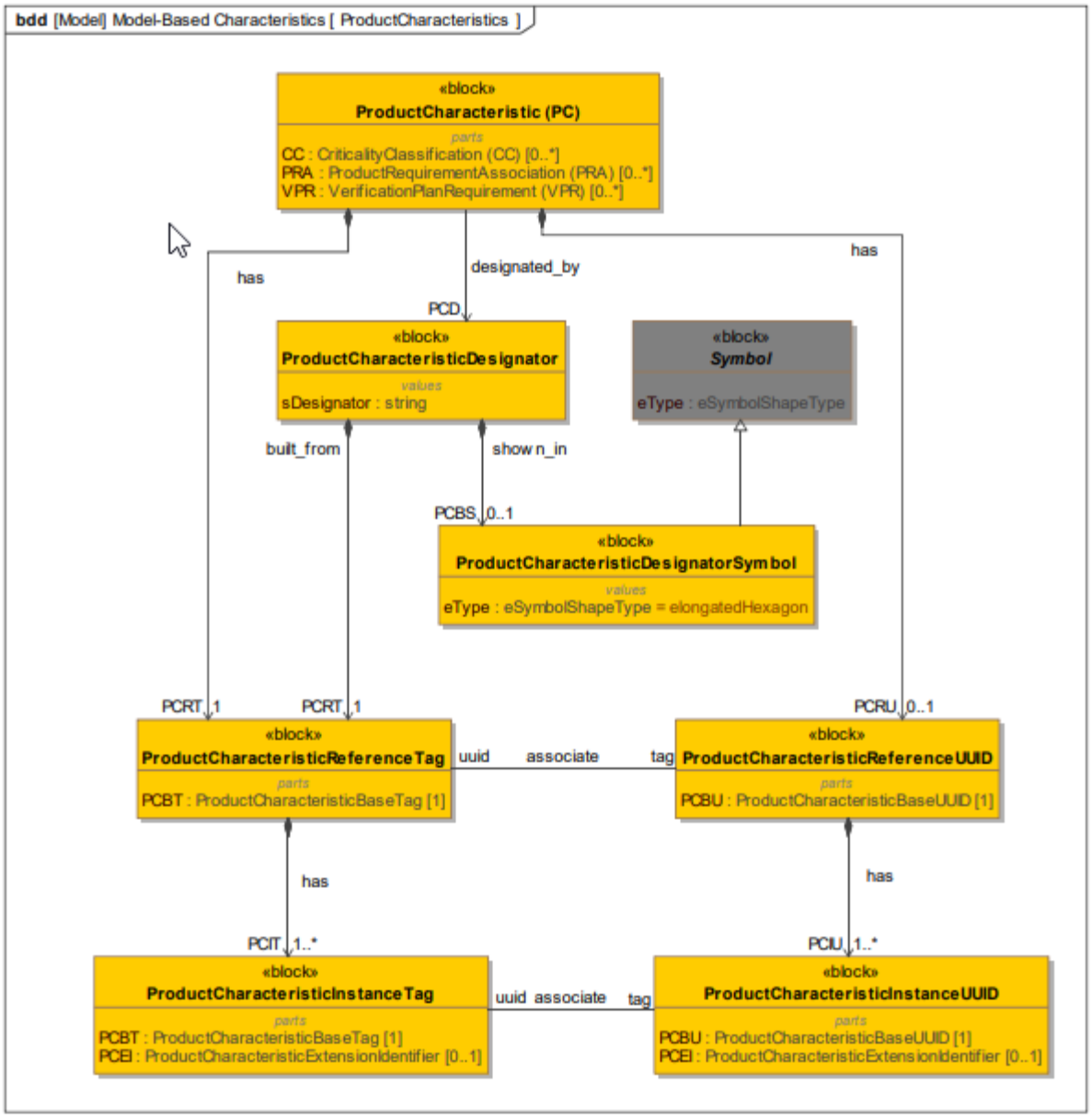


Figure 8: Block Diagram - Product Characteristic Data Model

5.3 Data Taxonomy

5.3.1 Product Characteristic Identification Framework Taxonomy

The Product Characteristic taxonomy consists of an identification framework, shown in Figure 6, and identifies both a Reference Identifier and an Instance Identifier for both Tags and UUIDs. The Reference Identifier and Instance Identifier are both built from a Base Identifier. Furthermore, as shown in Figure 8, the Instance Identifier is extended by the Extension

Identifier for use-cases involving multi-faceted annotations, multiple feature references, and repetitive groups.

5.3.2 Product Characteristic Data Taxonomy

Per Figure 8, the ProductCharacteristic taxonomy consists of a designator that is built from the Reference Tag and shown in a symbol and has a Reference UUID.

5.3.3 Product Characteristic with Augmentations

Again, as shown in Figure 5, a ProductCharacteristic may have one-to-many augmentations (e.g., CriticalityClassification, Product Requirement Association, and Verification Plan Requirement). Although infrequent, a ProductCharacteristic may be augmented by more than one augmentation of the same type.

<REQ-ME-4< <PC007> >CR:S.4>

5.4 Data Objects

5.4.1 ProductCharacteristic

The ProductCharacteristic is a type of Characteristic. Each ProductCharacteristic occurrence shall have a human-readable PC Reference Tag and have an optional machine-readable ProductCharacteristicReferenceUUID. A ProductCharacteristic may have zero-to-many augmentations such as CriticalityClassification, VerificationPlanRequirement, and/or ProductRequirementAssociation.

5.4.2 ProductCharacteristicDesignator

The ProductCharacteristicDesignator is human-readable and is built from the ProductCharacteristicReferenceTag. This relationship is shown in Figure 8.

The ProductCharacteristicDesignator may be shown within a graphical symbol and/or in a textual form. The PC Reference Tag always contains the PC Base Tag that shall be of any numeric form with optional prefix (e.g., PC). Below shows two examples of a PC Reference Tag. One is PC42 and the other is 007. Each is shown in both its symbolic format and its textual format.

PC Tag	With PC Prefix	Without PC Prefix
Within Graphical Symbol	PC42	007
Within Textual Format	<PC42>	<007>

Table 1: PC Reference Tag Designator Presentation Examples

Figure 9 shows the four approaches of representing PC Reference Tags, that is:

- Top-Left Cell: PC Tag with PC prefix, designated within a graphical symbol (i.e., elongated hexagon)
- Top Right Cell: PC Tag without PC prefix, designated within a graphical symbol (i.e., elongated hexagon)

- Bottom Left Cell: PT Tag with PC prefix, designated within a textual format (i.e., between ASCII "<" and ">" characters)
- Bottom-Right Cell: PT Tag without PC prefix, designated within a textual format (i.e., between ASCII "<" and ">" characters)

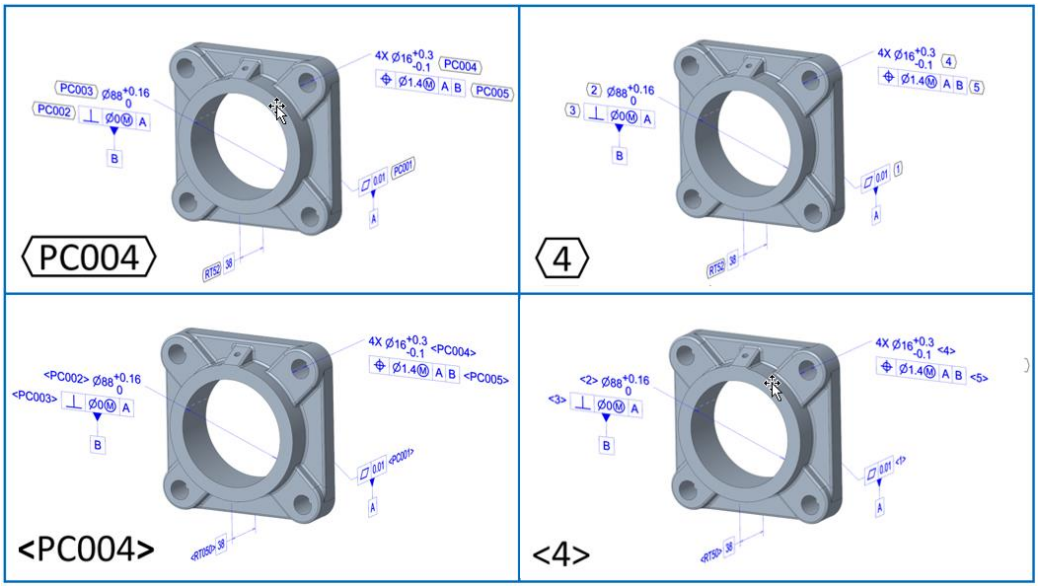
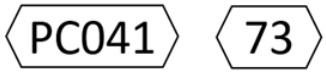


Figure 9: PC Reference Tag Format Examples on Single Annotations

5.4.3 ProductCharacteristicDesignatorSymbol

The optional ProductCharacteristicDesignatorSymbol is a Symbol element that outlines a human-readable symbol for housing the PC Reference Tag, which is often used as a product characteristic designator balloon presented within a printed drawing, electronic drawing, or digital model. According to Appendix 10.3, the preferred symbol for housing a PC Reference Tag should be an elongated hexagon symbol shape. The elongated hexagon symbol shall accommodate the PC Reference Tag and provide a visual connection to pre and post augmentation symbols. Below are two tag symbols, one using the PC prefix with numeric and the other using only a numeric.



5.4.4 ProductCharacteristicBaseIdentifier

The ProductCharacteristicBaseIdentifier is the basis identifier for both the ProductCharacteristicReferenceIdentifier and for each ProductCharacteristicInstanceIdentifier.

5.4.4.1 *ProductCharacteristicBaseTag*

The ProductCharacteristicBaseTag is a ProductCharacteristicBaseIdentifier that is often identified as a numeric with an optional letter character prefix (e.g., PC007, 7, 007). The PC Base Tag shall be a unique within the local context of the product definition.

Format: PC"numeric" or "numeric"

Example: PC007 or 7

The Base Tag numeric makes no distinction between the number of leading zeros for the PC Tag's value digits, therefore PC1, PC01, PC001, or PC0001 are all the same identifier. Likewise, 2, 02, 002, and 0002 are all the same identifier.

Furthermore, the uniqueness of the PC Base Tag is according to the numeric, therefore, a prefix or different prefixes do not change the relative uniqueness. As a result, PC007, 007, and RT007 are not unique identifiers as they all share the same numeric. Therefore, within the context of a product definition, any non-product characteristics general tags such as GT tags, discussed in section 5.19, shall use different numeric than what is currently be used for PC Tag. One suggestion is for the organization or application to reserve a set of numerics for GT Tags (e.g., 500-899) usage.

Finally, to enable future extensions to this standard with additional characteristic use-cases, the recommended practice for labeling a PC Base Tag should use a "PC" prefix, but any prefix is allowed, and no prefix is also allowed.

5.4.4.2 *ProductCharacteristicBaseUUID*

The ProductCharacteristicBaseUUID is a ProductCharacteristicBaseIdentifier that is used as a persistent machine-readable universally unique identifier (UUID). The UUID should be generated per the ISO/IEC 9834-8 standard. Below shows an example of a ProductCharacteristicBaseUUID.

CB23952F-0518-1961-0814-39020C268C13

The ProductCharacteristicBaseUUID is a ProductCharacteristicBaseIdentifier that is a universally unique identifier (UUID).

The Quality Information Framework (QIF) concept of a QIF Persistent ID (QPId) is implemented on the Product Characteristic as a universally unique identifier (UUID) element and is a persistent identifier for the characteristic tag. UUIDs are described in ISO/IEC 9834-8, which specifies the format and the generation rules that enable users to produce 128-bit identifiers. A UUID generated following the rules has an exceptionally high probability of being unique among all UUIDs ever generated.

For a digital implementation of a Characteristic, the UUID should be implemented as an attribute value of the Characteristic. When used, it should be generated using a widely competent UUID generator per the ISO/IEC 9834-8 standard.

5.4.5 ProductCharacteristicReferenceIdentifier

The ProductCharacteristicReferenceIdentifier element is the identifier of a Product Characteristic, which is constructed from the Product Characteristic Base Identifier. A ProductCharacteristicReferenceIdentifier shall have one or more instances that may be explicitly identified by their ProductCharacteristicInstanceIdentifier.

5.4.5.1 *ProductCharacteristicReferenceTag*

The ProductCharacteristicReferenceTag is the human-readable identifying text that is unique within the context of its product definition. It is constructed from the

ProductCharacteristicBaseTag within a symbol or textually between “less-than” and “greater-than” character, such as <PC007>. Since the PC Reference Tag is constructed from the PC Base Tag, the PC Tag numeric shall be unique within the local context of the product definition, see section 5.4.4.1. Like the PC Base Tag, below shows examples of a PC Tag, one with a “PC prefix and one without a prefix.

Format: <PC”numeric”>	or	<”numeric”>
Example: <PC007>	or	<007>

5.4.5.2 *ProductCharacteristicReferenceTag as Text*

The recommended textual format for a PC Reference Tag (e.g., PC007) is with the Tag preceded by a “less than” character and followed by a “greater than” character. The general format is < PC Reference Tag >. Some examples are: <PC007>, <42>. An example of textually using a PC Reference Tag within a sentence is “Please investigate product characteristic <PC201> on every part in Lot # 303 for possible workmanship nonconformance”.

5.4.5.3 *ProductCharacteristicReferenceUUID*

The ProductCharacteristicReferenceUUID is a ProductCharacteristicReferenceIdentifier element that is primarily intended as a machine-readable universally unique identifier (UUID) of a ProductCharacteristic occurrence. The ProductCharacteristicReferenceUUID shall be identified by its ProductCharacteristicBaseUUID, which enables persistence and digital connectivity.

5.4.6 ProductCharacteristicInstanceIdentifier

The ProductCharacteristicInstanceIdentifier element is the identifier of a unique ProductCharacteristic instance. The ProductCharacteristicInstanceIdentifier shall include the ProductCharacteristicBaseIdentifier plus one or more Extension Identifiers (e.g., PC008.1)

When a Product Characteristic Reference has multiple occurrences or instances, each Product Characteristic Instance shall be a separate verification requirement. Multiple permutations of instances may be caused by:

- a multi-faceted annotation
- a multiple feature reference annotation
- a repetitive group number annotation

Note: If a Product Characteristic Reference Identifier has only one instance, then the Product Characteristic Instance Identifier is not necessary. However, for completeness, the Product Characteristic Instance Identifier would be the same as the Product Characteristic Reference Identifier.

5.4.6.1 *ProductCharacteristicInstanceTag*

The ProductCharacteristicInstanceTag element is primarily intended as a human-readable identifier of a unique ProductCharacteristic instance. This tag shall be unique within the scope of an instance of a product. The PC Instance Tag will be constructed from the

Product Characteristic Base Tag along with any optional Extension Identifiers, such as PC007.1.

5.4.6.2 *ProductCharacteristicInstanceUUID*

The ProductCharacteristicInstanceUUID element is primarily intended as a machine-readable identifier of a unique ProductCharacteristic instance.

5.4.7 ProductCharacteristicInstanceTag with Extension Identifiers

Extension Identifiers uniquely extend the explicit identification of reference identifiers for use-cases that involve multi-faceted annotations, multiple feature references, and/or repetitive groups. The Extension Identifiers concatenate with the PC Base Identifier to assure that the PC Instance Identifier uniquely identifies the instance of the PC. Section 5.4.9 describes the format of how one-to-many optional extension identifiers are explicitly used to define PC Instance Identifiers.

- Multi-Faceted Annotations Identifier (MFAI). See section 5.4.7.1 for more details.
- Multiple Feature Reference Identifier (MFRI). See section 5.4.7.2 for more details.
  - Feature Reference Tag (FRT). See section 5.4.7.2.1 for more details.
  - Feature Reference UUID (FRU). See section 5.4.7.2.2 for more details.
  - Feature Reference Sequence Number (FRSN). See section 5.4.7.2.3 for more details.
- Repetitive Group Number Identifier (RGNI). See section 5.4.7.3 for more details.

5.4.7.1 *MultiFacetedAnnotationIdentifier (MFAI)*

A single annotation may be a multi-faceted annotation, which contains multiple verification requirements. When a single annotation contains multiple facets, then each facet's PC Base Identifier shall be extended by a MFAI starting from the alpha "a" proceeding in alphabetical order. Multi-faceted annotation examples are addressed in sections 5.7 and 5.85.6.14. A MFAI within a PC Instance Identifier shall be concatenated to the PC Base Tag as shown below.

- PC007a
- PC007b
- PC007c

5.4.7.2 *MultipleFeatureReferenceIdentifier (MFRI)*

For annotations that are explicitly applied to more than one feature instance, an identification framework for a multi feature reference shall be by one of three MultipleFeatureReferenceIdentifier options:

- By FeatureReferenceTag (e.g., 518, 1961, 1982)
- By FeatureReferenceUUID (e.g., A64B5992-3A8B-456C-81E6-39020C268C13, C3F0BAB3-DA40-4C21-B04B-DC495336A4D0)
- By FeatureReferenceSequenceNumber (e.g., 1, 2, 3, 4)

MultipleFeatureReferenceIdentifier options are based upon the maturity and capabilities of the application system and/or organizational business practices.

For the MFRI use case, the PC Instance Identifier contains the PC Base Identifier, supplemented with a MFRI to identify explicit uniqueness and application. A MFRI within a PC Instance Identifier shall be delimited by an ASCII period character "." as shown below.

Format: PCBaseTag.MFRI

As stated above, there are three options for identifying a MFRI. The three following sections shows examples of a PC Reference Tag, <PC007>, which has a PC Base Tag of PC007, which is extended by each type of MFRI option.

5.4.7.2.1 FeatureReferenceTag (FRT)

The FeatureReferenceTag is a MFRI that is a human-readable tag when a "feature" in the application system has its own integer tag reference. Examples of a FeatureReferenceTag include 518, 1961, and 1982. Therefore, the ProductCharacteristicInstanceTag with ProductCharacteristicBaseTag, "PC007" would be:

- PC007.518
- PC007.1961
- PC007.1982

5.4.7.2.2 FeatureReferenceUUID (FRU)

The FeatureReferenceUUID is a MFRI that is a machine-readable identifier when a "feature" in the application system (e.g., QIF) has its own UUID reference. Examples of a FeatureReferenceUUID include A64B5992-3A8B-456C-81E6-39020C268C13 and C3F0BAB3-DA40-4C21-B04B-DC495336A4D0. Therefore, the ProductCharacteristicInstanceTag with ProductCharacteristicBaseTag, "PC007" would be:

- PC007. A64B5992-3A8B-456C-81E6-39020C268C13
- PC007. C3F0BAB3-DA40-4C21-B04B-DC495336A4D0

5.4.7.2.3 FeatureReferenceSequenceNumber (FRSN)

The FeatureReferenceSequenceNumber is a MFRI that is a human-readable tag, used when there is no known feature identifier. Typically, the feature sequence identifier will start with the number "1" and will be sequenced numerically. Examples of a FeatureReferenceSequenceNumber include 1, 2, 3, 4, and so on. The determination of which feature the FRSN applies to may be problematic and examples are discussed in section 5.9. Therefore, the ProductCharacteristicInstanceTag with ProductCharacteristicBaseTag, "PC007" would be:

- PC007.1
- PC007.2
- PC007.3
- PC007.4

Note: FRSN may be problematic and possibly non-repeatable, however it is suggested that the sequence start with the first feature that the annotation leader line points to and then with respect to the camera view of the features, proceed along the sequence of features in a clockwise direction from the initial feature.

5.4.7.3 *RepetitiveGroupNumberIdentifier (RGNI)*

Occasionally, groups of features are repetitively specified by the use of an "X" in conjunction with a numeral to indicate the number of repetitions (e.g., 6X). For an annotation, that has repetitive group number, and then the PC Base Identifier shall be extended by a RGNI starting from "1" to the repetitive number. A RGNI within a PC Instance Identifier shall be delimited by an ASCII colon character ":" as shown below for a repetitive group of four.

- PC007.518:1
- PC007.1961:1
- PC007.1982:1
- PC007.518:2
- PC007.1961:2
- PC007.1982:2
- PC007.518:3
- PC007.1961:3
- PC007.1982:3
- PC007.518:4
- PC007.1961:4
- PC007.1982:4

5.4.8 ProductCharacteristicInstanceUUID with Extension Identifiers

As the ProductCharacteristicExtensionIdentifier discussed in section 5.4.7 for ProductCharacteristicInstanceTag, the ProductCharacteristicInstanceUUIDs follow the same format, except the ProductCharacteristicBaseTag will be replaced by the ProductCharacteristicBaseUUID.

5.4.9 Product Characteristic Instance Identifier Format

The Product Characteristic Instance Identifier format shall accommodate all permutations of the optional Extension Identifiers. The identification framework of a Product Characteristic Instance Identifier requires a format with a Product Characteristic Base Identifier followed by zero-to-many optional Extension Identifiers.

Format: "*PC_Base_ID" + opt{"PC_Extension_ID"}*

Where:

PC_Base_ID is the Base Identifier of the PC.

PC_Extension_ID is the Extension Identifier that makes the instance of the PC uniquely identified.

Format: *PC_Extension_ID* = opt{*MFAI*} + opt{.*MFRI*} + opt{:*RGNI*}

Where:

Opt = optional

MFAI is the optional multi-faceted annotation identifier as alpha character.

MFRI is the optional multiple feature reference identifier. The MFRI shall be formatted based upon the capability of the feature-based application.

Format: *MFRI* shall be a:

- Feature Reference Tag (FRT),
- Feature Reference UUID (FRU), or
- Feature Reference Sequence Number (FRSN).

RGNI is the optional repetitive group number identifier as numeric.

The three optional Extension Identifier types may result in multiple permutations; therefore, each Extension Identifier type may be identified by its usage and/or its delimiter.

Examples of these permutations with their optional Extension Identifiers are described later within this section. For the non-typical occurrence, where each permutation instance shall be accommodated, then a unique Product Characteristic Instance Identifier could have an Instance Tag.

Example: PC007b.518:4

For the PC Instance Tag example shown above, the PC007 is the Product Characteristic Reference Tag, "b" is the second facet of a multi-faceted annotation, "518" is a feature Reference Tag, and the numeric "4" is the fourth repetitive group.

A PC Instance ID Table for the PC007b.518.4 PC Extension ID is shown below.

PC Base ID	PC Extension ID			PC Instance ID Example
	MFAI	MFRI	RGNI	
PC007	Y	Y as FRT	Y	PC007b.518:4

A PC Instance ID Table that shows examples of additional different permutations of PC Extension IDs is shown below.

PC Base ID	PC Extension ID			PC Instance ID Example
	MFAI	MFRI	RGNI	
Zero Options				
PC007	n	n	n	PC007
One Option				
PC007	Y	n	n	PC007a
PC007	n	Y as FRSN	n	PC007.1
PC007	n	Y as FRT	n	PC007.518
PC007	n	Y as FRU	n	PC007.651ADED1-FF04-498A-968E-044147A25064
PC007	n	n	Y	PC007:1
Two Options				
PC007	Y	Y as FRSN	n	PC007a.1
PC007	Y	Y as FRT	n	PC007a.518
PC007	Y	Y as FRU	n	PC007a.651ADED1-FF04-498A-968E-044147A25064
PC007	Y	n	Y	PC007a.1
Three Options				
PC007	Y	Y as FRSN	Y	PC007a.1:1

PC007	Y	Y as FRT	Y	PC007a.518:1
PC007	Y	Y as FRU	Y	PC007a.651ADED1-FF04-498A-968E-044147A25064:1

Refer to sections within 5.6.14, 5.7, 5.8, 5.9, and 5.10 for additional test-case examples. See Section 5.11 for test-case examples that involve mixed permutations of multi-facet annotation, multiple feature reference, and repetitive group number.

5.4.10 PC Reference Tag with Augmentations

As discussed later within this standard, a ProductCharacteristic may have one-to-many augmentations (e.g., Criticality Classification, Product Requirement Association, and Verification Plan Requirement). The example below shows the textual-base form of a <PC007> , with a pre-augmentation of a ProductRequirementAssociation designated as <REQ-ME-4< and then a post-augmentation of a CriticalityClassification designation of >CR: S.4>.

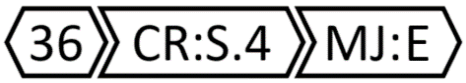
<REQ-ME-4< <PC007> >CR: S.4>

5.4.10.1 *Product Characteristic with Criticality Classification Example*

This example shows a ProductCharacteristicDesignator of <24> that is augmented with a Criticality Classification. The optional CriticalityClassification is placed as the first post-augmentation or after and in-line with the ProductCharacteristicDesignator symbol. Further information on Criticality Classification is given in section 6.

<24>>CR:S.4>

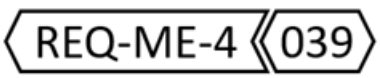
Another example shows a ProductCharacteristicDesignator of <36> that is augmented with two Criticality Classifications. The first Criticality Classification at a “Critical” level for “Safety” category with caveat “4” and then a second Criticality Classification at a “Major” level for “Environmental” category.



<36> >CR: S.4>>MJ: E>

5.4.10.2 *Product Characteristic with Product Requirement Association Example*

This example shows a ProductCharacteristicDesignator of <039> that is augmented with a ProductRequirementAssociation. The ProductRequirementAssociation is placed before and in-line with the Product Characteristic. Reference section 7.0 Product Requirement Association Augmentation.



<REQ-ME-4<<039>

5.4.10.3 *Product Characteristic with Verification Plan Requirement Example*

This example shows a ProductCharacteristicDesignator of <061> that is augmented with a VerificationPlanRequirement. Any VerificationPlanRequirement is placed as a post-augmentation or after and in-line with the ProductCharacteristicDesignator symbol and the optional CriticalityClassification. Reference section 8.0 Verification Plan Requirement Augmentation

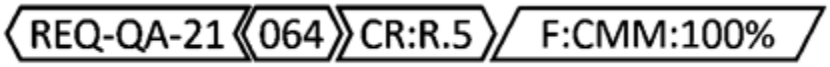
<061> /D: CMM: 100%/

Another example shows a ProductCharacteristicDesignator of <063> that is augmented with both a Criticality Classification and a VerificationPlanRequirement.

<063>>CR: R.5>/CMM/

5.4.10.4 *Product Characteristic with Multiple Augmentations*

This example shows a PC Reference Tag that is augmented with a Criticality Classification, a Product Requirement Association, and a VerificationPlanRequirement.



<REQ-QA-21<<064>>CR: R.5> /F: CMM: 100%/

5.5 Annotation Applications

A product definition is used to communicate the physical or functional characteristics of a product, to ensure design intent. A mechanical product designer defines the geometric shape of the part and places visible annotations to help communicate non-shape information (e.g., allow deviations from nominal). For the quality function, many of these visible annotations manifest themselves as verification requirements that become product characteristics, and, hence, should be identified for verification. Typically, a Product Characteristic annotation is a tolerance or specification applied to a shape feature, geometric element, or the overall product. Traditionally, from a drawing-based perspective, an annotation typically represents a single Product Characteristic for a feature. However, for convenience, a drawing-based designer could create one explicit annotation element that applies to multiple features which implies multiple product characteristics or one group of annotations that applies to one feature, which implies multiple product characteristics.

The sub-sections below attempt to address typical use-cases seen in practice or illustrated in geometric and dimension tolerancing standards.

5.6 Single Product Characteristic with Single Application (Feature)

The ideal approach for assigning Product Characteristics is a one-to-one mapping, which is one PC Reference Tag to a single feature application, resulting in one Product Characteristic instance.

5.6.1 Geometric Tolerance

Each occurrence of a geometric tolerance annotation shall be a unique Product Characteristic instance with one unique PC Reference Tag and one unique PC Reference UUID.

5.6.2 Dimensional Tolerance (limit, plus/minus, single Limit (MAX, MIN))

In general, each occurrence of a dimensional tolerance (numeric dimension with tolerance specification) annotation will be a unique Product Characteristic instance with one unique PC Reference Tag and one unique PC Reference UUID.

5.6.2.1 *Size (Diameter, Width) Dimension*

Each occurrence of a size dimensional tolerance annotation shall be a unique Product Characteristic instance with one unique PC Reference Tag and one unique PC Reference UUID.

5.6.2.2 *Distance-Between Dimension*

Each occurrence of a linear distance-between dimensional tolerance annotation shall be a unique Product Characteristic instance with one unique PC Reference Tag and one unique PC Reference UUID.

5.6.2.3 *Radial Dimension*

Each occurrence of a radial dimensional tolerance annotation shall be a unique Product Characteristic instance with one unique PC Reference Tag and one unique PC Reference UUID.

5.6.2.4 *Depth Dimension*

Each occurrence of a depth dimensional tolerance annotation shall be a unique Product Characteristic instance with one unique PC Reference Tag and one unique PC Reference UUID.

5.6.2.5 *Angle*

Each occurrence of an angle dimensional tolerance annotation shall be a unique Product Characteristic instance with one unique PC Reference Tag and one unique PC Reference UUID.

5.6.2.6 *Multiple Verification Requirement Dimensional Tolerance Annotations*

Often, a dimensional tolerance annotation may include multiple verification requirements within a single line annotation or within a multiple-line annotation. Each verification requirement within an annotation shall be uniquely identified, either at the PC Reference Tag level or at PC Instance Tag level. There is a preference that each verification requirement within an annotation should obtain this uniqueness at the PC Reference Tag level.

To implement the preference for multiple inline product characteristics, multiple PC Reference Tags should be placed in-line, one for each respective product characteristic

Below is single-line, multiple verification requirement annotation example, representing a blind-hole with a size tolerance and a depth tolerance. The preference is two PC Reference Tags <71> and <72>. PC Reference Tag <71> applies to the hole size tolerance, and <72> applies to the hole's depth tolerance.

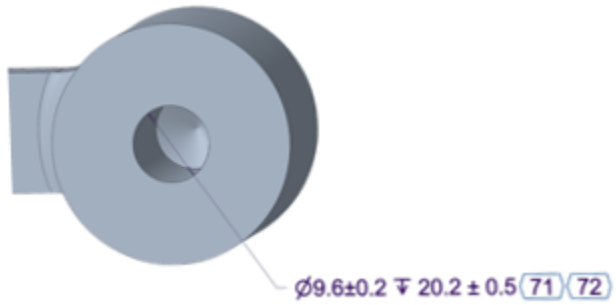


Figure 10: PC Reference Tags for Single Line, Multiple Verification Requirement Annotation

To implement the preference for multiple line product characteristics, multiple PC Reference Tags should be placed after each line, one for each respective product characteristic.

Below is multi-line, multiple verification requirement annotation example, representing a blind-hole with a size tolerance and a depth tolerance. Again, the preference is two PC Reference Tags <71> and <72>. PC Reference Tag <71> applies to the hole size tolerance, and <72> applies to the hole depth tolerance.

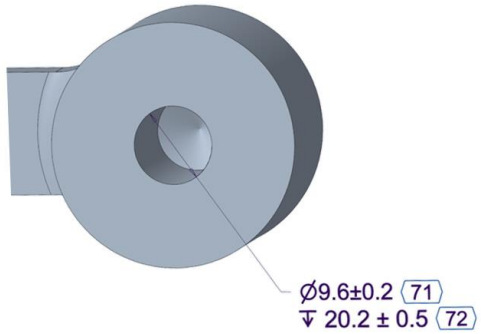


Figure 11: PC Reference Tags for Multi- Line, Multiple Verification Requirement Annotation

An acceptable alternative approach for a single-line, multiple verification requirement annotation as a single multi-faceted PC reference Tag is discussed in section 5.8.1.

5.6.3 Product Definition Notes

Organizations may apply notes as part of their product definition. Notes may be considered as:

- General Notes (Reference section 5.6.4),
  - Group General Notes
  - Individual General Notes
  - Supplemental Document General Notes

- Flagged or Local Notes (Reference section 5.6.7), and
- Supplemental Notes (Reference section 5.6.8).

5.6.4 General Note

General notes typically apply to the overall product definition. Those individual general notes that invoke a verification requirement shall be considered as a unique Product Characteristic and require instance identifiers. General notes typically are numbered and may be defined within a graphical drawing, a model-based definition, or a separate supplemental document. Today's computer aided drafting applications allow drafters to represent general notes via two approaches:

- Individual General Notes, see section 5.6.5.
- Group General Note, see section 5.8.6.

5.6.5 Individual General Notes

An individual General Note displays each individual note as an individual note annotation. Figure 12 shows an individual note annotation in which general note #4 has been previously deleted. Each instance of a general note annotation that invokes a verification requirement will be a unique Product Characteristic instance with one unique PC Reference Tag and one unique PC Reference UUID. A Group General Note example is described in section 5.8.6.

5.6.5.1 *Example*

Computer-aided drafting systems may represent general notes as multiple individual annotations where each annotation entity contains an individual general note.

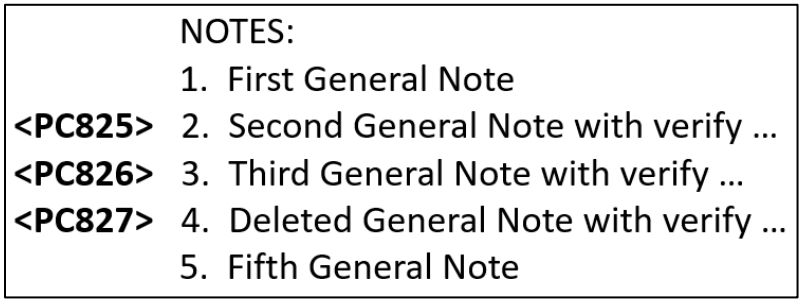


Figure 12: Individual General Note Example

5.6.5.2 *Application:*

For this example, each display note annotation will obtain a unique Reference Tag. Likewise, each display note annotation will obtain a PC Reference UUID. For this example, the individual note's PC Reference Tags would be <PC825>, <PC826>, and <PC827>. Likewise, the PC Referenced UUIDs will all be different.

Notice that PC Reference Tag <PC827> is still assigned to a General Note that has been deleted. This helps maintain integrity of the PC Reference Tags and helps prevent reuse of tags within the local product definition.

5.6.5.3 *BOC Example:*

Below is the BOC example listing PC Instance Tags, PC instance UUIDs and their PC descriptions for the use case example shown above.

Tag (Base)	Tag (Ext)	UUID (Base)	UUID (Ext)	Description
PC825	N/A	1af1745b-d897-491c-8887-05404c5941cc	N/A	Second General Note
PC826	N/A	29ef8155-113d-4f0d-8159-b766c836065d	N/A	Third General Note
PC827	N/A	cef53b0d-8891-4634-8cb9-0dd50ac25e53	N/A	Deleted General Note

Towards accommodating explicit product characteristics, it is preferred to have individual general notes; however, systems may uniquely accommodate both approaches as shown above.

5.6.6 Supplemental Document General Note

General notes may also be defined and communicated within a supplemental document; therefore, any notes within a supplemental document that invoke a verification requirement shall be identified as a Product Characteristic using either technique described previously.

5.6.6.1 *Example*

A sample supplemental product definition document with special specifications is shown below that identifies additional verification requirements for a product identified as 1A345.

SS1A345
Special Specifications for part 1A345:
1 – Special Workmanship
1.1 – Statement about …
1.2 – Statement about …
1.3 – Statement about …
1.4 – Verify that …
1.5 – Verify that …
1.6 – Statement about …
1.7 – Verify that …
1.8 – Verify that …
2 – Special Markings
2.1 – Statement about …
2.2 – Verify that …
2.3 – Statement about …

Figure 13: Supplemental Document with Verification Requirements Example

5.6.6.2 *Application:*

Each of the verification requirements, identified in SS1A345, should be a product characteristic and shall be identified accordingly with PC Reference Tags. Below, we see PC Reference Tags <PC901>, <PC902>, <PC903>, <PC904>, and <PC905> being applied to sections of SS1A345 that have verification requirements.

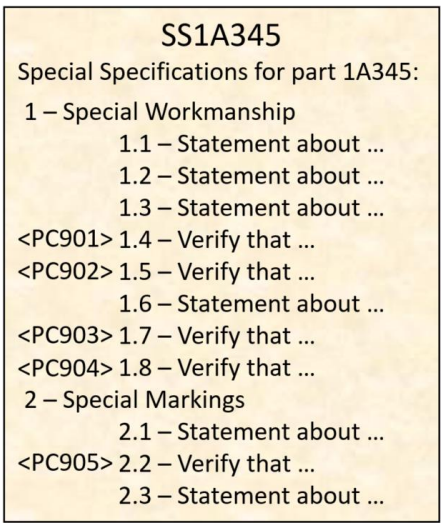


Figure 14: Supplemental Document with Verification Requirements Application

5.6.6.3 *BOC Example:*

Below is the BOC example listing PC Instance Tags, PC instance UUIDs and their PC descriptions for the use case example shown above.

Tag (Base)	Tag (Ext)	UUID (Base)	UUID (Ext)	Description
PC901	N/A	48bf4540-63eb-4460-999d-bdaffad3f793	N/A	Doc. SS1A345, Sect. 1.4
PC902	N/A	af11745b-d897-491c-8887-05404c5941cc	N/A	Doc. SS1A345, Sect. 1.5
PC903	N/A	9e2f8155-113d-4f0d-8159-b766c836065d	N/A	Doc. SS1A345, Sect. 1.7
PC904	N/A	ecf53b0d-8891-4634-8cb9-0dd50ac25e53	N/A	Doc. SS1A345, Sect. 1.8
PC905	N/A	7bbfbae1-65d8-4970-9987-da22e84f5403	N/A	Doc. SS1A345, Sect. 2.2

5.6.7 Flag or Local Note

Each occurrence of a flag or locally applied note annotation will be a unique Product Characteristic reference with one unique tag and one unique UUID.

5.6.7.1 *Example*

This example shows a Flag Note #2 that is amongst a list of general notes. The Flag Note #2 indicates that markings for the part name, part number, and logos as shown per "Class K-1-A" per an organizational standard 8819100.

NOTES:

1. PRODUCT CHARACTERISTICS ARE DENOTED BY <PC####>. GENERAL TAGS ARE DENOTED BY <GT####>. THESE SYMBOLS ARE USED TO REFER TO CORRESPONDING ANNOTATIONS.
   1A. THERE ARE 52 PRODUCT CHARACTERISTICS ON THIS DRAWING. THE NUMBERS RANGED FROM PC0001 TO PC0054. NUMBERS NOT USE ARE: PC0013.
   1B. THERE ARE 27 GENERAL TAGS ON THIS DRAWING. THE NUMBERS RANGED FROM GT5000 TO GT5026. NUMBERS NOT USED ARE: NONE.
2. PART MARKINGS:

2. MARK PART NAME, PART NUMBER, AND LOGOS AS SHOWN PER CLASS K-1-A, PER 8819100.

3. DRILL POINT ANGLES OPTIONAL
4. UNIFORM TOLERANE ZONE BOUNDARY IN ACCORDANCE WITH ASME Y14.5-2009
5. SUPPORT MODEL REPRESENTS NOMINAL GEOMETRY PER 8825016
6. CHROMATE COAT PER 8804151, TYPE 1, CLASS 3.

The application of Flag Note 2 is applied explicitly to three geometric elements, as shown in the figure below.

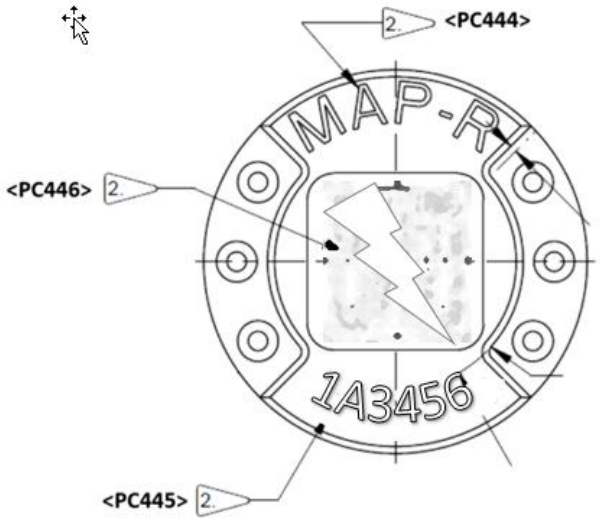


Figure 15: PC Reference Tag Example on Marking Flag Note Annotations

5.6.7.2 *Application:*

For this example, the Flag Note #2 for "PART MARKINGS" are applied three places; as shown, each Flag Note #2 receives an explicit a Product Characteristic Reference Identifier. <PC444> for marking the Part Name, "MAP-R", <PC445> for marking the Part Number, "4A1473-00", and <PC446> for marking the Logo of a thunderbird.

5.6.7.3 *BOC Example*

Below is the BOC example listing PC Instance Tags, PC instance UUIDs and their PC descriptions for the use case example shown above.

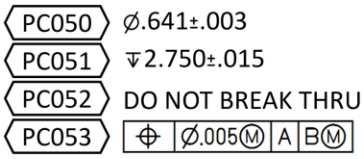
Tag (Base)	Tag (Ext)	UUID (Base)	UUID (Ext)	Description
PC444	N/A	f0079160-e989-420a-b83a-19df30d57c7b	N/A	Marking of Part Name
PC445	N/A	7f92f319-3175-4214-8eb2-62f857f1c3b4	N/A	Marking of Part Number
PC446	N/A	5ce27d7f-e66d-404a-8327-d7c24229b20d	N/A	Marking of Logo

5.6.8 Supplement Note

Supplemental notes contribute to an existing Product Characteristic annotation to convey additional information that may invoke a separate verification requirement. Each instance of a supplement note annotation that invokes a separate verification requirement shall be a unique Product Characteristic instance with one PC Reference Tag and one PC Reference UUID.

5.6.8.1 *Example*

Example includes a "DO NOT BREAK THRU" supplementary note for a .641 diameter hole with 2.750 depth.



5.6.8.2 *Application:*

For this example, the supplemental note is a verification requirement and thus a product characteristic, which has a reference identifier of <PC052>.

5.6.8.3 *BOC Example*

Below is the BOC example listing PC Instance Tags, PC instance UUIDs and their PC descriptions for the use case example shown above.

Tag (Base)	Tag (Ext)	UUID (Base)	UUID (Ext)	Description
PC050	N/A	ccba74d2-cd72-46cd-9196-d137d94a00e7	N/A	Ø .641 ±.003
PC051	N/A	a2e26814-e971-41e4-8ab2-cad079682020	N/A	↧2.750 ±.015
PC052	N/A	ac2e6b4f-4c02-417a-8d96-f122f0fc7978	N/A	DO NOT BREAK THRU
PC053	N/A	42837051-e747-46f2-b15b-73b7683774d1	N/A	[Pos | Ø.005m | A | Bm]

5.6.9 Surface Texture

An occurrence of a surface texture annotation may be a unique Product Characteristic instance with one unique tag and one unique UUID. However, a surface texture may be multi-faceted with multiple subcomponents, which may require multiple Product Characteristic Instances, see section 5.8.5.

5.6.9.1 *Example*

The example below is a surface texture symbol, per ASME Y14.36M-2018, with only a maximum roughness average (Ra) of 3.2 µm.

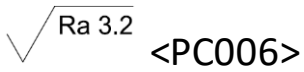


Figure 16: Average Roughness Surface Texture

5.6.9.1 *Application:*

For this example, the surface texture has a Reference Tag of <PC006>.

5.6.9.2 *BOC Example*

Below is the BOC example listing the one PC Instance Tag, one PC instance UUID and one PC description for the PC Reference Tag example shown above.

Tag (Base)	Tag (Ext)	UUID (Base)	UUID (Ext)	Description
PC006	N/A	745A8DDB-242E-401D-96DB-34C387797718	N/A	Surface Texture 3.2 µm Ra

5.6.10 Thread Specification

An occurrence of a thread specification may be a unique Product Characteristic instance with one unique PC Reference Tag and one unique PC Reference UUID. However, a thread specification may be multi-faceted with multiple subcomponents, which may require multiple Product Characteristic Instances, see section 5.7.2.

5.6.10.1 *Example*

The figure below shows a metric thread specification of M6x1 – 5H6H with a maximum drill depth of 9mm and a minimum full thread depth of 6.3 mm, which is positionally located with respect to datum reference frame, DBmCm. Furthermore, it is applied to four different threaded holes.

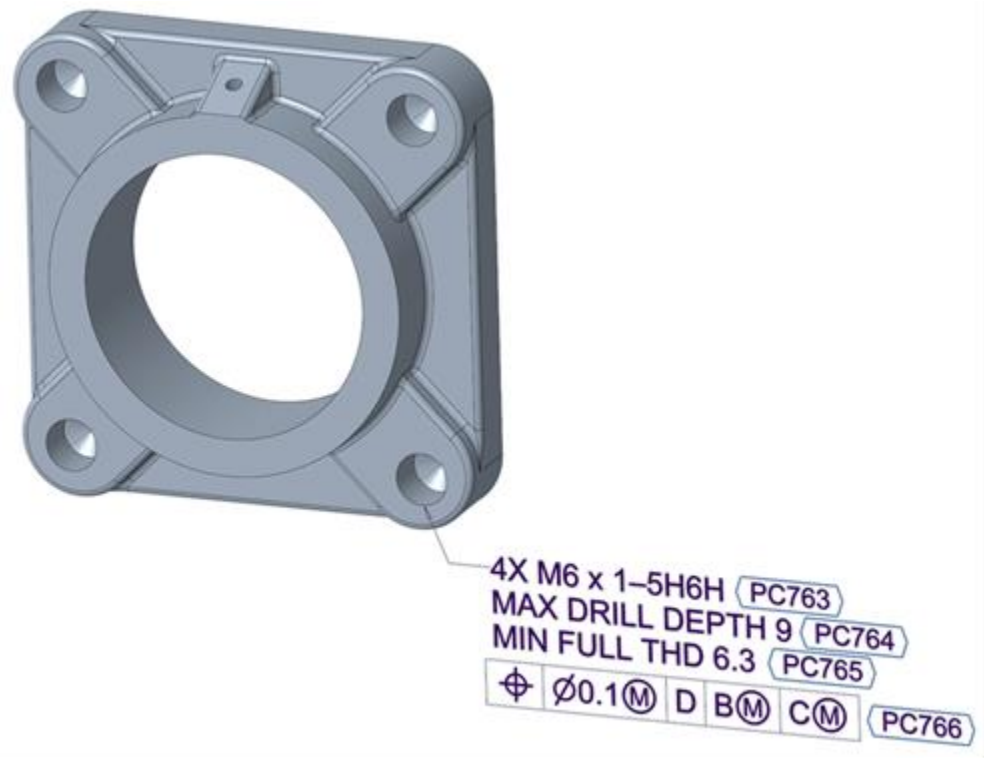


Figure 17: Thread Specification Applied to Multiple Features

5.6.10.1 *Application:*

In this example, the threaded hole specification's annotation is tagged with one PC Reference Tag, <PC763>, the maximum drill depth is tagged with PC Reference Tag, <PC764>, the minimum full thread depth is tagged with PC Reference Tag, <PC765>, and the geometric tolerance has one PC Reference Tag, <PC766>. Furthermore, the annotation is repetitively applied four times.

The PC Instance ID Table for this multiple feature use case is shown below:

PC Base ID	PC Extension ID			PC Instance ID Example
	MFAI	MFRI	RGNI	
PC763	n	Y	n	PC763.1

5.6.10.2 *BOC Example*

Below is the BOC example listing PC Instance Tags with their multiple feature extensions, PC instance UUIDs with their multiple feature extensions and PC descriptions for the use case example shown above.

Tag (Base)	Tag (Ext)	UUID (Base)	UUID (Ext)	Description
PC763	1	215168B5-BE94-4A7E-8AE9-1E96ACB04EF3	1	Thread spec M6x1-5H6H on Hole 1
PC764	1	00E9BCD5-E654-4F61-A3E8-248BB4A1B699	1	Max Drill Depth9mm on Hole 1
PC765	1	03A0F8E3-986F-41BD-8A00-DD2C10DC5F05	1	Min Full Thd 6.3mm on Hole 1
PC766	1	EC346666-D854-4415-8E42-B34FC77CC054	1	PosTol w.r.t. DBmCm on Hole 1
PC763	2	215168B5-BE94-4A7E-8AE9-1E96ACB04EF3	2	Thread spec M6x1-5H6H on Hole 2
PC764	2	00E9BCD5-E654-4F61-A3E8-248BB4A1B699	2	Max Drill Depth9mm on Hole 2
PC765	2	03A0F8E3-986F-41BD-8A00-DD2C10DC5F05	2	Min Full Thd 6.3mm on Hole 2
PC766	2	EC346666-D854-4415-8E42-B34FC77CC054	2	PosTol w.r.t. DBmCm on Hole 2
PC763	3	215168B5-BE94-4A7E-8AE9-1E96ACB04EF3	3	Thread spec M6x1-5H6H on Hole 3
PC764	3	00E9BCD5-E654-4F61-A3E8-248BB4A1B699	3	Max Drill Depth9mm on Hole 3
PC765	3	03A0F8E3-986F-41BD-8A00-DD2C10DC5F05	3	Min Full Thd 6.3mm on Hole 3
PC766	3	EC346666-D854-4415-8E42-B34FC77CC054	3	PosTol w.r.t. DBmCm on Hole 3
PC763	4	215168B5-BE94-4A7E-8AE9-1E96ACB04EF3	4	Thread spec M6x1-5H6H on Hole 4
PC764	4	00E9BCD5-E654-4F61-A3E8-248BB4A1B699	4	Max Drill Depth9mm on Hole 4
PC765	4	03A0F8E3-986F-41BD-8A00-DD2C10DC5F05	4	Min Full Thd 6.3mm on Hole 4
PC766	4	EC346666-D854-4415-8E42-B34FC77CC054	4	PosTol w.r.t. DBmCm on Hole 4

5.6.11 Composite Tolerance as Separate PC Reference Tags

Composite tolerances (e.g., Composite Position, Composite Profile) are a tolerance treated by the standards with a composite feature control frame containing a single geometric symbol (e.g., Position, Profile) followed by two or more segments, each containing tolerance and any required datum references, one above the another. The preferred approach to characterize this type of annotation is for each segment to be assigned a separate PC Reference Tag. An alternative approach using the multi-faceted method is described in section 5.8.5.

5.6.11.1 *Example*

This example shows a two segmented composite position tolerance with two separate Reference Tags.

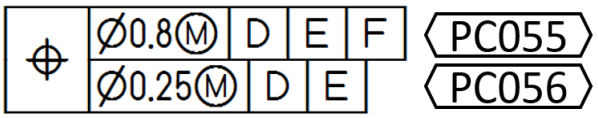


Figure 18: Composite Position Tolerance with Segment Reference Tags

5.6.11.2 *Application:*

For the alternative composite tolerance example, each composite segment acquires its own PC Reference Tag, <PC055> for the top segment and <PC056> for the lower segment.

5.6.11.3 *BOC Example*

Below is the BOC example listing PC Instance Tags, PC instance UUIDs and their PC descriptions for the use case example shown above.

Tag (Base)	Tag (Ext)	UUID (Base)	UUID (Ext)	Description
PC055	N/A	36646821-f9e9-4834-93b2-59c9170dbd8f	N/A	[CPos|0.8m|D|E|F] on Feat1
PC056	N/A	593ff034-23ed-4fc2-b7e2-c6b319506fcf	N/A	[CPos|0.25m|D|E] on Feat1

5.6.12 Individual Multiple Segment Feature Control Frame

In contrast to a composite tolerance, a multiple-segment feature control frame shows multiple individual single segments in which each segment is a product characteristic.

5.6.12.1 *Example*

This example shows a two-segmented position tolerance with two separate PC Reference Tags.

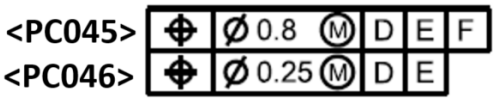


Figure 19: Two-Segment Feature Control Frame Tolerance

5.6.12.2 *Application:*

For a multiple single segment positional tolerance annotation, each single segment positional tolerance will have a separate PC Reference Tag for each segment such as <PC045> and <PC046>.

5.6.12.3 *BOC Example*

Below is the BOC example listing PC Instance Tags, PC instance UUIDs and their PC descriptions for the use case example shown above.

Tag (Base)	Tag (Ext)	UUID (Base)	UUID (Ext)	Description
PC045	N/A	DDC7053F-A663-4561-94E1-F423E97A9568	N/A	[Pos|0.8m|D|E|F] on Feat1
PC046	N/A	9130DA8C-BF0F-48E9-B8EC-785BCBD52176	N/A	[Pos|0.25m|D|E] on Feat1

5.6.13 Mixed Multiple Segment Feature Control Frame

5.6.13.1 *Example*

This example shows a three-segment feature control frame. The top segment is a surface profile tolerance with respect to datum reference frame ABC, middle segment is a parallel tolerance with respect to datum reference frame A, and bottom segment is a flatness tolerance.

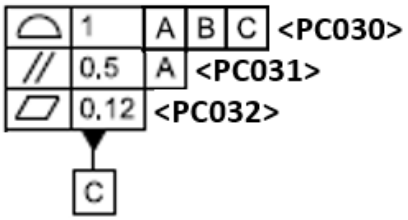


Figure 20: Mixed Multiple Segment Feature Control Frame

See section 5.8.9 for a mixed multi-faceted feature control frame as multiple PC Reference Tags.

5.6.13.2 *Application:*

This represents three different product characteristics applied on datum feature C. Therefore, each Product Characteristic will have a unique characteristic tag and/or UUID such as <PC030>, <PC031>, and <PC032>.

5.6.13.3 *BOC Example*

Below is the BOC example listing PC Instance Tags, PC instance UUIDs and their PC descriptions for the use case example shown above.

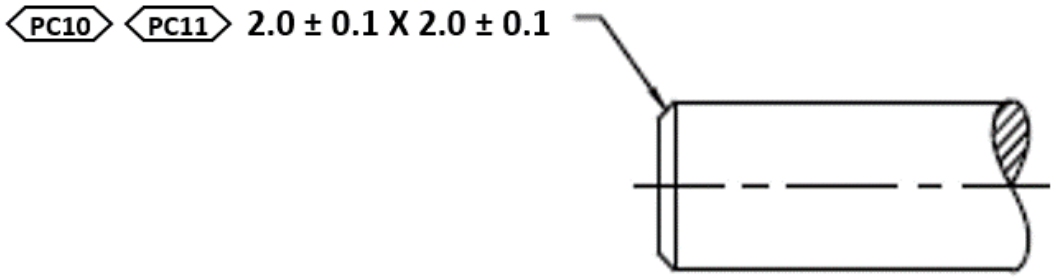
Tag (Base)	Tag (Ext)	UUID (Base)	UUID (Ext)	Description
PC030	N/A	651ADED1-FF04-498A-968E-044147A25064	N/A	[SurfProf|1|A|B|C] on DFeatC
PC031	N/A	9586E974-2AE2-4C3F-9897-9EEFF768E637	N/A	[Para|0.5|A] on DFeatC
PC032	N/A	7D2C2174-3C02-4867-B7B7-B94453015549	N/A	[Flat|0.12] on DFeatC

5.6.14 Chamfer Dimensions

Chamfers may be dimensionally toleranced multiple ways per GD&T standards. Then can also have PC Reference Tags applied in multiple ways.

5.6.14.1 *Example*

Below shows a way for explicitly applying two PC Reference Tags to each facet of a linear direct chamfer tolerance.



5.6.14.2 *Application:*

PC Reference Tag <PC10> applies to the first dimensional facet of the chamfer tolerance and PC Reference Tag <PC11> applies to the last dimensional facet of the chamfer tolerance.

5.6.14.3 *BOC Example*

Tag (Base)	Tag (Ext)	UUID (Primary/PC)	UUID (Ext)	Description
PC010	n/a	1B89A55C-8687-4B0A-B7A0-7555696A3569	n/a	**2.0 ± 0.1 Boss Top to Conical Taper Intersect Boss Side**
PC011	n/a	61A593A5-52D0-4604-B7D5-B969F362425D	n/a	**2.0 ± 0.1 Boss Side to Conical Taper Intersect Boss Top**

5.7 Single Product Characteristic with Defined Multi-Faceted Annotation

A single annotation may have a well-defined multi-faceted annotation containing multiple verification requirements.

5.7.1 Surface Texture with Multiple Elements

This use case is a multi-faceted surface texture annotation, per ASME Y14.36 -1996, which defines multiple requirements via both the symbol modifier identified and the attribute information. The figure below shows some multi-facets from the Surface Texture symbol.

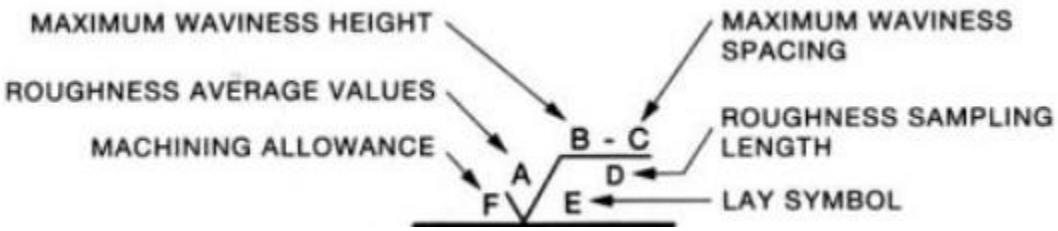


Figure 21: Multi-Faceted Surface Texture Symbol Example

5.7.1.1 *Example*

For this example, multi-faceted surface texture shown below, we have PC Reference Tag <PC898> assigned and the surface texture contains an upper and lower roughness, a maximum waviness height and spacing, a roughness sampling length, and a lay direction symbol.

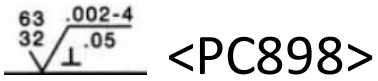


Figure 22: Multi-Faceted Surface Texture Symbol Application

5.7.1.2 *Application:*

In this example, the surface texture specifications annotation is tagged with one PC Reference Tag, <PC898>. The surface-texture symbol annotation's PC Reference Tag <PC898> has multiple faceted applications:

- a surface texture max roughness of 63 as PC898a,
- a surface texture min roughness of 32 as PC898b,
- a surface texture max waviness height of .002 as PC898c,
- a surface texture max waviness spacing of 4 as PC898d,
- a surface texture roundness sampling length of .05 as PC898e, and
- a surface texture lay direction of perpendicular as PC898f.

The PC Instance ID Table for this multi-facet use case is shown below:

PC Base ID	PC Extension ID			PC Instance ID Example
	MFAI	MFRI	RGNI	
PC898	Y	n	n	PC898a

5.7.1.3 *BOC Example*

Tag (Base)	Tag (Ext)	UUID (Base)	UUID (Ext)	Description
PC898	a	587D4300-E2CE-4191-87DA-CE0F09FEDEEB	a	Surface Texture 63 Max Roughness
PC898	b	587D4300-E2CE-4191-87DA-CE0F09FEDEEB	b	Surface Texture 32 Min Roughness

PC898	c	587D4300-E2CE-4191-87DA-CE0F09FEDEEB	c	Surface Texture .002 Max Waviness Height
PC898	d	587D4300-E2CE-4191-87DA-CE0F09FEDEEB	d	Surface Texture 4 Max Waviness Spacing
PC898	e	587D4300-E2CE-4191-87DA-CE0F09FEDEEB	e	Surface Texture .05 Roundness Sampling Length
PC898	f	587D4300-E2CE-4191-87DA-CE0F09FEDEEB	f	Surface Texture Perp. Lay Direction

5.7.2 Thread Spec with Multiple Elements

This use case is a multi-faceted thread specification annotation, which defines multiple elements. The figure below shows some multi-facets from an internal thread specification designation.

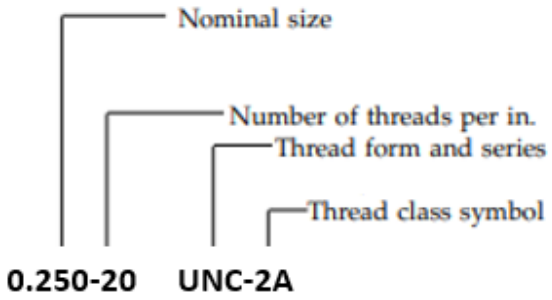


Figure 23: Multi-Faceted Thread Specification Designation Example

5.7.2.1 *Example*

Figure 24 shows a thread specification of .112-40 UNC 2B, a maximum hole drill depth of .350 and a minimum full thread depth of .250, which is positionally located with respect to datum reference frame DBmCm. Furthermore, it is applied to four different threaded holes.

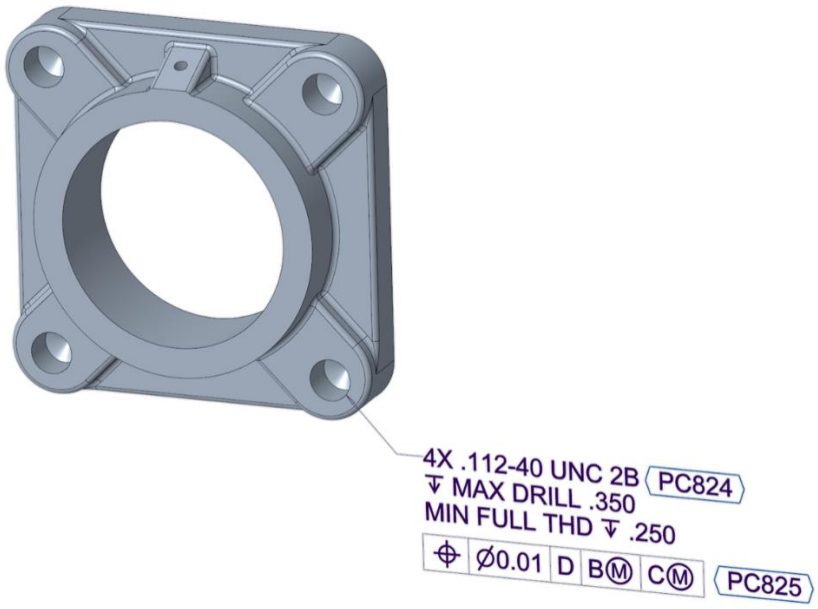


Figure 24: Multi-Facet Thread Specs Applied to Multiple Features

5.7.2.2 *Application:*

In this example, the threaded-hole specifications' annotation is tagged with one PC Reference Tag, <PC824> and the geometric tolerance has one PC Reference Tag, <PC825>. The threaded hole specification annotation's PC Reference Tag <PC824> has multiple-faceted applications: a thread specification of Ø.112-40 UNC 2B as PC824a, a maximum hole drill depth of .350 as PC824b, and a minimum full thread depth of .250 as PC824c. Furthermore, the annotation is repetitively applied four times.

5.7.2.3 *BOC Example*

Below is the BOC example listing PC Instance Tags with their multi-faceted extensions, PC instance UUIDs with their multi-faceted extensions and PC descriptions for the use case example shown above. Furthermore, this BOC included additional extensions for the repetitive features. These use-cases are discussed more within the standard.

Tag (Base)	Tag (Ext)	UUID (Base)	UUID (Ext)	Description
PC824	a.1	35C20045-4E9E-476E-8987-4BAC7F0801A5	a.1	Thread spec .112-40 UNC 2B on Hole 1
PC824	b.1	35C20045-4E9E-476E-8987-4BAC7F0801A5	b.1	Max Drill Depth.350 on Hole 1
PC824	c.1	35C20045-4E9E-476E-8987-4BAC7F0801A5	c.1	Min Full Thd .250 on Hole 1
PC824	a.2	35C20045-4E9E-476E-8987-4BAC7F0801A5	a.2	Thread spec .112-40 UNC 2B on Hole 2
PC824	b.2	35C20045-4E9E-476E-8987-4BAC7F0801A5	b.2	Max Drill Depth.350 on Hole 2
PC824	c.2	35C20045-4E9E-476E-8987-4BAC7F0801A5	c.2	Min Full Thd .250 on Hole 2
PC824	a.3	35C20045-4E9E-476E-8987-4BAC7F0801A5	a.3	Thread spec .112-40 UNC 2B on Hole 3
PC824	b.3	35C20045-4E9E-476E-8987-4BAC7F0801A5	b.3	Max Drill Depth.350 on Hole 3
PC824	c.3	35C20045-4E9E-476E-8987-4BAC7F0801A5	c.3	Min Full Thd .250 on Hole 3
PC824	a.4	35C20045-4E9E-476E-8987-4BAC7F0801A5	a.4	Thread spec .112-40 UNC 2B on Hole 4
PC824	b.4	35C20045-4E9E-476E-8987-4BAC7F0801A5	b.4	Max Drill Depth.350 on Hole 4
PC824	c.4	35C20045-4E9E-476E-8987-4BAC7F0801A5	c.4	Min Full Thd .250 on Hole 4
PC825	1	1F45A49C-53CA-444C-8AC0-28C4655B52D0	1	PosTol w.r.t. DBmCm on Hole 1
PC825	2	1F45A49C-53CA-444C-8AC0-28C4655B52D0	2	PosTol w.r.t. DBmCm on Hole 2
PC825	3	1F45A49C-53CA-444C-8AC0-28C4655B52D0	3	PosTol w.r.t. DBmCm on Hole 3
PC825	4	1F45A49C-53CA-444C-8AC0-28C4655B52D0	4	PosTol w.r.t. DBmCm on Hole 4

5.7.3 Welding Symbol with Multiple Elements

A welding symbol is a graphical representation of the specifications for producing a welded joint. A welding symbol is multi-faceted. The required facets may include the horizontal reference line and arrow. Additional optional facets include the weld symbol and supplementary symbols and information in the tail. Therefore, a welding symbol is a well-defined multi-faceted annotation.

5.7.3.1 *Example*

A welding symbol may have elements that include dimensions of a weld such as the size, length, pitch (center-to-center spacing of the weld), the groove angle, and where the root opening of the weld should be. Where this weld and its dimensions are located depends on the side of the reference line on which the weld symbol is placed.

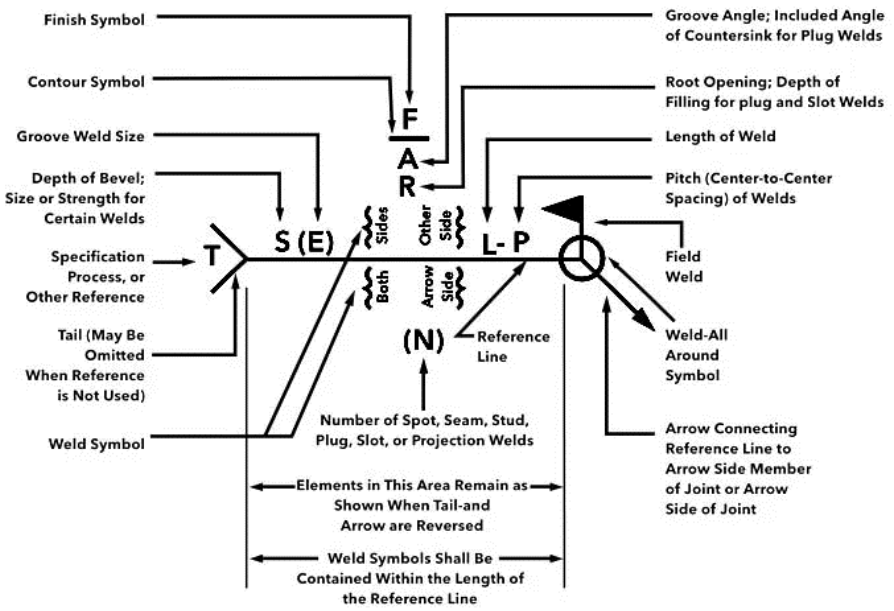


Figure 25: Welding Symbol Elements

The elements of a welding symbol carry a specific kind of information that a symbol conveys. There are elements that may be a part of a typical welding symbol. They are as follows:

- A Groove Angle
- (E) Groove Weld Size
- F Finishing
- L Length of Weld
- P Pitch of Weld (Center-to-Center)
- R Root Opening
- S Size or Depth of Bevel
- T Tail: With or Without Note
- Arrow Side Weld Symbol Type
- Other Side Weld Symbol Type

5.7.3.2 *Application:*

A welding specification will typically be tagged with a single PC Reference Tag, such as <PC899> and then because each welding specification has multiple-faceted applications, there will be multiple PC Instance Tags.

The PC Instance ID Table for this multi-facet use case is shown below:

PC Base ID	PC Extension ID			PC Instance ID Example
	MFAI	MFRI	RGNI	
PC899	Y	n	n	PC899a

5.7.3.3 *BOC Example*

Below is the BOC example listing PC Instance Tags with their multi-faceted extensions, PC instance UUIDs with their multi-faceted extensions and PC descriptions for the use case example shown above.

Tag (Base)	Tag (Ext)	UUID (Base)	UUID (Ext)	Description
PC899	a	E648EDF7-860C-4034-8F16-C7BCE0496D9C	a	Welding Symbol Grove Angle A
PC899	b	E648EDF7-860C-4034-8F16-C7BCE0496D9C	b	Welding Symbol Grove Weld Size (E)
PC899	c	E648EDF7-860C-4034-8F16-C7BCE0496D9C	c	Welding Symbol Finish F
PC899	d	E648EDF7-860C-4034-8F16-C7BCE0496D9C	d	Welding Symbol Length of Weld L
PC899	e	E648EDF7-860C-4034-8F16-C7BCE0496D9C	e	Welding Symbol Pitch of Weld P
PC899	f	E648EDF7-860C-4034-8F16-C7BCE0496D9C	f	Welding Symbol Root Opening R
PC899	g	E648EDF7-860C-4034-8F16-C7BCE0496D9C	g	Welding Symbol Size S
PC899	h	E648EDF7-860C-4034-8F16-C7BCE0496D9C	h	Welding Symbol Tail T
PC899	i	E648EDF7-860C-4034-8F16-C7BCE0496D9C	i	Welding Symbol Arrow Side Weld Type
PC899	j	E648EDF7-860C-4034-8F16-C7BCE0496D9C	j	Welding Symbol Other Side Weld Type

5.8 Single Product Characteristic with Derived Multi-Faceted Annotation

To accommodate some current drafting practices or application limitations, annotations may be multi-faceted, in which multiple verification requirements must be derived. This results in one PC Reference Tag identifying multiple PC Instance Tags for each annotation's multi-facet. Therefore, it is recommended that annotations or annotation segments should be distinct and defined, preferably one PC Instance for each PC Reference. As a result, a MBC preference is for single product characteristics as described in sections 5.6 and 5.7 over single product characteristics as described in this section.

A single annotation may be a multi-faceted annotation containing multiple verification requirements. That is, multiple characteristics may be defined within a single CAD annotation. Typically, this involves holes with depth, counterbores with depths, countersinks with tapers, and threaded holes with thread depth applied to different features. Many use cases are shown below.

5.8.1 Hole with Depth Annotation

This use case is a multi-faceted annotation defining a hole's size tolerance and hole's depth tolerance.

5.8.1.1 *Example*

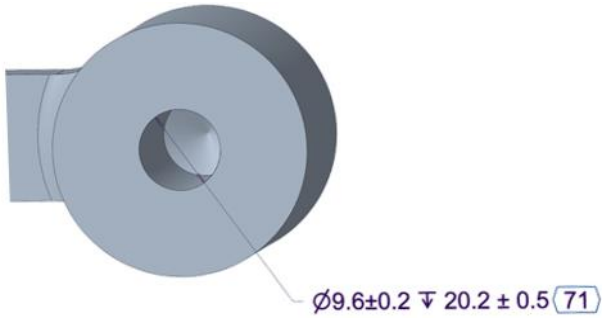


Figure *26* shows a Ø 9.6mm blind hole with diameter size tolerance of +/- 0.2mm and ↧ 20.0mm hole depth of +/- 0.5mm.

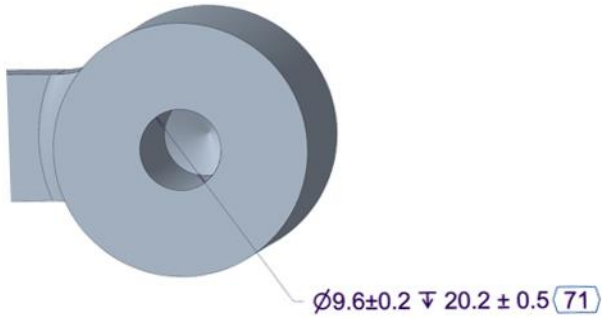


Figure 26: Single PC Reference Tag for Single-line, Multiple Verification Requirement Annotation

5.8.1.2 *Application:*

For this example, the annotation has one Reference Tag, e.g., <71>; however, there are two product characteristics facets to the annotation: 71a for the hole's size product characteristics and 71b for the hole's depth product characteristic. A more explicit representation would require each facet to have its own PC Reference Tag, not shown.

The PC Instance ID Table for this use case is shown below:

PC Base ID	PC Extension ID			PC Instance ID Example
	MFAI	MFRI	RGNI	
71	Y	n	n	71a

5.8.1.3 *BOC Example*

Below is the BOC example listing PC Instance Tags with their multi-faceted extensions, PC instance UUIDs with their multi-faceted extensions and PC descriptions for the use case example shown above.

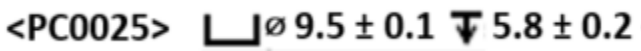
Tag (Base)	Tag (Ext)	UUID (Base)	UUID (Ext)	Description
71	a	661ADED1-FF04-498A-968E-044147A25042	a	+/- 0.2 for 9.6 Hole Diameter
71	b	661ADED1-FF04-498A-968E-044147A25042	b	+/- 0.5 for 20.2 Hole Depth

5.8.2 Counterbore with Depth Annotation

This use case is a multi-facet annotation defining a counterbore size tolerance and counterbore depth tolerance.

5.8.2.1 *Example*

The example is an Ø 9.5 counterbore with diameter size tolerances of +/- 0.1 and ↧ 5.8 counterbore depth of +/- 0.2.



5.8.2.2 *Application:*

For this example, the annotation has one PC Reference Tag, <PC0025>; however, there are two product characteristics facets to the annotation: PC0025.a for the counterbore size product characteristics and PC0025.b for the counterbore depth product characteristic. A more explicit decomposed representation could require each facet to have its own PC Reference Tag, not shown.

The PC Instance ID Table for this use case is shown below:

PC Base ID	PC Extension ID			PC Instance ID Example
	MFAI	MFRI	RGNI	
PC025	Y	n	n	PC025a

5.8.2.3 *BOC Example*

Below is the BOC example listing PC Instance Tags with their multi-faceted extensions, PC instance UUIDs with their multi-faceted extensions and PC descriptions for the use case example shown above.

Tag (Base)	Tag (Ext)	UUID (Base)	UUID (Ext)	Description
PC0025	a	2022AD12-FF25-498A-968E-044147A21225	a	+/- 0.1 for 9.5 Counterbore Diameter
PC0025	b	2022AD12-FF25-498A-968E-044147A21225	b	+/- 0.2 for 5.8 Counterbore Depth

5.8.3 Multi-Counterbore Annotation

This use case is a single annotation defining a multi-faceted annotation containing multiple verification requirements such as a blind-hole's size tolerance with depth tolerance, multiple coaxial counterbore sizes with tolerance and multiple counterbore depth tolerances

5.8.3.1 *Example*

Figure 27 shows a multi counterbore-hole example that has a Ø 4.0 blind hole with ↧13.0 depth, a Ø 8.0 counterbore with ↧10.0 depth and a Ø 12.0 diameter counterbore with ↧ 3.0 depth.

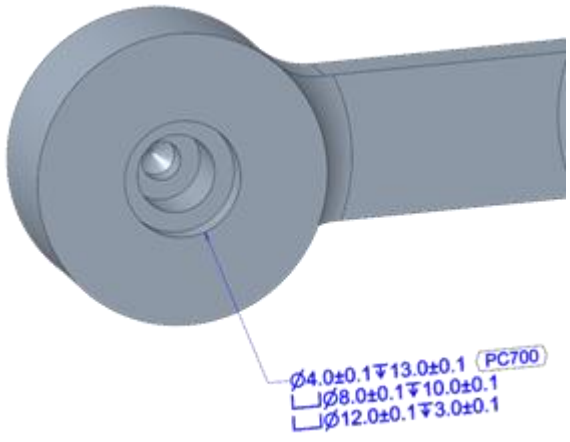


Figure 27: Multi-Counterbore Annotation

5.8.3.2 *Application:*

For this example, the annotation has only one PC Reference Tag, e.g., <PC700>. However, there are six product characteristics facets from this annotation:

- PC700a.1 for the Ø 4.0 hole's size product characteristics and PC700b.1 for the hole's ↧ 13.0 depth product characteristic.
- PC700a.2 for the Ø 8.0 counterbore's size product characteristics and PC700b.2 for the counterbore's ↧ 10.0 depth product characteristic.
- PC700a.3 for the Ø 12.0 counterbore's size product characteristics and PC700b.3 for the counterbore's ↧ 3.0 depth product characteristic.

A more explicit representation would require each facet to have its own PC Reference Tag, not shown.

The PC Instance ID Table for this use case is shown below:

PC Base ID	PC Extension ID			PC Instance ID Example
	MFAI	MFRI	RGNI	
PC700	Y	Y as FRSN	n	PC700a.1

5.8.3.3 *BOC Example*

Below is the BOC example listing PC Instance Tags with their multi-faceted extensions, PC instance UUIDs with their multi-faceted extensions and PC descriptions for the use case example shown above.

Tag (Base)	Tag (Ext)	UUID (Base)	UUID (Ext)	Description
PC700	a.1	3f2e8414-5074-48c9-976f-0b890cefcd00	a.1	± 0.1 for Ø 4.0 Diameter
PC700	b.1	3f2e8414-5074-48c9-976f-0b890cefcd00	b.1	± 0.1 for ↧ 13.0 Hole Depth
PC700	a.2	3f2e8414-5074-48c9-976f-0b890cefcd00	a.2	± 0.1 for Ø 8.0 Counterbore Diameter
PC700	b.2	3f2e8414-5074-48c9-976f-0b890cefcd00	b.2	± 0.1 for ↧ 10.0 Counterbore Depth
PC700	a.3	3f2e8414-5074-48c9-976f-0b890cefcd00	a.3	± 0.1 for Ø 12.0 Counterbore Diameter
PC700	b.3	3f2e8414-5074-48c9-976f-0b890cefcd00	b.3	± 0.1 for ↧ 3.0 Counterbore Depth

5.8.4 Hole with Countersink with Taper & Depth Annotation

This use case is a multi-faceted annotation defining blind hole with depth and a coaxial countersink with conical taper.

5.8.4.1 *Example*

Figure 28 shows a Blind Ø.257 hole with ↧ .465 depth and an Ø.300 countersink with an 82° conical taper, which is positioned with respect to datum reference frame ABC.

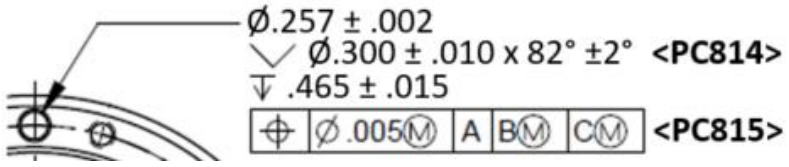


Figure 28: Multi-Facet Dimensional Tolerance

5.8.4.2 *Application:*

For this example, the dimensional annotations are tagged with one Reference Tag, <PC814> and the geometric tolerance has one PC Reference Tag <PC815>. The dimensional annotation's PC Reference Tag <PC814> has four multiple faceted applications (i.e., Hole's Diameter, Countersink's Diameter, Conical Taper, Hole's Depth) and applies to three different features (i.e., Blind-Hole, Countersink, Hole Bottom).

The PC Instance ID Table for <PC814> multi-facet annotation and multiple featured countersink use case is shown below:

PC Base ID	PC Extension ID			PC Instance ID Example
	MFAI	MFRI	RGNI	
PC814	Y	Y as FRSN	n	PC814a.1

The instances of <PC814> are:

Facet	On Feature	PC Instance ID
HoleDiam Ø .257 ±.002	Hole	PC814a.1
CSinkDiam Ø.300 ±.010	Countersink	PC814b.2
ConicalTaper 82° ±2°	Countersink	PC814c.2
HoleDepth ↧.465±.015	Hole Bottom	PC814d.3

5.8.4.3 *BOC Example*

Below is the BOC example listing PC Instance Tags with their multi-faceted extensions, PC instance UUIDs with their multi-faceted extensions and PC descriptions for the use case example shown above.

Tag (Base)	Tag (Ext)	UUID (Base)	UUID (Ext)	Description
PC814	a.1	E4630F20-0BC2-43CC-9B7E-1D0DFC1325B9	a.1	Hole of Ø .257 ±.002
PC814	b.2	E4630F20-0BC2-43CC-9B7E-1D0DFC1325B9	b.2	Countersink Ø of .300 ±.010
PC814	c.2	E4630F20-0BC2-43CC-9B7E-1D0DFC1325B9	c.2	Countersink hole conical taper 82° ±2°
PC814	d.3	E4630F20-0BC2-43CC-9B7E-1D0DFC1325B9	d.3	Hole Bottom at depth ↧.465±.015
PC815	N/A	3F424A5B-8DF7-43F2-BC12-90F5127FDE68	N/A	Positional Tolerance w.r.t. A, Bm, Cm

5.8.5 Composite Tolerance as Single Feature Control Frame

Composite tolerances (e.g., Composite Position, Composite Profile) are a tolerance treated by the standards with a composite feature control frame containing a single geometric symbol (e.g., Position, Profile) followed by two or more segments, each containing tolerance and any required datum references, one above the another. Because a composite tolerance contains two or more segments, it may be treated as a multi-faceted annotation. (The preferred non-multi-faceted approach is described in section 5.6.11.)

5.8.5.1 *Example*

This example shows a two-segmented, composite positional tolerance with a single PC Reference Tag of <PC044> that is multi-faceted as two PC Instance Tags.

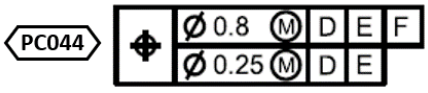


Figure 29: Composite Position Tolerance with Single Reference Tag Example

5.8.5.2 *Application:*

For the composite tolerance example, the tolerance is considered as one Product Characteristic that has multiple segments, which are applied within a common framework. Therefore, there is one common PC Reference Tag <PC044> for the framework, which is extended by a multi-facet identifier for each composite segment such as PC044a and PC044b.

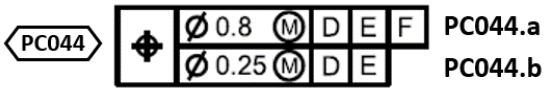


Figure 30: Composite Position Tolerance with Single Reference Tag Application

The PC Instance ID Table for a multi-facet use case is shown below:

PC Base ID	PC Extension ID			PC Instance ID Example
	MFAI	MFRI	RGNI	
PC044	Y	n	n	PC044a

5.8.5.3 *BOC Example*

Tag (Base)	Tag (Ext)	UUID (Base)	UUID (Ext)	Description
PC044	a	2c97b85c-ab1f-11ec-b909-0242ac120002	a	[CPos|0.8m|D|E|F] on Feat1
PC044	b	2c97b85c-ab1f-11ec-b909-0242ac120002	b	[CPos|0.25m|D|E] on Feat1

5.8.6 Group General Note:

A Group General Note displays multiple notes as a single annotation with a single PC Reference Tag. An Individual General Notes example is described in section 5.6.5

5.8.6.1 *Example*

A computer-aided drafting system may represent general notes as a single group general note annotation that is formatted to display five general notes in which general note #4 has been previously deleted.

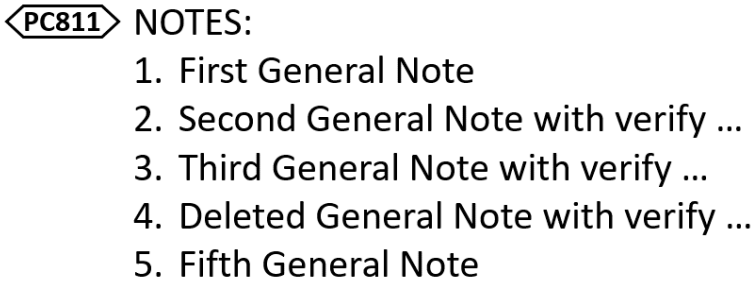


Figure 31: Group General Note Example

5.8.6.2 *Application:*

For this example, the annotation is reference tagged once, e.g., <PC811>, and then each displayed note that is a verification requirement – Note 2, 3, & 4 - will have an instance

identification using the note's sequence number. If the overall general note block has a single PC Reference Tag assigned, then the individual note index number should be used as the multiple feature reference identifier. For this example, this would be PC811.2, PC811.3, and PC811.4. Likewise, the referenced UUID will be the same with extended sequence numbering instance, (example: 651ADED1-FF04-498A-968E-044147A25061.1) for each.

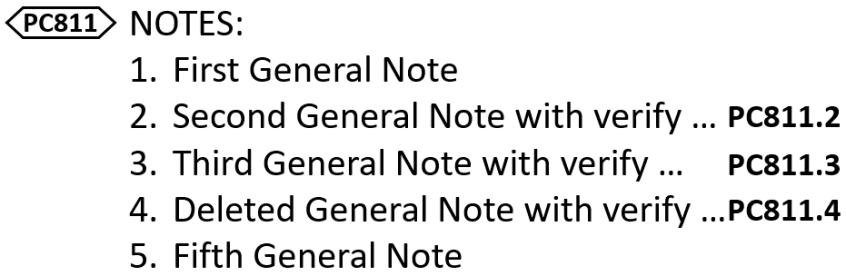


Figure 32: Group General Note Application

The PC Instance ID Table for this multiple feature use case is shown below:

PC Base ID	PC Extension ID			PC Instance ID Example
	MFAI	MFRI	RGNI	
PC811	n	Y	n	PC811.2

5.8.6.3 *BOC Example*

Below is the BOC example listing PC Instance Tags (base +extension) using their general note's sequence numbering extensions, PC instance UUIDs (base +extension) using their general note's sequence numbering extensions and their PC descriptions for the use case example shown above.

Tag (Base)	Tag (Ext)	UUID (Base)	UUID (Ext)	Description
PC811	2	651ADED1-FF04-498A-968E-044147A25061	2	Second General Note
PC811	3	651ADED1-FF04-498A-968E-044147A25061	3	Third General Note
PC811	4	651ADED1-FF04-498A-968E-044147A25061	4	Deleted General Note

Notice that PC Instance Tag PC811.4 is still maintained event though it has been deleted. This helps maintain integrity of the PC Instance Tags and helps prevent reuse of tags.

5.8.7 Chamfer of 45° with Tolerances Annotation Note

Chamfers may be dimensionally toleranced multiple ways per GD&T standards. Two approaches commonly allowed by drafting standards for 45° chamfers become problematic as they introduce ambiguity to how they are explicitly applied. As a result, the ambiguity must be addressed.

Informative Note: Some common drafting practices help the drafting of the annotation, however these practices often lead to ambiguity for the interpreter of the annotation on how explicitly the tolerances are to be applied and to which feature. As a result, for model-based

applications, the following example drafting practice should be deprecated in favor of more explicit alternative annotation approaches.

5.8.7.1 *Example*

Three examples of chamfer annotations are shown below. Figure 33 shows two examples of annotating the most common 45°chamfer tolerance either by a two linear dimensions annotation or by a linear dimension with an angle dimension. Figure 34 shows two dimensional tolerance annotations that convey a chamfer tolerance.

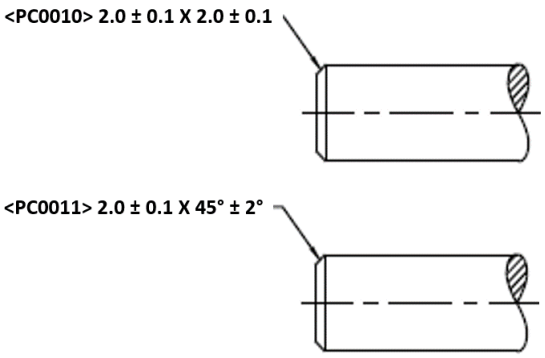


Figure 33: 45° Chamfer Tolerance Note

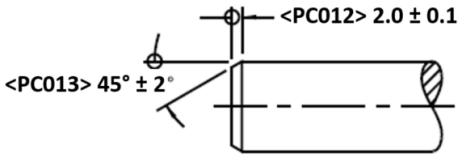


Figure 34: Chamfer Tolerance

5.8.7.2 *Application:*

Figure 33 illustrates two different multi-faceted approaches for dimensioning and tolerancing a 45° chamfer. The top illustration shows a PC Reference Tag <PC0010> being applied to the 2.0 ± 0.1 x 2.0 ± 0.1 annotation and PC Reference Tag <PC0011> applied to the 2.0 ± 0.1 x 45° ±2° annotation. Each approach contains two measurable components and therefore requires two Product Characteristic instances, which results in Instance Tags of PC0010a and PC0010b for the upper part of Figure 33 and PC0011a and PC0011b for the lower part, respectively.

The PC Instance ID Table for this multi-facet use case is shown below:

PC Base ID	PC Extension ID			PC Instance ID Example
	MFAI	MFRI	RGNI	
PC010	Y	n	n	PC010a

Figure 34 shows the preferred way of using two annotations to accommodate the chamfer and thus yield two explicit PC Reference Tags, <PC012> and <PC013>.

5.8.7.3 *BOC Example*

An example BoC for all three approaches with both tags and UUIDs are shown in the table below.

Tag (Base)	Tag (Ext)	UUID (Primary/PC)	UUID (Ext)	Description
PC010	a	101ADED1-FF04-498A-968E-044147A25064	a	**2.0 ± 0.1 Boss Top to Conical Taper Intersect Boss Side**
PC010	b	101ADED1-FF04-498A-968E-044147A25064	b	**2.0 ± 0.1 Boss Side to Conical Taper Intersect Boss Top**
PC011	a	111ADED1-FF04-498A-968E-044147A25064	a	**2.0 ± 0.1 Boss Top to Conical Taper Intersect Boss Side**
PC011	b	1186E974-2AE2-4C3F-9897-9EEFF768E637	b	**45° ± 2° Boss Side to Conical Taper Intersect Boss Top**
PC012	N/A	121ADED1-FF04-498A-968E-044147A25064	N/A	**2.0 ± 0.1 Boss Top to Conical Taper Intersect Boss Side**
PC013	N/A	1386E974-2AE2-4C3F-9897-9EEFF768E637	N/A	**45° ± 2° Boss Side to Conical Taper Intersect Boss Top**

5.8.8 Per Unit Basis

Flatness may be applied on a per unit basis as a means of limiting an abrupt surface variation within a relatively small area of the feature. The unit variation is used either in combination with a specified total variation or alone.

5.8.8.1 *Example*

This example shows a composite like flatness tolerance that has two segments of application. The top segment is an overall total flatness applied to the feature, whereas the bottom segment is a unit variation flatness applied by a square or circular per unit area.

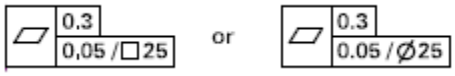


Figure 35: Form with Per Unit Basis

5.8.8.2 *Application:*

The PC Reference Tag <PC080> was applied to the flatness annotation with total and per unit basis annotation. The PC Instance Tag for the total flatness is PC080a and the Instance Tag for the per unit variation flatness is PC080b.

The PC Instance ID Table for this multi-facet use case is shown below:

PC Base ID	PC Extension ID			PC Instance ID Example
	MFAI	MFRI	RGNI	
PC080	Y	n	n	PC080a

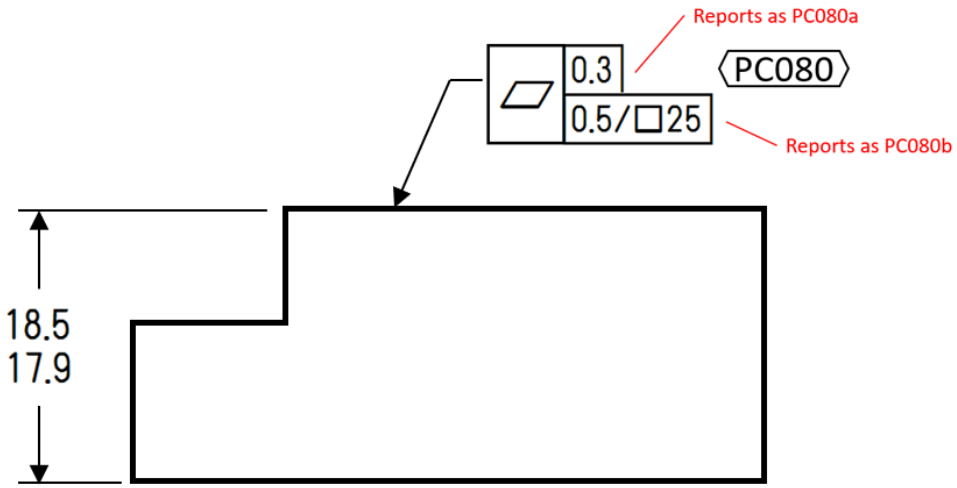


Figure 36: Form with Per Unit Basis with Reference and Instance Tags

5.8.8.3 *BOC Example*

Tag (Base)	Tag (Ext)	UUID (Base)	UUID (Ext)	Description
PC080	a	651ADED1-FF04-498A-968E-044147A25064	a	**[Flat|0.3]**
PC080	b	651ADED1-FF04-498A-968E-044147A25064	b	**[Flat|0.05 / □ 25]**

5.8.9 Mixed Multi-Faceted Feature Control Frame

5.8.9.1 *Example*

This example shows a three-segment feature control frame that is recognized as one PC Reference Tag, <PC062> but represented as a multi-facet annotation. The top segment is a surface profile tolerance with respect to datum reference frame ABC, the middle segment is a parallel tolerance with respect to datum reference frame A, and the bottom segment is a flatness tolerance.

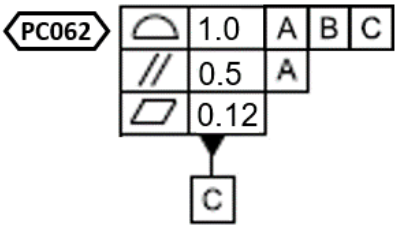


Figure 37: Mixed Multiple Segment Feature Control Frame

See section 5.6.13 for a mixed multiple segment feature control frame with multiple PC Reference Tags.

5.8.9.2 *Application:*

The PC Reference Tag <PC062> is assigned to an annotation that has multiple segments, which are represented as multiple facets. Therefore, each facet will have a unique PC Instance Tag such as PC062a, PC062b, and PC062c.

The PC Instance ID Table for this multi-facet use case is shown below:

PC Base ID	PC Extension ID			PC Instance ID Example
	MFAI	MFRI	RGNI	
PC062	Y	n	n	PC062a

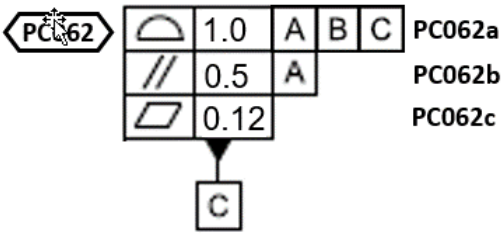


Figure 38: Mixed Multiple Segment Feature Control Frame

5.8.9.3 *BOC Example*

Below is the BOC example listing PC Instance Tags, PC instance UUIDs and their PC descriptions for the use case example shown above.

Tag (Base)	Tag (Ext)	UUID (Base)	UUID (Ext)	Description
PC062	a	222513DC-C4A4-4B3F-801E-6A7F8E790B1B	a	[SurfProf|1|A|B|C] on DFeatC
PC062	b	222513DC-C4A4-4B3F-801E-6A7F8E790B1B	b	[Para|0.5|A] on DFeatC
PC062	c	222513DC-C4A4-4B3F-801E-6A7F8E790B1B	c	[Flat|0.12] on DFeatC

5.9 Single Product Characteristic with Multiple Feature References Applications

This section addresses a single PC Reference Tag, which has multiple feature applications requiring multiple PC Instance Tags for each feature reference.

5.9.1 Dimensional Tolerance with Multiple Feature References

Repetitive feature references for a single dimensional tolerance annotation may be specified by the use of an “X” in conjunction with a numeral to indicate the number of instances applied. (e.g., 3X R3.0 MAX).

5.9.1.1 *Example*

Figure 39 shows an example of a repetitive feature of size tolerance applied to eight holes with a single PC Reference Tag <PC008>.

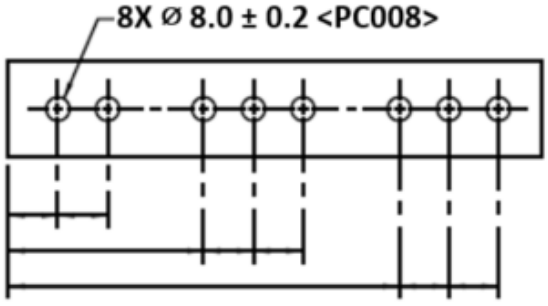


Figure 39: Repetitive Feature Size Tolerance with Reference Tag

5.9.1.2 *Application (Feature Reference Tags):*

Figure 40 shows an example of eight different product characteristics Instance Tags using the Feature Reference Tag approach for the single size tolerance annotation that is applied to eight holes in Figure 39.

The PC Instance ID Table for this multiple feature use case is shown below:

PC Base ID	PC Extension ID			PC Instance ID Example
	MFAI	MFRI	RGNI	
PC008	n	Y as FRT	n	PC008.518

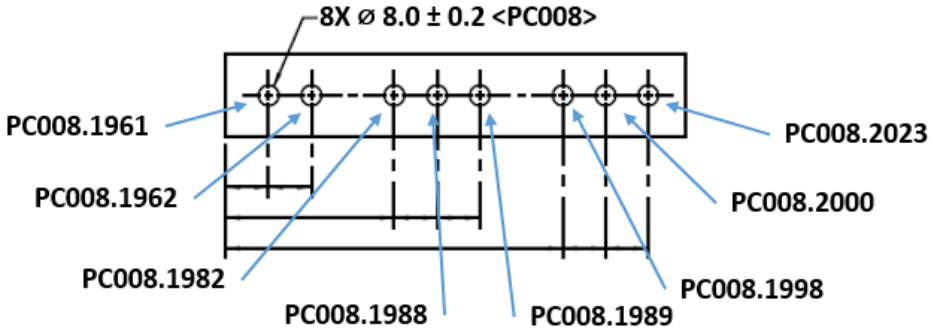


Figure 40: Repetitive Feature Size Tolerance with PC Instance Tags using Feature Reference Tag

For the FeatureReferenceTag example, the 8X ⌀ 8.0 ± 0.2 annotation has a PC Reference Tag of <PC008>, and then the PC Instance Tag would use the PC Base Tag of "PC008" and a FeatureReferenceTag for each feature reference instance could be:

- PC008.1961,
- PC008.1962,
- PC008.1982,
- PC008.1988,
- PC008.1989,
- PC008.1998,
- PC008.2000,
- PC008.2023.

5.9.1.3 *BOC Example (Feature Reference Tags)*

Below is the BOC example listing PC Instance Tags with their feature tag extensions, PC instance UUIDs with their feature tag extensions and their PC descriptions for the use case example shown above.

Tag (Base)	Tag (Ext)	UUID (Base)	UUID (Ext)	Description
PC008	1961	651ADED1-FF04-498A-968E-044147A25068	1961	SizeTol +/- 0.02 on 8.0dFeat1
PC008	1962	651ADED1-FF04-498A-968E-044147A25068	1962	SizeTol +/- 0.02 on 8.0dFeat2
PC008	1982	651ADED1-FF04-498A-968E-044147A25068	1982	SizeTol +/- 0.02 on 8.0dFeat3

PC008	1988	651ADED1-FF04-498A-968E-044147A25068	1988	SizeTol +/- 0.02 on 8.0dFeat4
PC008	1989	651ADED1-FF04-498A-968E-044147A25068	1989	SizeTol +/- 0.02 on 8.0dFeat5
PC008	1998	651ADED1-FF04-498A-968E-044147A25068	1998	SizeTol +/- 0.02 on 8.0dFeat6
PC008	2000	651ADED1-FF04-498A-968E-044147A25068	2000	SizeTol +/- 0.02 on 8.0dFeat7
PC008	2023	651ADED1-FF04-498A-968E-044147A25068	2023	SizeTol +/- 0.02 on 8.0dFeat8

5.9.1.4 *Application (Feature Reference UUIDs):*

Figure 41 shows an example of eight different PC Instance Tags using the feature UUID approach for the single size tolerance annotation that is applied to eight holes in Figure 39.

The PC Instance ID Table for this multiple feature use case is shown below:

PC Base ID	PC Extension ID			PC Instance ID Example
	MFAI	MFRI	RGNI	
PC008	n	Y as FRU	n	PC008.651ADED1-FF04-498A-968E-044147A25064

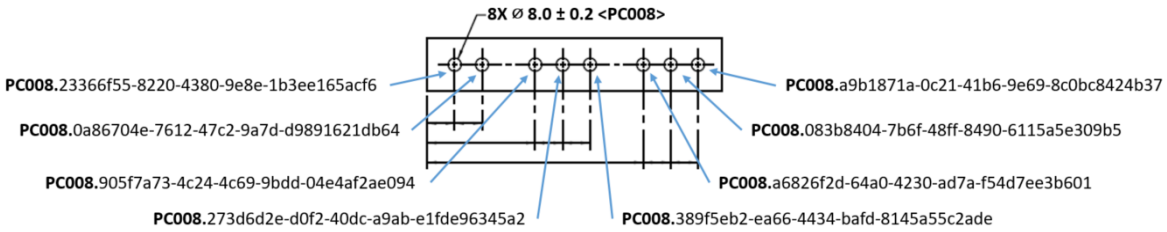


Figure 41: Repetitive Feature Size Tolerance with Instance Tags using Feature Reference UUID

For the FeatureReferenceUUID example, the 8X ⌀ 8.0 ± 0.2 annotation has a PC Reference Tag of <PC008>, and then the PC Instance Tag would use the PC Base Tag of "PC008" and a FeatureReferenceUUID for each feature reference instance could be:

- **PC008.**23366f55-8220-4380-9e8e-1b3ee165acf6,
- **PC008.**0a86704e-7612-47c2-9a7d-d9891621db64,
- **PC008.**905f7a73-4c24-4c69-9bdd-04e4af2ae094,
- **PC008.**273d6d2e-d0f2-40dc-a9ab-e1fde96345a2,
- **PC008.**389f5eb2-ea66-4434-bafd-8145a55c2ade,
- **PC008.**a6826f2d-64a0-4230-ad7a-f54d7ee3b601,
- **PC008.**083b8404-7b6f-48ff-8490-6115a5e309b5,
- **PC008.**a9b1871a-0c21-41b6-9e69-8c0bc8424b37.

5.9.1.5 *BOC Example (Feature Reference UUIDs)*

Below is the BOC example listing PC Instance Tags with their feature UUID extensions, PC instance UUIDs with their feature UUID extensions and their PC descriptions for the use case example shown above.

Tag (Base)	Tag (Ext)	UUID (Base)	UUID (Ext)	Description

PC008	23366f55-8220-4380-9e8e-1b3ee165acf6	651ADED1-FF04-498A-968E-044147A25068	23366f55-8220-4380-9e8e-1b3ee165acf6	SizeTol+/-0.02 on 8.0dFeat1
PC008	0a86704e-7612-47c2-9a7d-d9891621db64	651ADED1-FF04-498A-968E-044147A25068	0a86704e-7612-47c2-9a7d-d9891621db64	SizeTol+/-0.02 on 8.0dFeat2
PC008	905f7a73-4c24-4c69-9bdd-04e4af2ae094	651ADED1-FF04-498A-968E-044147A25068	905f7a73-4c24-4c69-9bdd-04e4af2ae094	SizeTol+/-0.02 on 8.0dFeat3
PC008	273d6d2e-d0f2-40dc-a9ab-e1fde96345a2	651ADED1-FF04-498A-968E-044147A25068	273d6d2e-d0f2-40dc-a9ab-e1fde96345a2	SizeTol+/-0.02 on 8.0dFeat4
PC008	389f5eb2-ea66-4434-bafd-8145a55c2ade	651ADED1-FF04-498A-968E-044147A25068	389f5eb2-ea66-4434-bafd-8145a55c2ade	SizeTol+/-0.02 on 8.0dFeat5
PC008	a6826f2d-64a0-4230-ad7a-f54d7ee3b601	651ADED1-FF04-498A-968E-044147A25068	a6826f2d-64a0-4230-ad7a-f54d7ee3b601	SizeTol+/-0.02 on 8.0dFeat6
PC008	083b8404-7b6f-48ff-8490-6115a5e309b5	651ADED1-FF04-498A-968E-044147A25068	083b8404-7b6f-48ff-8490-6115a5e309b5	SizeTol+/-0.02 on 8.0dFeat7
PC008	a9b1871a-0c21-41b6-9e69-8c0bc8424b37	651ADED1-FF04-498A-968E-044147A25068	a9b1871a-0c21-41b6-9e69-8c0bc8424b37	SizeTol+/-0.02 on 8.0dFeat8

5.9.1.6 *Application (Feature Reference Sequence Number):*

Figure 42 shows an example of eight different PC Instance Tags using the feature sequence numbering approach for the single size tolerance annotation that is applied to eight holes in Figure 39.

The PC Instance ID Table for this multiple feature use case is shown below:

PC Base ID	PC Extension ID			PC Instance ID Example
	MFAI	MFRI	RGNI	
PC008	n	Y as FRSN	n	PC008.1

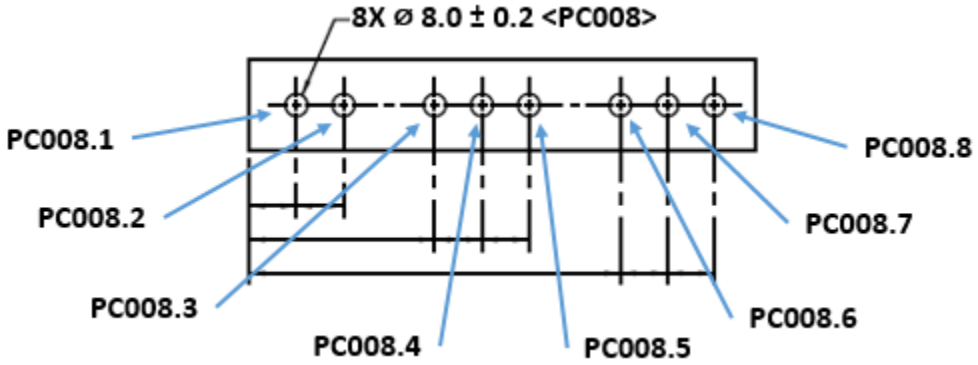


Figure 42: Repetitive Feature Size Tolerance with Instance Tags using Feature Sequencing

For the FeatureReferenceSequenceNumber example, the 8X ⌀ 8.0 ± 0.2 annotation has a PC Reference Tag of <PC008>, and then the PC Instance Tag would use the PC Base Tag of "PC008" and a FeatureReferenceSequenceNumber for each feature reference instance could be:

- PC008.1,
- PC008.2,
- PC008.3,
- PC008.4,
- PC008.5,

- PC008.6,
- PC008.7,
- PC008.8.

5.9.1.7 *BOC Example (Feature Reference Sequence Number)*

Below is the BOC example listing PC Instance Tags with their feature sequence numbering extensions, PC instance UUIDs with their feature sequence numbering extensions and their PC descriptions for the use case example shown above.

Tag (Base)	Tag (Ext)	UUID (Base)	UUID (Ext)	Description
PC008	1	651ADED1-FF04-498A-968E-044147A25068	1	SizeTol+/-0.02 on 8.0dFeat1
PC008	2	651ADED1-FF04-498A-968E-044147A25068	2	SizeTol+/-0.02 on 8.0dFeat2
PC008	3	651ADED1-FF04-498A-968E-044147A25068	3	SizeTol+/-0.02 on 8.0dFeat3
PC008	4	651ADED1-FF04-498A-968E-044147A25068	4	SizeTol+/-0.02 on 8.0dFeat4
PC008	5	651ADED1-FF04-498A-968E-044147A25068	5	SizeTol+/-0.02 on 8.0dFeat5
PC008	6	651ADED1-FF04-498A-968E-044147A25068	6	SizeTol+/-0.02 on 8.0dFeat6
PC008	7	651ADED1-FF04-498A-968E-044147A25068	7	SizeTol+/-0.02 on 8.0dFeat7
PC008	8	651ADED1-FF04-498A-968E-044147A25068	8	SizeTol+/-0.02 on 8.0dFeat8

5.9.2 Geometric Tolerance with Multiple Feature References

Repetitive feature references for a single geometric tolerance annotation may be specified by the use of an “X” in conjunction with a numeral to indicate the number of instances applied.

5.9.2.1 *Example*

Figure *43* shows a profile tolerance applied to two different features via the 2X and multiple leaders.

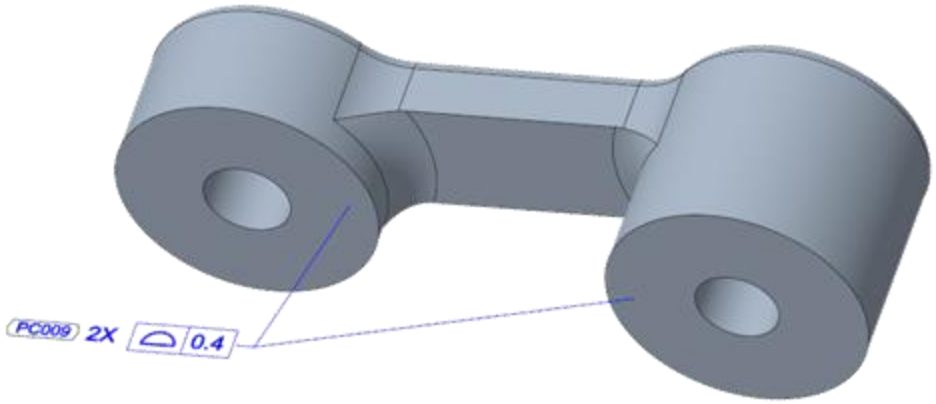


Figure 43: Repetitive Feature Geometric Tolerance

5.9.2.2 *Application:*

For this example, the annotation is tagged once, <PC009>, and then the unique Feature Reference Tag for each feature reference instance could be 2022 and 2023, resulting in PC Instance Tags of PC009.2022 and PC009.2023, respectively.

The PC Instance ID Table for this multiple feature use case is shown below:

PC Base ID	PC Extension ID			PC Instance ID Example
	MFAI	MFRI	RGNI	
PC009	n	Y as FRT	n	PC009.518

5.9.2.3 *BOC Example*

Below is the BOC example listing PC Instance Tags with their repetitive extensions, PC instance UUIDs with their repetitive extensions and PC descriptions for the use case example shown above.

Tag (Base)	Tag (Ext)	UUID (Base)	UUID (Ext)	Description
PC009	2022	0742FBF3-F393-49FE-8370-B6348C387CD5	2022	[SurfProf|0.4] on Feat1
PC009	2023	0742FBF3-F393-49FE-8370-B6348C387CD5	2023	[SurfProf|0.4] on Feat2

5.9.3 Multiple Direct Leader Annotation

Any annotation that has multiple leader lines is considered a multi-faceted annotation.

5.9.3.1 *Multi Leader Geometric Tolerance Annotation*

A geometric tolerance annotation that has multiple leader lines.

5.9.3.1.1 Example (Geometric Tolerance)

Figure 44 shows an example of a surface profile geometric tolerance that contains four leaders, each to a different feature.

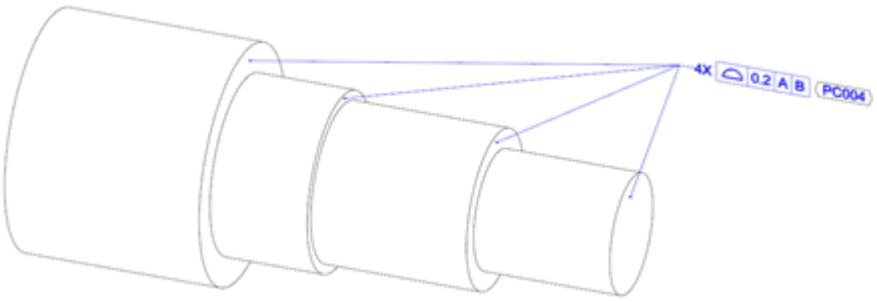


Figure 44: Multiple Direct Leader Geometric Tolerance Annotation

5.9.3.1.2 Application (Geometric Tolerance):

For this example, the annotation's PC Reference Tag is <PC004>, however each leader requires a unique PC Reference Tag for each direct leader instance (for a BoC). The Instance Tags will be PC004.1, PC004.2, PC004.3 and PC004.4.

The PC Instance ID Table for this multiple feature use case is shown below:

PC Base ID	PC Extension ID			PC Instance ID Example
	MFAI	MFRI	RGNI	
PC004	n	Y	n	PC004.1

5.9.3.1.3 BOC Example (Geometric Tolerance)

Below is the BOC example listing PC Instance Tags with their multi-faceted extensions, PC instance UUIDs with their multi-faceted extensions and PC descriptions for the use case example shown above.

Tag (Base)	Tag (Ext)	UUID (Base)	UUID (Ext)	Description
PC004	1	593FF034-23ED-4FC2-B7E2-C6B319506FCF	1	[SurfProf|0.2|A|B] on Feat1
PC004	2	593FF034-23ED-4FC2-B7E2-C6B319506FCF	2	[SurfProf|0.2|A|B] on Feat2
PC004	3	593FF034-23ED-4FC2-B7E2-C6B319506FCF	3	[SurfProf|0.2|A|B] on Feat3
PC004	4	593FF034-23ED-4FC2-B7E2-C6B319506FCF	4	[SurfProf|0.2|A|B] on Feat4

5.9.3.1 *Multi Leader Flag Note Annotation*

A flag note annotation that has multiple leader lines.

5.9.3.1.1 Example (Flag Note)

Example of a flag note <PC0002> that contains four leaders, each to a different geometric element.

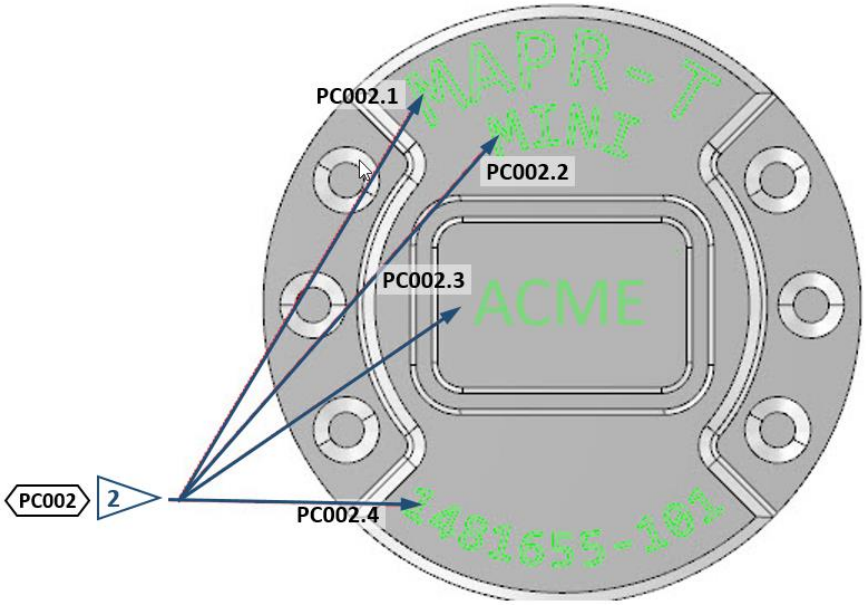


Figure 45: Multiple Direct Leader Flag Note Annotation

5.9.3.1.2 Application (Flag Note):

For this example, the Flag Note annotation has a PC Reference Tag <PC002> and it has four leaders that point to four instances. The PC Instance Tags will be PC002.1, pointing to the “MAPR-T” marking; PC002.2 pointing to the “MINI” marking; PC002.3 pointing to the ”ACME” marking, and PC002.4 pointing to the part number, 1481655-101.

The PC Instance ID Table for this multiple element use case is shown below:

PC Base ID	PC Extension ID	PC Instance ID Example

	MFAI	MFRI	RGNI	
PC002	n	Y	n	PC002.1

5.9.3.1.3 BOC Example (Flag Note)

Below is the BOC example listing PC Instance Tags with their multi-faceted extensions, PC instance UUIDs with their multi-faceted extensions and PC descriptions for the use case example shown above.

Tag (Base)	Tag (Ext)	UUID (Base)	UUID (Ext)	Description
PC002	1	F7D197BF-D87E-4DB5-94E3-793C36D06999	GElem1 UUID	Flag Note on Name
PC002	2	F7D197BF-D87E-4DB5-94E3-793C36D06999	GElem2 UUID	Flag Note on Sub-Name
PC002	3	F7D197BF-D87E-4DB5-94E3-793C36D06999	GElem3 UUID	Flag Note on ACME
PC002	4	F7D197BF-D87E-4DB5-94E3-793C36D06999	GElem4 UUID	Flag Note on Part Number

5.9.4 Tolerance All Around Specification

When a profile tolerance applies all around the true profile of the designated features of the part (in the view in which it is specified), the “all around” symbol is placed on the leader from the feature control frame.

5.9.4.1 *Example*

A surface profile that is applied “all-around” to surfaces all around a feature or group of features, in the view specified.

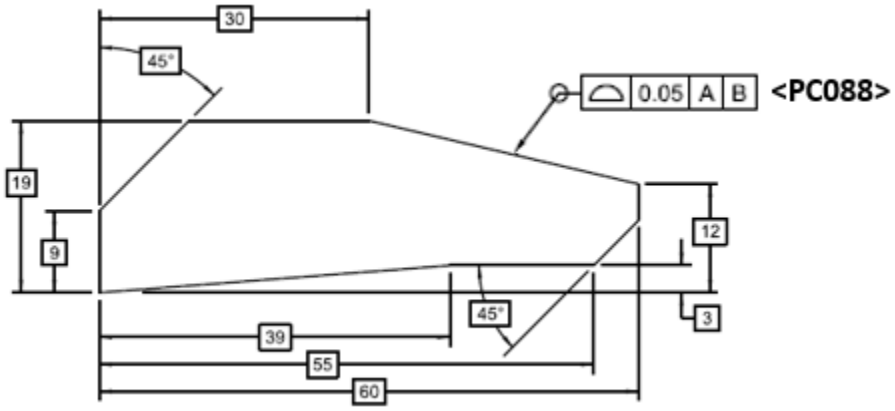


Figure 46: All-Around Profile Tolerance

5.9.4.2 *Application:*

For this example, a profile tolerance applies “All around” involving one profile group feature of eight faces. Each measurement-reporting instance uses a unique PC Instance Tag as shown in the figure below. The PC Instance Tags start as PC088.1 on the face that the <PC088> leader line points to and then are sequenced in a clockwise rotation from PC088.2 through PC088.8.

The PC Instance ID Table for this multiple face use case is shown below:

PC Base ID	PC Extension ID			PC Instance ID Example
	MFAI	MFRI	RGNI	

PC088	n	Y	n	PC088.1

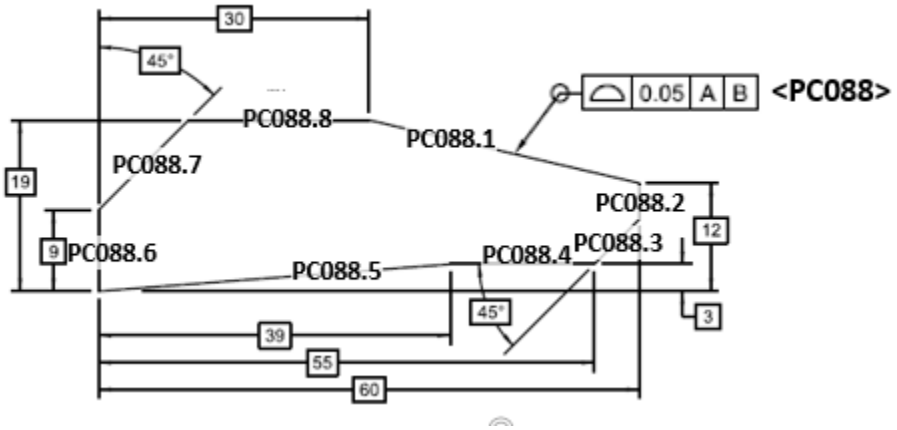


Figure 47: All-Around Profile Tolerance with Instance Tags

5.9.4.3 *BOC Example*

Below is the BOC example listing PC Instance Tags with their multi-faceted extensions, PC instance UUIDs with their multi-faceted extensions and PC descriptions for the use case example shown above.

Tag (Base)	Tag (Ext)	UUID (Base)	UUID (Ext)	Description
PC088	1	EA834B0B-5E68-4993-9434-E1663A634D62	Feat1Ref's UUID	[SurfProf|0.05|A|B] on Feat1
PC088	2	1E75F2BE-9AD6-4727-BDE2-895FE6E2A1C4	Feat2Ref's UUID	[SurfProf|0.05|A|B] on Feat2
PC088	3	E9971332-CE8C-4F14-B246-89B9DC297A76	Feat3Ref's UUID	[SurfProf|0.05|A|B] on Feat3
PC088	4	EE78DFE3-EA17-4A9D-AFEC-D5C7A7CBE63F	Feat4Ref's UUID	[SurfProf|0.05|A|B] on Feat4
PC088	5	30E84E30-20EE-4325-BBE2-5E1BA20AB3F9	Feat5Ref's UUID	[SurfProf|0.05|A|B] on Feat5
PC088	6	5ED33136-6B25-4494-A7DF-1585B131E427	Feat6Ref's UUID	[SurfProf|0.05|A|B] on Feat6
PC088	7	94D99481-6230-4944-BE5E-02FB14DD5BC9	Feat7Ref's UUID	[SurfProf|0.05|A|B] on Feat7
PC088	8	A7CA38A1-F3DA-4B99-A307-07EB20024D8C	Feat8Ref's UUID	[SurfProf|0.05|A|B] on Feat8

5.9.5 Tolerance All Over Specification

When the requirement is that the tolerance applies all over a part, the "all over" symbol may be used. A profile tolerance may be applied all over the three-dimensional profile of a part unless otherwise specified (UOS). It shall be applied in one of the following ways:

(a) Place the "all over" symbol on the leader from the feature control frame

(b) Place the term "ALL OVER" beneath the feature control frame

(c) Place the profile tolerance requirement in the general tolerance block or general notes

5.9.5.1 *Example*

A surface profile that is applied "all-over" to surfaces all over the three-dimensional profile of a part shown.

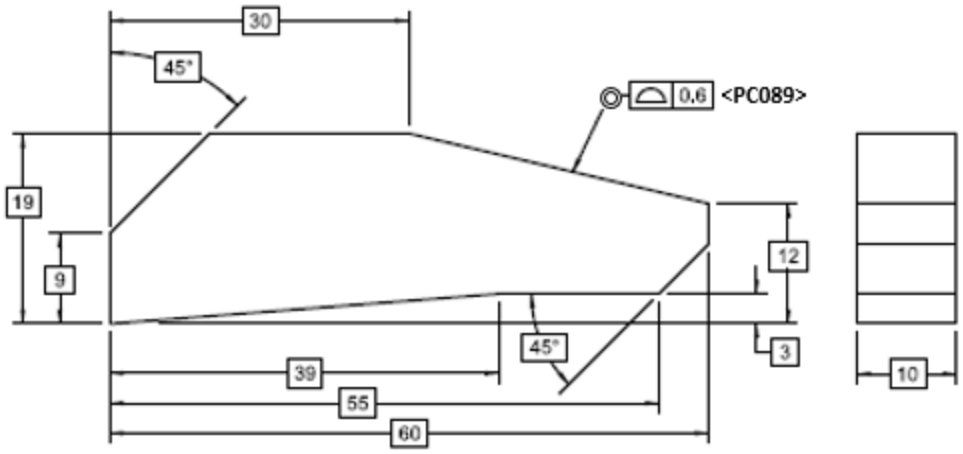


Figure 48: All-Over Profile Tolerance

5.9.5.2 *Application:*

For this example, a profile tolerance applies "All Over" involving one profile group feature of ten faces (including the front and back faces). Each application has a unique Instance Tag as shown in the figure below. The Instance Tags start as PC089.1 on the face that the <PC089> leader line points to and then are sequenced in a clockwise rotation from PC089.2 through PC089.8. The Instance Tags for the front and back are PC089.9 and PC089.10.

The PC Instance ID Table for this multiple face use case is shown below:

PC Base ID	PC Extension ID			PC Instance ID Example
	MFAI	MFRI	RGNI	
PC089	n	Y	n	PC089.1

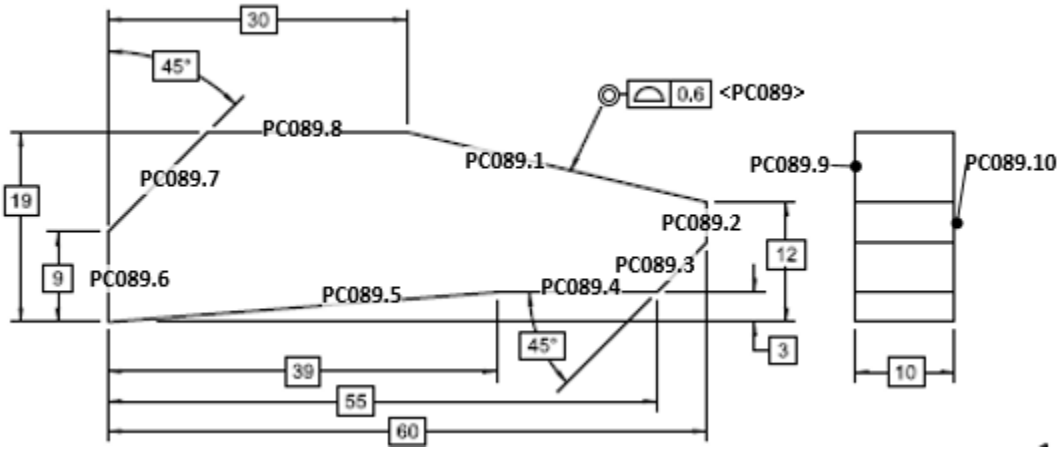


Figure 49: All-Over Profile Tolerance with Instance Tags

5.9.5.3 *BOC Example*

Below is the BOC example listing PC Instance Tags with their multi-faceted extensions, PC instance UUIDs with their multi-faceted extensions and PC descriptions for the use case example shown above.

Tag (Base)	Tag (Ext)	UUID (Base)	UUID (Ext)	Description
PC089	1	29FCEB81-F1F6-41B9-9AF9-AE9961D086ED	Feat1Ref's UUID	[SurfProf|0.6] on Feat1
PC089	2	4969017B-3850-47B1-B3E5-EFC4AEEF8342	Feat2Ref's UUID	[SurfProf|0.6] on Feat2
PC089	3	43DC8D58-D2BD-432F-AC38-5EC34AA55F76	Feat3Ref's UUID	[SurfProf|0.6] on Feat3
PC089	4	8077667B-979D-45D6-9527-25070E131E0B	Feat4Ref's UUID	[SurfProf|0.6] on Feat4
PC089	5	AE9FCD9D-94B2-4A7F-8CD8-8FC068A9BD00	Feat5Ref's UUID	[SurfProf|0.6] on Feat5
PC089	6	0D460D13-030F-4ADB-9FE5-A879DDAE739A	Feat6Ref's UUID	[SurfProf|0.6] on Feat6
PC089	7	0D7EEE3F-7758-48D2-93EF-EBE08C74118E	Feat7Ref's UUID	[SurfProf|0.6] on Feat7
PC089	8	25B27DDF-2E63-44B8-8474-9FA4E4DDFB02	Feat8Ref's UUID	[SurfProf|0.6] on Feat8
PC089	9	987A8BF8-F436-472E-AE2D-A5A5F6B8955D	Feat9Ref's UUID	[SurfProf|0.6] on Feat9
PC089	10	3935710D-3947-48E3-A264-6C357EC1BAD0	Feat10Ref's UUID	[SurfProf|0.6] on Feat10

5.9.6 Unless Otherwise Specified (UOS) Tolerance

The "Unless Otherwise Specified" phrase may be used to specify a default requirement.

5.9.6.1 *Example*

Where a tolerance, such as a profile geometric tolerance, is placed in a general note or within a general tolerance block, the tolerance applies to all features unless otherwise specified. Therefore, the usage of an "Unless Otherwise Specified" (UOS) requirement (e.g., [SurfProf|0.010|A|B|C]) will be a unique Product Characteristic of one unique PC Reference Tag and one unique Reference UUID and be applied to multiple instances.

UNLESS OTHERWISE SPECIFIED, ALL FEATURES ARE [SurfProf|0.1|A|B|C]

5.9.6.2 *Application:*

For this example, first assume that PC Reference Tag <PC099> is applied to the "UNLESS OTHERWISE SPECIFIED, ALL FEATURES ARE [SurfProf|0.1|A|B|C]. Next, let us assume that there are four "UNLESS OTHERWISE SPECIFIED features, having Feature Reference Tags 5, 18, 19, and 61. Then each feature instance extends the PC Base Tag with a Multiple Feature Reference identifier as PC099.5, PC099.18, PC099.19 and PC099.61, respectively.

The PC Instance ID Table for this multiple feature use case is shown below:

PC Base ID	PC Extension ID			PC Instance ID Example
	MFAI	MFRI	RGNI	
PC099	n	Y as FRT	n	PC099.1

5.9.6.3 *BOC Example*

Tag (Base)	Tag (Ext)	UUID (Base)	UUID (Ext)	Description
PC099	5	651ADED1-FF04-498A-968E-044147A2170F	5	SurfProf|0.010|A|B|C] on Feat5
PC099	18	651ADED1-FF04-498A-968E-044147A2170F	18	SurfProf|0.010|A|B|C] on Feat18
PC099	19	651ADED1-FF04-498A-968E-044147A2170F	19	SurfProf|0.010|A|B|C] on Feat19
PC099	61	651ADED1-FF04-498A-968E-044147A2170F	61	SurfProf|0.010|A|B|C] on Feat61

5.9.7 Tolerance Applied Between Points

Per some tolerancing standards, the symbolic means of indicating that a tolerance, or other specification, applies across multiple features or to a limited segment of a feature between designated extremities, may be shown via a “between” annotation.

5.9.7.1 *Example*

For this example, a profile tolerance applies between two designated points.

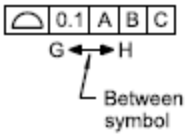


Figure 50: Profile Tolerance between Points Annotation

5.9.7.2 *Application:*

For this example, three profile tolerances are applied between points. The surface profile with PC Reference Tag, <PC016>, applies from C to D, <PC017> applies from B to C, and <PC018> applies from A to B, respectively. Notice that the surface profile tolerance with PC Reference Tag <PC017> has three instances as its application between points B and C includes three faces/features.

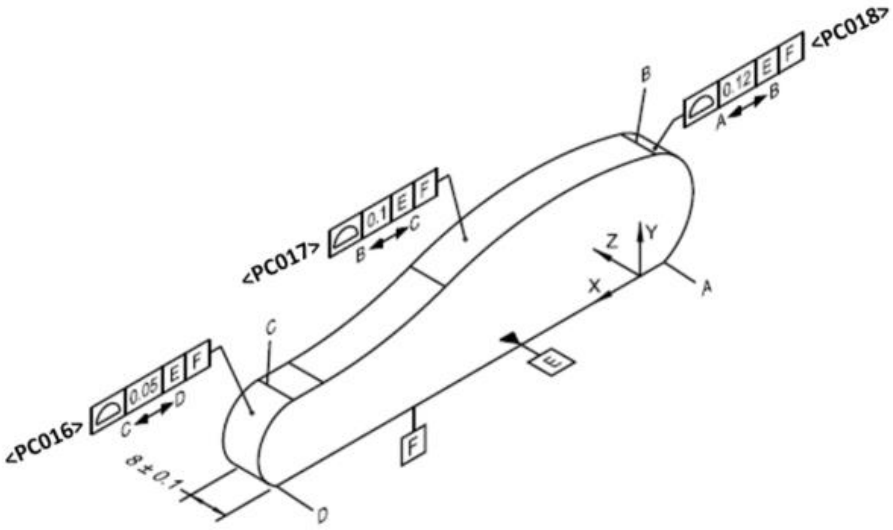


Figure 51: Profile Tolerance between Points Application

The PC Instance ID Table for this multiple face use case is shown below:

PC Base ID	PC Extension ID			PC Instance ID Example
	MFAI	MFRI	RGNI	
PC017	n	Y	n	PC017.1

5.9.7.3 *BOC Example*

The BoC is shown below:

Tag (Base)	Tag (Ext)	UUID (Base)	UUID (Ext)	Description
PC016	N/A	FC236F7A-58D0-47E4-ACE3-C36E2FABD0F7	N/A	SurfProf|0.05|E|F] on Feat5
PC017	6	FC236F7A-58D0-47E4-ACE3-C36E2FABD0F7	6	SurfProf|0.1|E|F] on Feat6
PC017	7	FC236F7A-58D0-47E4-ACE3-C36E2FABD0F7	7	SurfProf|0.1|E|F] on Feat7
PC017	8	FC236F7A-58D0-47E4-ACE3-C36E2FABD0F7	8	SurfProf|0.1|E|F] on Feat8
PC018	N/A	FC236F7A-58D0-47E4-ACE3-C36E2FABD0F7	N/A	SurfProf|0.12|E|F] on Feat9

5.9.8 Simple Bidirectional Position Pattern

This is multiple, two different Bidirectional Positional Tolerancing, using the Rectangular Coordinate Method (ASME Y14.5- 2018: Figure 10-29)

5.9.8.1 *Bidirectional Position, Rectangular Coordinate Method*

5.9.8.1.1 Example

This example shows two bi-directional positional tolerances and a size tolerance that are applied on three different features. The PC Reference Tag for the horizontal bidirectional positional tolerance is <PC711>, the PC Reference Tag for the vertical bidirectional positional tolerance is <PC712>, and the PC Reference Tag for the 16 +0.2-.0 is <PC713>

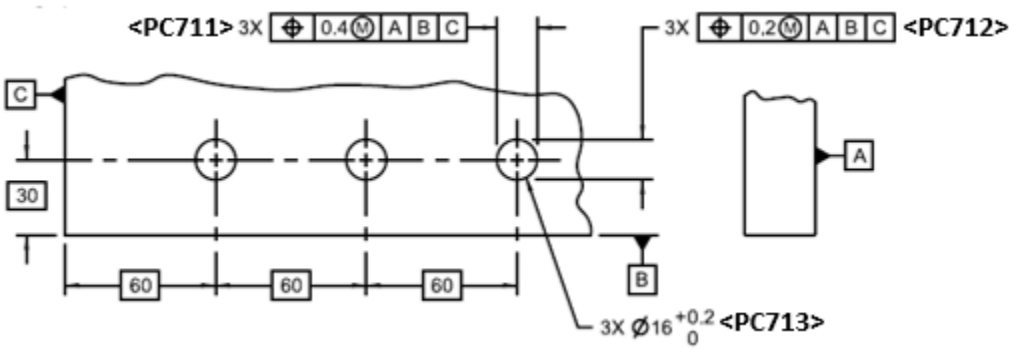


Figure 52: Bidirectional Positional Tolerance, Rectangular Coordinate Method with PC Reference Tags

5.9.8.1.2 Application:

The Instance Tags for the 3X features uses the feature sequence numbering approach. The PC Instance ID Table for this multiple feature use case is shown below:

PC Base ID	PC Extension ID			PC Instance ID Example
	MFAI	MFRI	RGNI	
PC011	n	Y	n	PC011.1

Using the suggestion of starting with the first feature being the one pointed at by the annotation leader and sequencing increased in a clockwise direction.

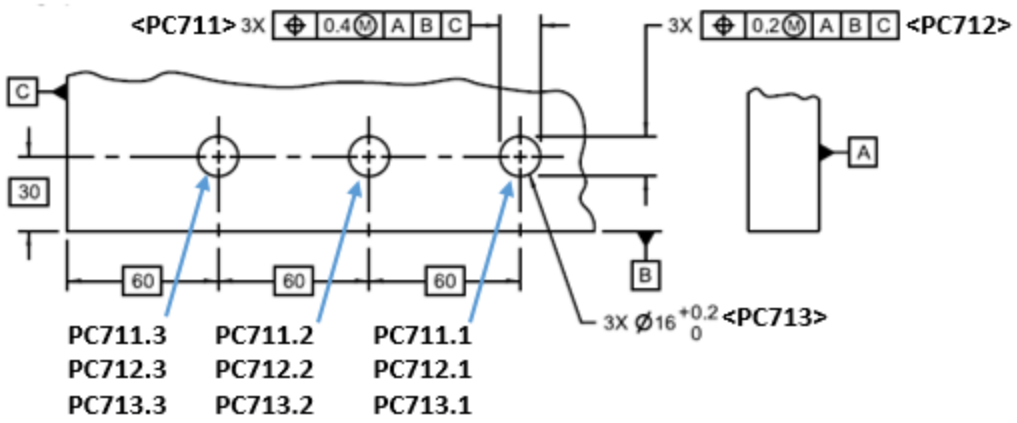


Figure 53: Bidirectional Positional Tolerance, Rectangular Coordinate Method with PC Instance Tags

5.9.8.1.3 BOC Example

Tag (Base)	Tag (Ext)	UUID (Base)	UUID (Ext)	Description
PC711	1	651ADED1-FF04-498A-968E-044147A25064	Feat1Ref's UUID	[BiPos|0.4|A|B|C] on Feat1
PC711	2	9586E974-2AE2-4C3F-9897-9EEFF768E637	Feat2Ref's UUID	[BiPos|0.4|A|B|C] on Feat2
PC711	3	7D2C2174-3C02-4867-B7B7-B94453015549	Feat3Ref's UUID	[BiPos|0.4|A|B|C] on Feat3
PC712	1	651ADED1-FF04-498A-968E-044147A25064	Feat1Ref's UUID	[BiPos|0.2|A|B|C] on Feat1
PC712	2	9586E974-2AE2-4C3F-9897-9EEFF768E637	Feat2Ref's UUID	[BiPos|0.2|A|B|C] on Feat2
PC712	3	7D2C2174-3C02-4867-B7B7-B94453015549	Feat3Ref's UUID	[BiPos|0.2|A|B|C] on Feat3
PC713	1	B76834B0-93AA-45F6-823C-3CC9963B17A7	Feat1Ref's UUID	Diameter 16 +0.2 -0 on Feat1
PC713	2	D97277EB-64A9-4EBF-BAA8-6524407371B0	Feat2Ref's UUID	Diameter 16 +0.2 -0 on Feat2
PC713	3	20266242-7789-4DF9-BC57-D5EE4065D5BA	Feat3Ref's UUID	Diameter 16 +0.2 -0 on Feat3

5.9.8.2 *Bidirectional Position, Polar Coordinate Method*

This is multiple, two different Bidirectional Positional Tolerancing, using the Polar Coordinate Method (ASME Y14.5- 2018: Figure 10-30)

5.9.8.2.1 Example

This example shows two bi-directional positional tolerances with a size and perpendicular tolerance.

5.9.8.2.2 Application:

The PC Reference Tag for the radial bidirectional positional tolerance is <PC715>; the PC Reference Tag for the perpendicular to the line-of-centers bidirectional positional tolerance is <PC716>. The PC Reference Tag for the hole's diameter is <PC713>, and the PC Reference Tag for the perpendicular tolerance is <PC714>.

The PC Instance ID Table for this multiple feature use case is shown below:

PC Base ID	PC Extension ID			PC Instance ID Example
	MFAI	MFRI	RGNI	
PC007	n	Y	n	PC007.1

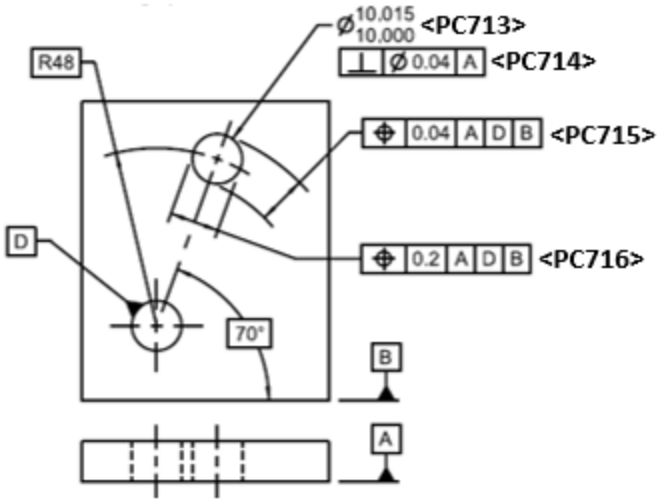


Figure 54: Bidirectional Positional Tolerance, Polar Coordinate Method

5.9.8.2.3 BOC Example

Tag (Base)	Tag (Ext)	UUID (Base)	UUID (Ext)	Description
PC713	N/A	EF1E5DAE-D6DA-420C-BC08-1A83FDC1D169	N/A	SizeTol 10.015/10.000 on Feat1
PC714	N/A	F8E41B75-925D-4609-ABD8-5AD901590FB5	N/A	[Perp|0.04|A] on Feat1
PC715	N/A	CF0FB663-E9D6-4298-8C27-0EAB12167B30	N/A	[BiPos|0.04|A|D|B] on Feat1
PC716	N/A	5694E34C-7F25-4DBE-B801-5E47092579D4	N/A	[BiPos|0.02|A|D|B] on Feat1

5.9.9 Simple Position Pattern

One positional tolerance may be applied to multiple holes and a position tolerance may be applied to a pattern feature of size.

5.9.9.1 *Position Pattern of Holes*

This is a pattern of multiple holes, each with a size (diameter) tolerance and a position tolerance applied to multiple features.

5.9.9.1.1 Example

Example inspired by ASME Y14.5's Cover, shows eight multiple hole features centered about an axis.

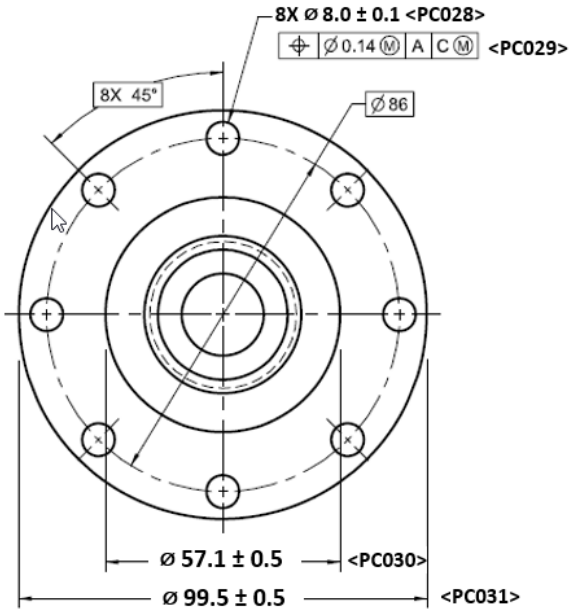


Figure 55: Simple Position Pattern of Holes

5.9.9.1.2 Application:

The example shows eight holes with an individual position tolerance. Each hole has a size tolerance and a positional tolerance. The hole's size annotation has a PC Reference Tag <PC028> and the hole's positional tolerance annotation has a single PC Reference Tag <PC029>. Assuming that the explicit feature reference identifiers are not known, then a feature reference sequence number should be used for each hole occurrence. Starting with sequence number "1", the hole's size Instance Tags would result in PC028.1, PC028.2, PC028.3, PC028.4, PC028.5, PC028.6, PC028.7, and PC028.8. Likewise, using the feature reference sequence number approach, the hole position Instance Tags would result in PC029.1, PC029.2, PC029.3, PC029.4, PC029.5, PC029.6, PC029.7, PC029.8.

The PC Instance ID Table for this multiple feature use case is shown below:

PC Base ID	PC Extension ID			PC Instance ID Example
	MFAI	MFRI	RGNI	
PC028	n	Y	n	PC028.1

Informative Note: The placement of each instance of the 8X holes may be problematic. However, it is recommended that in the primary view of the annotation, starting from the single leader of the 8X annotation, the sequence would proceed in a clockwise direction.

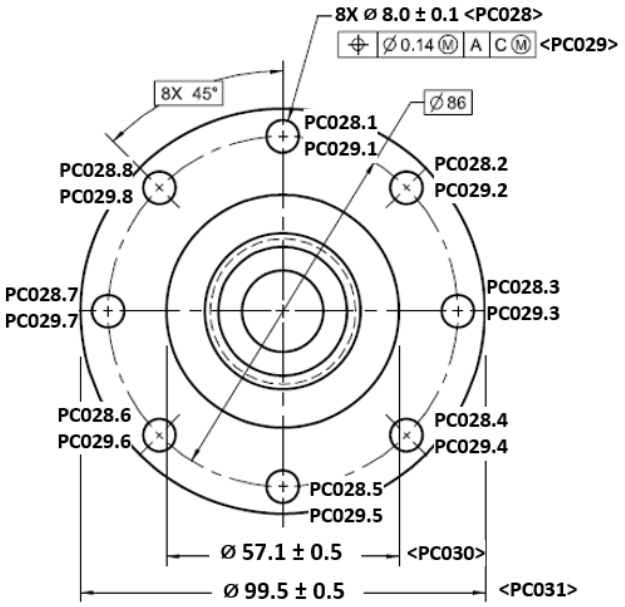


Figure 56: Simple Position of Bolt-Hole Pattern with Feature Instance (informative)

5.9.9.1.3 BOC Example

An example BoC is shown in the table below.

Tag (Base)	Tag (Ext)	UUID (Base)	UUID (Ext)	Description
PC028	1	651ADED1-FF04-498A-968E-044147A25064	1	**8.0 ± 0.1 Size on Hole1**
PC028	2	651ADED1-FF04-498A-968E-044147A25064	2	**8.0 ± 0.1 Size on Hole2**
PC028	3	651ADED1-FF04-498A-968E-044147A25064	3	**8.0 ± 0.1 Size on Hole3**
PC028	4	651ADED1-FF04-498A-968E-044147A25064	4	**8.0 ± 0.1 Size on Hole4**
PC028	5	651ADED1-FF04-498A-968E-044147A25064	5	**8.0 ± 0.1 Size on Hole5**
PC028	6	651ADED1-FF04-498A-968E-044147A25064	6	**8.0 ± 0.1 Size on Hole6**
PC028	7	651ADED1-FF04-498A-968E-044147A25064	7	**8.0 ± 0.1 Size on Hole7**
PC028	8	651ADED1-FF04-498A-968E-044147A25064	8	**8.0 ± 0.1 Size on Hole8**
PC029	1	B6504F34-D5B4-4BD5-86A7-29E35292EF82	1	**[Pos|0.14m|A|Cm] on Hole1**
PC029	2	B6504F34-D5B4-4BD5-86A7-29E35292EF82	2	**[Pos|0.14m|A|Cm] on Hole2**
PC029	3	B6504F34-D5B4-4BD5-86A7-29E35292EF82	3	**[Pos|0.14m|A|Cm] on Hole3**
PC029	4	B6504F34-D5B4-4BD5-86A7-29E35292EF82	4	**[Pos|0.14m|A|Cm] on Hole4**
PC029	5	B6504F34-D5B4-4BD5-86A7-29E35292EF82	5	**[Pos|0.14m|A|Cm] on Hole5**
PC029	6	B6504F34-D5B4-4BD5-86A7-29E35292EF82	6	**[Pos|0.14m|A|Cm] on Hole6**
PC029	7	B6504F34-D5B4-4BD5-86A7-29E35292EF82	7	**[Pos|0.14m|A|Cm] on Hole7**
PC029	8	B6504F34-D5B4-4BD5-86A7-29E35292EF82	8	**[Pos|0.14m|A|Cm] on Hole8**
PC030	N/A	FB94A2F6-F84D-4355-8FA7-221D43798217	N/A	**57.1 ± 0.5 Size on 57.1 Boss**
PC031	N/A	BE37188D-0849-472D-9ECF-76887F09E389	N/A	**99.5 ± 0.5 Size on 99.5 diameter**

5.9.9.2 *Position Pattern of Slots or Tabs*

This is a pattern of multiple slots or tabs, each with a size (width) tolerance and a position tolerance applied to multiple features.

5.9.9.2.1 Example

Example of six slot features centered about an axis.

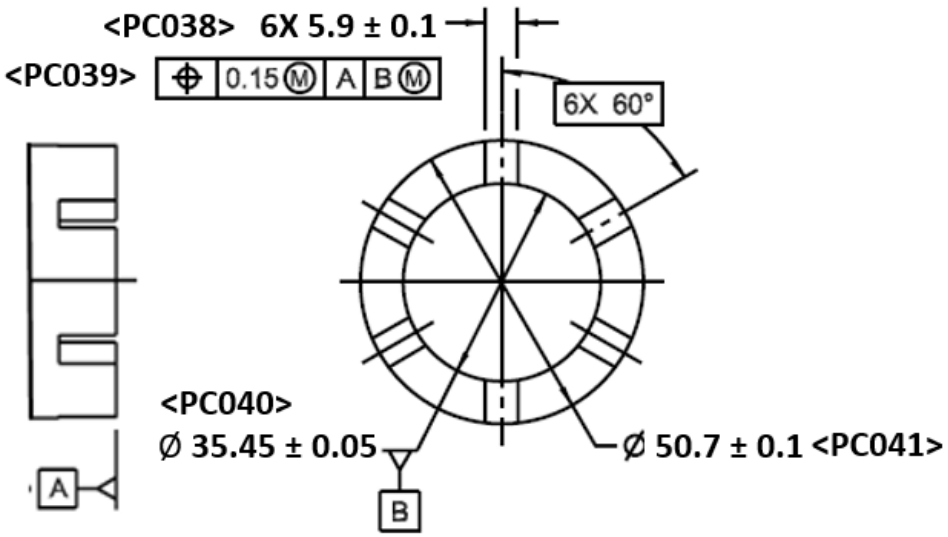


Figure 57: Simple Position Pattern of Slots with PC Reference Tags

5.9.9.2.2 Application:

The example shows a pattern feature with six slots. Each slot has a size tolerance and a positional tolerance. The slot's size annotation has a PC Reference Tag <PC038> and the slot's positional tolerance annotation has a single PC Reference Tag <PC039>.

The PC Instance ID Table for this multiple feature use case is shown below:

PC Base ID	PC Extension ID			PC Instance ID Example
	MFAI	MFRI	RGNI	
PC038	n	Y	n	PC038.1

Assuming that the explicit feature reference identifiers are not known, a FRSN approach should be used for each slot's occurrence. Starting with sequence number "1", the slot's size Instance Tags would result in PC038.1, PC038.2, PC038.3, PC038.4, PC038.5, and PC038.6. Likewise, using a FRSN approach, the slot's position Instance Tags would result in PC039.1, PC039.2, PC039.3, PC039.4, PC039.5, and PC039.6.

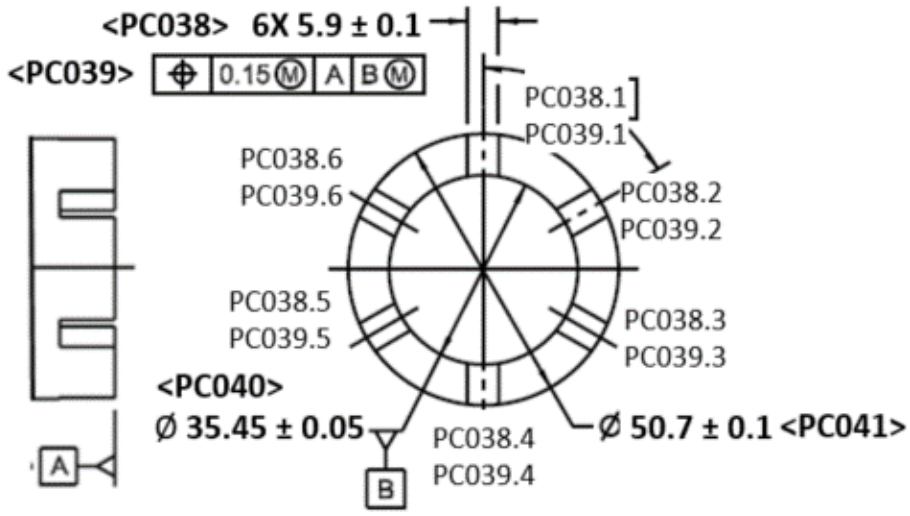


Figure 58: Simple Position Pattern of Slots with PC Instance Tags

5.9.9.2.3 BOC Example

An example BoC is shown in the table below.

Tag (Base)	Tag (Ext)	UUID (Base)	UUID (Ext)	Description
PC038	1	9586E974-2AE2-4C3F-9897-9EEFF768E637	1	**8.0 ± 0.1 Size on Hole1**
PC038	2	9586E974-2AE2-4C3F-9897-9EEFF768E637	2	**8.0 ± 0.1 Size on Hole2**
PC038	3	9586E974-2AE2-4C3F-9897-9EEFF768E637	3	**8.0 ± 0.1 Size on Hole3**
PC038	4	9586E974-2AE2-4C3F-9897-9EEFF768E637	4	**8.0 ± 0.1 Size on Hole4**
PC038	5	9586E974-2AE2-4C3F-9897-9EEFF768E637	5	**8.0 ± 0.1 Size on Hole5**
PC038	6	9586E974-2AE2-4C3F-9897-9EEFF768E637	6	**8.0 ± 0.1 Size on Hole6**
PC038	7	9586E974-2AE2-4C3F-9897-9EEFF768E637	7	**8.0 ± 0.1 Size on Hole7**
PC038	8	9586E974-2AE2-4C3F-9897-9EEFF768E637	8	**8.0 ± 0.1 Size on Hole8**
PC039	1	81F9D931-FCD9-44CE-BF23-B25AB920DFDE	1	**[Pos|0.14m|A|Cm] on Hole1**
PC039	2	81F9D931-FCD9-44CE-BF23-B25AB920DFDE	2	**[Pos|0.14m|A|Cm] on Hole2**
PC039	3	81F9D931-FCD9-44CE-BF23-B25AB920DFDE	3	**[Pos|0.14m|A|Cm] on Hole3**
PC039	4	81F9D931-FCD9-44CE-BF23-B25AB920DFDE	4	**[Pos|0.14m|A|Cm] on Hole4**
PC039	5	81F9D931-FCD9-44CE-BF23-B25AB920DFDE	5	**[Pos|0.14m|A|Cm] on Hole5**
PC039	6	81F9D931-FCD9-44CE-BF23-B25AB920DFDE	6	**[Pos|0.14m|A|Cm] on Hole6**
PC039	7	81F9D931-FCD9-44CE-BF23-B25AB920DFDE	7	**[Pos|0.14m|A|Cm] on Hole7**
PC039	8	81F9D931-FCD9-44CE-BF23-B25AB920DFDE	8	**[Pos|0.14m|A|Cm] on Hole8**
PC040	N/A	FB94A2F6-F84D-4355-8FA7-221D43798217	N/A	**57.1 ± 0.5 Size on 57.1 Boss**
PC041	N/A	BE37188D-0849-472D-9ECF-76887F09E389	N/A	**99.5 ± 0.5 Size on 99.5 diameter**

5.9.9.3 *Composite Position Pattern of Holes*

This is a pattern of multiple holes, each with a size (diameter) tolerance and a composite position tolerance applied to the pattern feature.

5.9.9.3.1 Example

The example shows an eight-hole pattern feature.

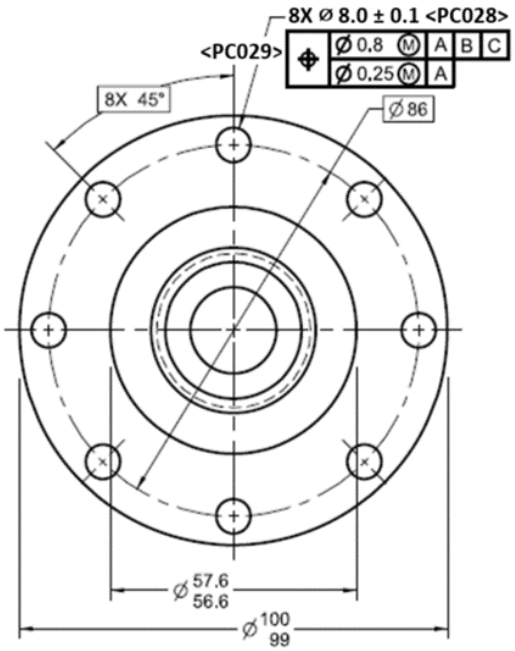


Figure 59: Simple Composite Position of Pattern Feature of Size

5.9.9.3.2 Application

The example shows a diameter tolerance applied to each of the eight holes of the feature pattern and a composite position geometric tolerance applied to the hole-pattern feature. For this example, the holes' size annotation will have a PC Reference Tag <PC028> with PC Reference UUID coupled with extended identifier tags and/or UUID for each hole occurrence. This could result in PC Instance Tags of PC028.1, PC028.2, PC028.3, PC028.4, PC028.5, PC028.6, PC028.7, and PC028.8. Then for the pattern of holes feature's composite position tolerance, there would be a single PC Reference Tag <PC029> with PC Reference UUID.

The PC Instance ID Table for <PC028> multiple feature use case is shown below:

PC Base ID	PC Extension ID			PC Instance ID Example
	MFAI	MFRI	RGNI	
PC028	n	Y	n	PC028.1

The PC Instance ID Table for <PC029> multi-facet use case a pattern feature is shown below:

PC Base ID	PC Extension ID			PC Instance ID Example
	MFAI	MFRI	RGNI	
PC029	Y	n	n	PC029.a

Informative Note: The placement of each instance of the 8X holes for <PC028> may not be explicit, however it is recommended that in the primary view of the

annotation, starting sequence numbering from the single leader feature of the 8X annotation, the sequence would proceed in a clockwise direction.

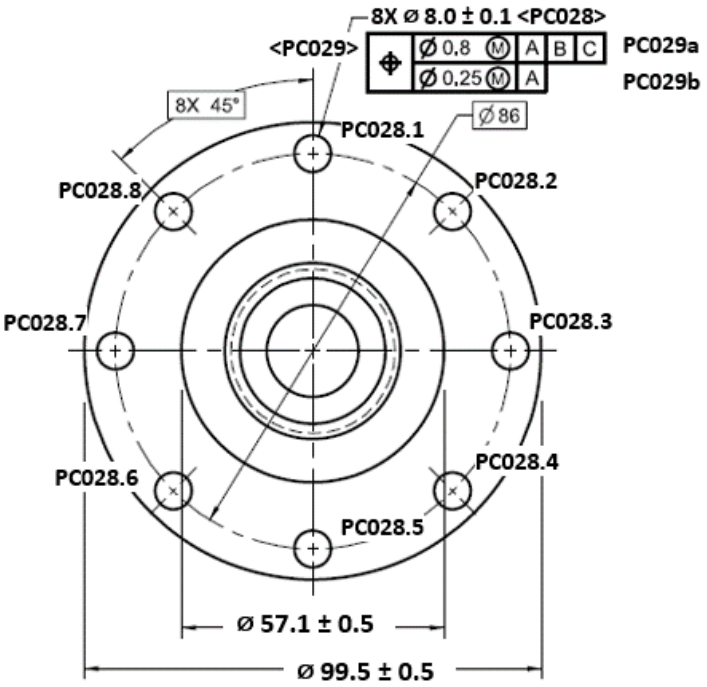


Figure 60: Composite Position of Pattern Feature with Feature Instance (informative)

5.9.9.3.3 BOC Example

An example BoC is shown in the table below. A size tolerance Instance Tag is applied to each of the eight hole features, whereas each facet of the composite position tolerance Instance Tag is applied to the hole pattern feature.

Tag (Base)	Tag (Ext)	UUID (Base)	UUID (Ext)	Description
PC028	1	651ADED1-FF04-498A-968E-044147A25064	F1's UUID	**8.0 ± 0.1 Size on Hole1**
PC028	2	651ADED1-FF04-498A-968E-044147A25064	F2's UUID	**8.0 ± 0.1 Size on Hole2**
PC028	3	651ADED1-FF04-498A-968E-044147A25064	F3's UUID	**8.0 ± 0.1 Size on Hole3**
PC028	4	651ADED1-FF04-498A-968E-044147A25064	F4's UUID	**8.0 ± 0.1 Size on Hole4**
PC028	5	651ADED1-FF04-498A-968E-044147A25064	F5's UUID	**8.0 ± 0.1 Size on Hole5**
PC028	6	651ADED1-FF04-498A-968E-044147A25064	F6's UUID	**8.0 ± 0.1 Size on Hole6**
PC028	7	651ADED1-FF04-498A-968E-044147A25064	F7's UUID	**8.0 ± 0.1 Size on Hole7**
PC028	8	651ADED1-FF04-498A-968E-044147A25064	F8's UUID	**8.0 ± 0.1 Size on Hole8**
PC029	a	1986E974-2AE2-4C3F-9897-9EEFF768E673	a	**[CompPos|0.8m|A|B|C] on Hole Pattern**
PC029	b	1986E974-2AE2-4C3F-9897-9EEFF768E673	b	**[CompPos|0.25|A|B] on Hole Pattern**

5.9.10 Position of Counterbored Holes

The positional tolerance of counterbored holes may be applied via multiple techniques.

5.9.10.1 *Counterbored Holes – Single Position Tolerance*

A positional tolerance that is applied to both the hole and the counterbore.

5.9.10.1.1 Example

Example of four counterbored holes with diameter size tolerances, depth tolerance, and a single positional tolerance.

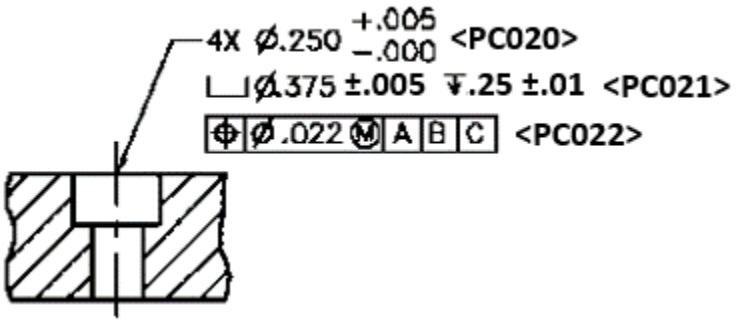


Figure 61: 4X Counterbored Holes with One Position Tolerance

5.9.10.1.2 Application:

This example shows how a single positional tolerance's feature control frame may be placed to indicate the same position tolerance applies to both the hole and the counterbore, four times. In this case, each tolerance line of the counterbore annotation has its own explicit PC Reference Tag. PC Reference Tag <PC020> is applied to the hole's diameter tolerance. Next, PC Reference Tag <PC021> is applied to the counterbore's size tolerance and a depth tolerance to the counterbore's bottom. Finally, PC Reference Tag <PC022> is applied to the overall counterbore and hole's positional tolerance.

The PC Instance ID Table for the <PC020> multiple feature use case is shown below:

PC Base ID	PC Extension ID			PC Instance ID Example
	MFAI	MFRI	RGNI	
PC020	n	Y	n	PC020.1

The PC Instance ID Table for the <PC021> and <PC022> multi-facet and multiple feature use case is shown below:

PC Base ID	PC Extension ID			PC Instance ID Example
	MFAI	MFRI	RGNI	
PC021	Y	Y	n	PC021a.1

Assuming that the explicit feature reference identifiers are not known, a feature reference sequence number approach should be used for each counterbore's occurrence. Starting with sequence number "1", the hole's size PC Instance Tags would result in PC020.1, PC020.2, PC020.3, and PC020.4. Next, since the counterbore's size and depth tolerance have the same PC Reference Tag, the size tolerance instance will receive the "a" instance extension and the depth tolerance instance will receive the "b" instance extension, then the FRSN approach will increase

the instance resulting in PC021a.1, PC021b.1, PC021a.2, PC021b.2 PC021a.3, PC021b.3 PC021a.1, and PC021b.4. Finally, the single positional tolerance, with PC Reference Tag <PC022>, is applied to both the hole and counterbore, resulting in two instance extension branches, one for the hole starting with "a" and one for the counterbore starting with "b". Then, as the positional tolerance is applied 4X places, the instance tags will be extended resulting in PC022a.1, PC022a.2, PC022a.3, and PC022a.4 for the 4X holes and PC022b.1, PC022b.2, and PC022b.3, and PC022b.4 for the 4X Counterbores.

5.9.10.1.3 BOC Example

An example BoC is shown in the table below.

Tag (Base)	Tag (Ext)	UUID (Base)	UUID (Ext)	Description
PC020	1	FEC2B504-F457-4766-9905-0CD3C5264139	F1's UUID	**.250 +.005-.000 Size on Hole1**
PC020	2	FEC2B504-F457-4766-9905-0CD3C5264139	F2's UUID	**.250 +.005-.000 Size on Hole2**
PC020	3	FEC2B504-F457-4766-9905-0CD3C5264139	F3's UUID	**.250 +.005-.000 Size on Hole3**
PC020	4	FEC2B504-F457-4766-9905-0CD3C5264139	F4's UUID	**.250 +.005-.000 Size on Hole4**
PC021	a.1	5226FBBB-FB7C-4B2A-B536-2E400B662DA2	CB1's UUID	**.375 ± 0.05 Size on CBore1**
PC021	b.1	5226FBBB-FB7C-4B2A-B536-2E400B662DA2	CB1's UUID	**.25 ± 0.01 Depth on CBore1**
PC021	a.2	5226FBBB-FB7C-4B2A-B536-2E400B662DA2	CB2's UUID	**.375 ± 0.05 Size on CBore2**
PC021	b.2	5226FBBB-FB7C-4B2A-B536-2E400B662DA2	CB2's UUID	**.25 ± 0.01 Depth on CBore2**
PC021	a.3	5226FBBB-FB7C-4B2A-B536-2E400B662DA2	CB3's UUID	**.375 ± 0.05 Size on CBore3**
PC021	b.3	5226FBBB-FB7C-4B2A-B536-2E400B662DA2	CB3's UUID	**.25 ± 0.01 Depth on CBore3**
PC021	a.4	5226FBBB-FB7C-4B2A-B536-2E400B662DA2	CB4's UUID	**.375 ± 0.05 Size on CBore4**
PC021	b.4	5226FBBB-FB7C-4B2A-B536-2E400B662DA2	CB4's UUID	**.25 ± 0.01 Depth on CBore4**
PC022	a.1	9B4142BF-9264-46BD-9E97-A4C732AF0D9F	F1's UUID	**[Pos|.022m|A|B|C] on Hole1**
PC022	b.1	9B4142BF-9264-46BD-9E97-A4C732AF0D9F	CB1's UUID	**[Pos|.022m|A|B|C] on CBore1**
PC022	a.2	9B4142BF-9264-46BD-9E97-A4C732AF0D9F	F2's UUID	**[Pos|.022m|A|B|C] on Hole2**
PC022	b.2	9B4142BF-9264-46BD-9E97-A4C732AF0D9F	CB2's UUID	**[Pos|.022m|A|B|C] on CBore2**
PC022	a.3	9B4142BF-9264-46BD-9E97-A4C732AF0D9F	F3's UUID	**[Pos|.022m|A|B|C] on Hole3**
PC022	b.3	9B4142BF-9264-46BD-9E97-A4C732AF0D9F	CB3's UUID	**[Pos|.022m|A|B|C] on CBore3**
PC022	a.4	9B4142BF-9264-46BD-9E97-A4C732AF0D9F	F4's UUID	**[Pos|.022m|A|B|C] on Hole4**
PC022	b.4	9B4142BF-9264-46BD-9E97-A4C732AF0D9F	CB4's UUID	**[Pos|.022m|A|B|C] on CBore4**

5.9.10.2 *Counterbored Holes - Separate Position Tolerances*

Separate positional tolerances that are applied to both the hole and the counterbore, respectively.

5.9.10.2.1 Example

Example of four counterbored holes with diameter size tolerances, depth tolerance, and separate positional tolerance.

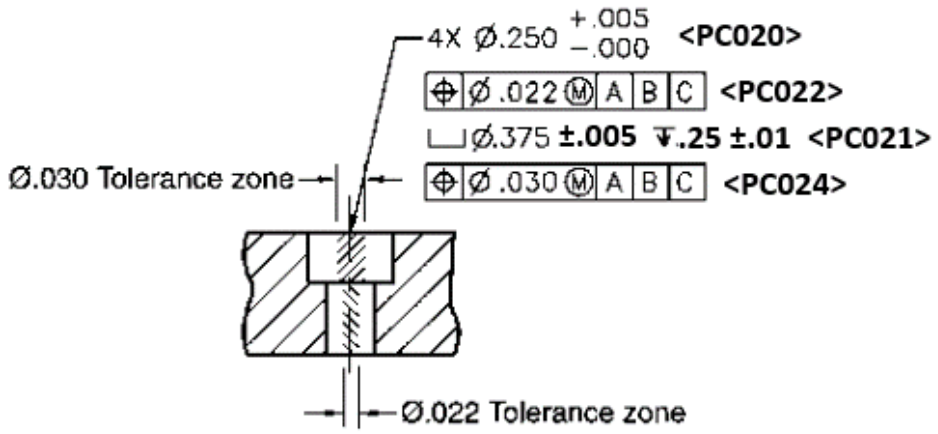


Figure 62: 4X Counterbored Holes with Separate Position Tolerances

5.9.10.2.2 Application:

This example shows a different position tolerance is applied to the hole and counterbore callout to indicate that different position tolerances are applie d to the hole and counterbore, four times. In this case, each tolerance line of the coun terbore annotation has its own explicit PC Reference Tag. PC Reference Tag <PC020> is applied to the hole's diameter tolerance. Next, PC Reference Tag <PC022> is applied to the hole's position tolerance. Then, PC Reference Tag <PC021> is applied to the counterbore's size tolerance and a depth tolerance to the counterbore's bottom. Finally, PC Reference Tag <PC024> is applied to the overall counterbore's positional tolerance.

The PC Instance ID Table for <PC020>, <PC022>, and <PC024> multiple feature use case is shown below:

PC Base ID	PC Extension ID			PC Instance ID Example
	MFAI	MFRI	RGNI	
PC020	n	Y	n	PC020.1

The PC Instance ID Table for <PC021> multi-facet and multiple feature use case is shown below:

PC Base ID	PC Extension ID			PC Instance ID Example
	MFAI	MFRI	RGNI	
PC021	Y	Y	n	PC021a.1

Assuming that the explicit feature reference identifiers are not known, a feature reference sequence number approach should be used for each counterbore's occurrence. Starting with sequence number "1", the hole's size PC Instance Tags would result in PC020.1, PC020.2, PC020.3, and PC020.4. Next, the hole's position tolerance PC Instance Tags would result in PC022.1, PC022.2, PC022.3, and PC022.4. Then, since the counterbore's size and depth tolerance have the same PC Reference

Tag, the size tolerance instance will receive the "a" instance extension and the depth tolerance instance will receive the "b" instance extension, then the FRSN approach will increase the instance resulting in PC021a.1, PC021b.1, PC021a.2, PC021b.2 PC021a.3, PC021b.3 PC021a.4, and PC021b.4. Finally, the counterbore's position tolerance PC Instance Tags would result in PC024.1, PC024.2, PC024.3, and PC024.4.

5.9.10.2.1 BOC Example

An example BoC is shown in the table below.

Tag (Base)	Tag (Ext)	UUID (Base)	UUID (Ext)	Description
PC020	1	FEC2B504-F457-4766-9905-0CD3C5264139	F1's UUID	**.250 +.005-.000 Size on Hole1**
PC020	2	FEC2B504-F457-4766-9905-0CD3C5264139	F2's UUID	**.250 +.005-.000 Size on Hole2**
PC020	3	FEC2B504-F457-4766-9905-0CD3C5264139	F3's UUID	**.250 +.005-.000 Size on Hole3**
PC020	4	FEC2B504-F457-4766-9905-0CD3C5264139	F4's UUID	**.250 +.005-.000 Size on Hole4**
PC021	a.1	5226FBBB-FB7C-4B2A-B536-2E400B662DA2	CB1's UUID	**.375 ± 0.05 Size on CBore1**
PC021	b.1	5226FBBB-FB7C-4B2A-B536-2E400B662DA2	CB1's UUID	**.25 ± 0.01 Depth on CBore1**
PC021	a.2	5226FBBB-FB7C-4B2A-B536-2E400B662DA2	CB2's UUID	**.375 ± 0.05 Size on CBore2**
PC021	b.2	5226FBBB-FB7C-4B2A-B536-2E400B662DA2	CB2's UUID	**.25 ± 0.01 Depth on CBore2**
PC021	a.3	5226FBBB-FB7C-4B2A-B536-2E400B662DA2	CB3's UUID	**.375 ± 0.05 Size on CBore3**
PC021	b.3	5226FBBB-FB7C-4B2A-B536-2E400B662DA2	CB3's UUID	**.25 ± 0.01 Depth on CBore3**
PC021	a.4	5226FBBB-FB7C-4B2A-B536-2E400B662DA2	CB4's UUID	**.375 ± 0.05 Size on CBore4**
PC021	b.4	5226FBBB-FB7C-4B2A-B536-2E400B662DA2	CB4's UUID	**.25 ± 0.01 Depth on CBore4**
PC022	1	9B4142BF-9264-46BD-9E97-A4C732AF0D9F	F1's UUID	**[Pos|.022m|A|B|C] on Hole1**
PC022	2	9B4142BF-9264-46BD-9E97-A4C732AF0D9F	F2's UUID	**[Pos|.022m|A|B|C] on Hole2**
PC022	3	9B4142BF-9264-46BD-9E97-A4C732AF0D9F	F3's UUID	**[Pos|.022m|A|B|C] on Hole3**
PC022	4	9B4142BF-9264-46BD-9E97-A4C732AF0D9F	F4's UUID	**[Pos|.022m|A|B|C] on Hole4**
PC024	1	7CBB99EF-3934-41B0-9F5A-1AB629EF266D	CB1's UUID	**[Pos|.030m|A|B|C] on CBore1**
PC024	2	7CBB99EF-3934-41B0-9F5A-1AB629EF266D	CB2's UUID	**[Pos|.030m|A|B|C] on CBore2**
PC024	3	7CBB99EF-3934-41B0-9F5A-1AB629EF266D	CB3's UUID	**[Pos|.030m|A|B|C] on CBore3**
PC024	4	7CBB99EF-3934-41B0-9F5A-1AB629EF266D	CB4's UUID	**[Pos|.030m|A|B|C] on CBore4**

5.9.10.3 *Counterbores Related to Datum Holes Individually*

A positional tolerance applied to an individual hole and the hole's counterbore is positional tolerance with respect to the hole.

5.9.10.3.1 Example

Example of four counterbored holes with diameter size tolerances, depth tolerance, and separate positional tolerance, the counterbore positional tolerance is individually located to its coaxial hole as a datum feature.

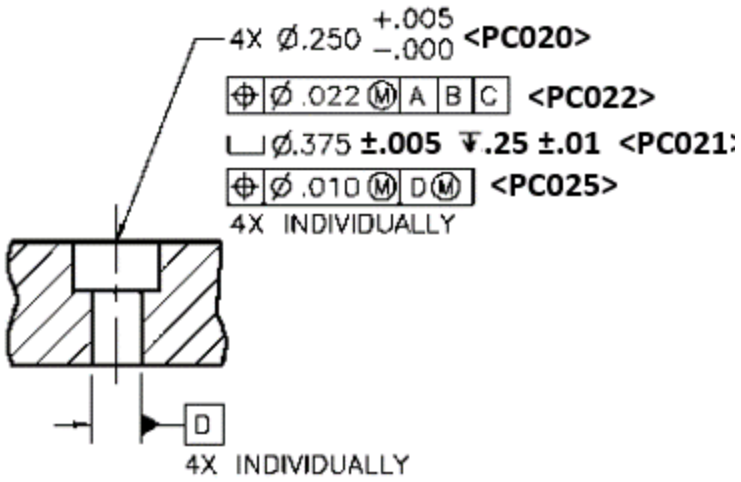


Figure 63: 4X Counterbores Related to Datum Holes Individually

5.9.10.3.2 Application:

This example shows a different position tolerance is applied to the hole and counterbore callout to indicate that different position tolerances are applied to the hole and counterbore, four times. In this case, each tolerance line of the counterbore annotation has its own explicit Reference Tag. PC Reference Tag <PC020> is applied to the hole's diameter tolerance. Next, PC Reference Tag <PC022> is applied to the hole's position tolerance. Then, PC Reference Tag <PC021> is applied to the counterbore's size tolerance and a depth tolerance to the counterbore's bottom. Finally, PC Reference Tag <PC025> is applied to the overall counterbore's positional tolerance.

The PC Instance ID Table for <PC020>, <PC022>, and <PC025> multiple feature use case is shown below:

PC Base ID	PC Extension ID			PC Instance ID Example
	MFAI	MFRI	RGNI	
PC020	n	Y	n	PC020.1

The PC Instance ID Table for <PC021> multi-facet and multiple feature use case is shown below:

PC Base ID	PC Extension ID			PC Instance ID Example
	MFAI	MFRI	RGNI	
PC021	Y	Y	n	PC021a.1

Assuming that the explicit feature reference identifiers are not known, a feature reference sequence number approach should be used for each counterbore's occurrence. Starting with sequence number "1", the hole's size PC Instance Tags would result in PC020.1, PC020.2, PC020.3, and PC020.4. Next, the hole's position tolerance PC Instance Tags would result in PC022.1, PC022.2, PC022.3, and PC022.4.

Then, since the counterbore's size and depth tolerance have the same PC Reference Tag, the size tolerance instance will receive the "a" instance extension and the depth tolerance instance will receive the "b" instance extension, then the FRSN approach will increase the instance resulting in PC021a.1, PC021b.1, PC021a.2, PC021b.2 PC021a.3, PC021b.3 PC021a.4, and PC021b.4. Finally, the counterbore's position tolerance PC Instance Tags would result in PC025.1, PC025.2, PC025.3, and PC025.4.

5.9.10.3.3 BOC Example

An example BoC is shown in the table below.

Tag (Base)	Tag (Ext)	UUID (Base)	UUID (Ext)	Description
PC020	1	FEC2B504-F457-4766-9905-0CD3C5264139	F1's UUID	**.250 +.005-.000 Size on Hole1**
PC020	2	FEC2B504-F457-4766-9905-0CD3C5264139	F2's UUID	**.250 +.005-.000 Size on Hole2**
PC020	3	FEC2B504-F457-4766-9905-0CD3C5264139	F3's UUID	**.250 +.005-.000 Size on Hole3**
PC020	4	FEC2B504-F457-4766-9905-0CD3C5264139	F4's UUID	**.250 +.005-.000 Size on Hole4**
PC021	a.1	5226FBBB-FB7C-4B2A-B536-2E400B662DA2	CB1's UUID	**.375 ± 0.005 Size on CBore1**
PC021	b.1	5226FBBB-FB7C-4B2A-B536-2E400B662DA2	CB1's UUID	**.25 ± 0.01 Depth on CBore1**
PC021	a.2	5226FBBB-FB7C-4B2A-B536-2E400B662DA2	CB2's UUID	**.375 ± 0.005 Size on CBore2**
PC021	b.2	5226FBBB-FB7C-4B2A-B536-2E400B662DA2	CB2's UUID	**.25 ± 0.01 Depth on CBore2**
PC021	a.3	5226FBBB-FB7C-4B2A-B536-2E400B662DA2	CB3's UUID	**.375 ± 0.005 Size on CBore3**
PC021	b.3	5226FBBB-FB7C-4B2A-B536-2E400B662DA2	CB3's UUID	**.25 ± 0.01 Depth on CBore3**
PC021	a.4	5226FBBB-FB7C-4B2A-B536-2E400B662DA2	CB4's UUID	**.375 ± 0.005 Size on CBore4**
PC021	b.4	5226FBBB-FB7C-4B2A-B536-2E400B662DA2	CB4's UUID	**.25 ± 0.01 Depth on CBore4**
PC022	1	9B4142BF-9264-46BD-9E97-A4C732AF0D9F	F1's UUID	**[Pos|.022m|A|B|C] on Hole1**
PC022	2	9B4142BF-9264-46BD-9E97-A4C732AF0D9F	F2's UUID	**[Pos|.022m|A|B|C] on Hole2**
PC022	3	9B4142BF-9264-46BD-9E97-A4C732AF0D9F	F3's UUID	**[Pos|.022m|A|B|C] on Hole3**
PC022	4	9B4142BF-9264-46BD-9E97-A4C732AF0D9F	F4's UUID	**[Pos|.022m|A|B|C] on Hole4**
PC025	1	46636880-21DE-4346-B8E3-16ECBE8E7A65	CB1's UUID	**[Pos|.010m|Dm] on CBore1**
PC025	2	46636880-21DE-4346-B8E3-16ECBE8E7A65	CB2's UUID	**[Pos|.010m|Dm] on CBore2**
PC025	3	46636880-21DE-4346-B8E3-16ECBE8E7A65	CB3's UUID	**[Pos|.010m|Dm] on CBore3**
PC025	4	46636880-21DE-4346-B8E3-16ECBE8E7A65	CB4's UUID	**[Pos|.010m|Dm] on CBore4**

5.10 Single Product Characteristic including Repetitive Group Applications

This section addresses PC Reference Tags, which are grouped and repetitively applied, requiring multiple PC Instance Tags for each repetitive group.

5.10.1 Repetitive Group of Multiple Features

A group of annotations, which are repetitively applied.

5.10.1.1 *Example*

This example shows five groups of annotations, which include a six-hole pattern with each hole having a size and depth tolerance, and a positional tolerance applied and a two-segmented composite profile tolerance applied to a cam-shaped pocket. Both the six-hole pattern and the cam-shaped pocket are grouped and repeated five times.

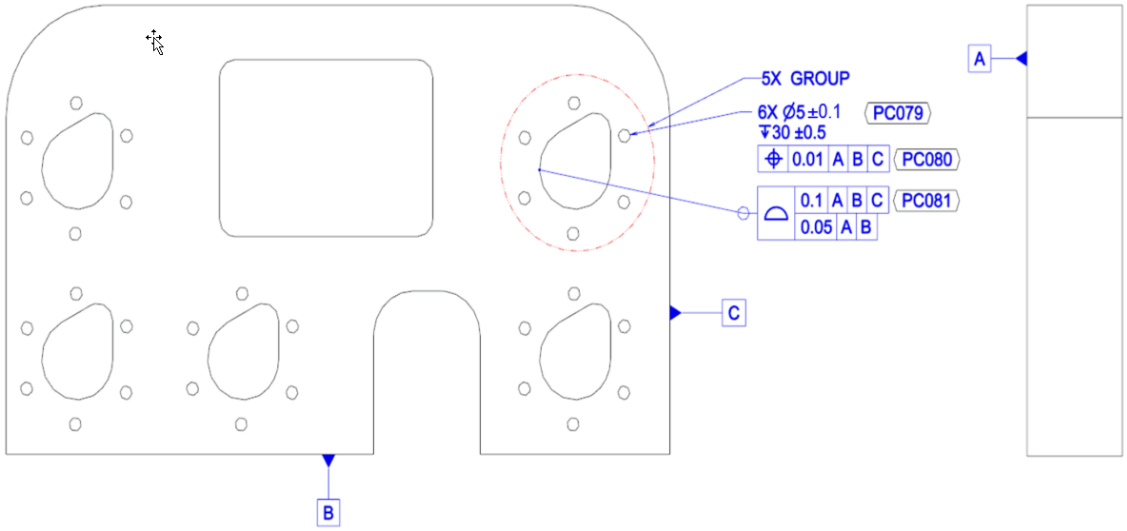


Figure 64: Repetitive Group of Multiple Features Example

5.10.1.2 *Application:*

The application of interest is the application of the one diameter tolerance with depth annotation identified as PC Reference Tag <PC079>. For this example, <PC079>> has 60 permutations of PC Instance Tags for each PC079 instance.

- One Characteristic Reference Tag < PC079>
- A Two Multi-Faceted Dimensional Tolerance Annotation (a, b)
- Six Multiple Featured References (1,2,3,4,5,6)
- 5X Repetitive Group Number (1,2,3,4,5)

The PC Instance ID Table for multi-facet, multiple features, and repetitive group use case is shown below:

PC Base ID	PC Extension ID			PC Instance ID Example
	MFAI	MFRI	RGNI	
PC079	Y	Y	Y	PC079a.1:1
PC079	Y	Y	Y	PC079b.6:5

Following the format provided in section 5.4.7, each PC Instance Tag for the PC Reference Tag <PC079> is illustrated below.

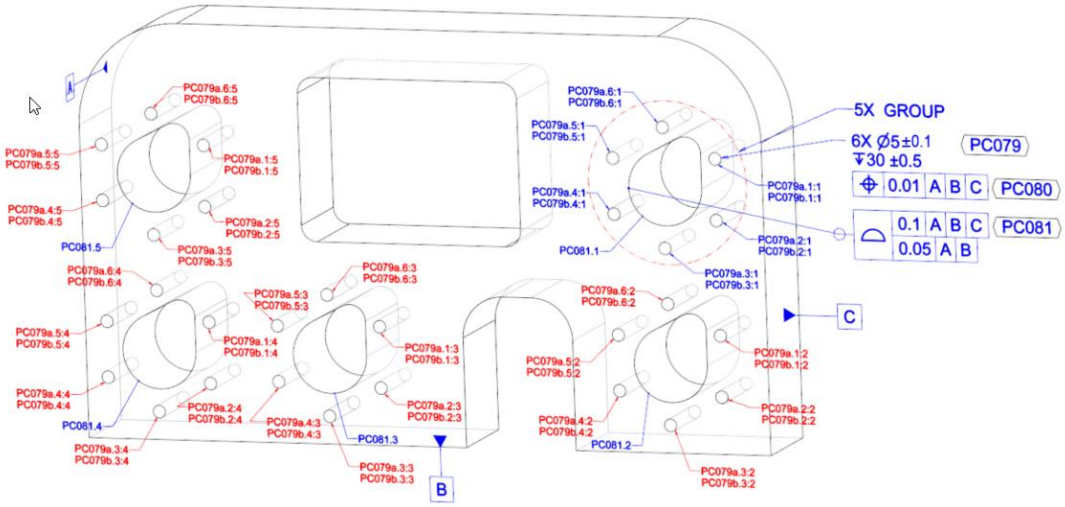


Figure 65: Repetitive Group of Multiple Features Application

For this example, there are 60 permutations of PC Reference Tag <PC079>, which starts with the first pattern hole feature of the first repetitive PC Instance Tags PC079a.1:1, and PC079b.1:1 and ending with the fifth pattern hole feature of the sixth repetitive PC Instance Tags PC079a.6:5, and PC079b.6:5.

5.10.1.3 *BOC Example*

The BOC example is quite extensive and only captures the first repetitive group instance and the last repetitive group instance.

Tag (Base)	Tag (Ext)	UUID (Base)	UUID (Ext)	Description
PC079	a.1:1	651ADED1-FF04-498A-968E-044147A25064	a.1:1	**SizeTol of Pattern Hole 1 on Repetitive Group 1.**
PC079	b.1:1	651ADED1-FF04-498A-968E-044147A25064	b.1:1	**DepthTol of Pattern Hole 1 on Repetitive Group 1.**
PC080	1:1	B44264BA-6A20-4409-9CD4-9E68F8EE27A1	1:1	**[Pos|0.01|A|B|C] on Pattern Hole 1 on Repetitive Group 1**
PC079	a.2:1	651ADED1-FF04-498A-968E-044147A25064	a.2:1	**SizeTol on Pattern Hole 2 on Repetitive Group 1.**
PC079	b.2:1	651ADED1-FF04-498A-968E-044147A25064	b.2:1	**DepthTol of Pattern Hole 2 on Repetitive Group 1.**
PC080	2:1	B44264BA-6A20-4409-9CD4-9E68F8EE27A1	2:1	**[Pos|0.01|A|B|C] on Pattern Hole 2 on Repetitive Group 1**
PC079	a.3:1	651ADED1-FF04-498A-968E-044147A25064	a.3:1	**SizeTol on Pattern Hole 3 on Repetitive Group 1.**
PC079	b.3:1	651ADED1-FF04-498A-968E-044147A25064	b.3:1	**DepthTol of Pattern Hole 3 on Repetitive Group 1.**
PC080	3:1	B44264BA-6A20-4409-9CD4-9E68F8EE27A1	3:1	**[Pos|0.01|A|B|C] on Pattern Hole 3 on Repetitive Group 1**
PC079	a.4:1	651ADED1-FF04-498A-968E-044147A25064	a.4:1	**SizeTol on Pattern Hole 4 on Repetitive Group 1.**
PC079	b.4:1	651ADED1-FF04-498A-968E-044147A25064	b.4:1	**DepthTol of Pattern Hole 4 on Repetitive Group 1.**
PC080	4:1	B44264BA-6A20-4409-9CD4-9E68F8EE27A1	4:1	**[Pos|0.01|A|B|C] on Pattern Hole 4 on Repetitive Group 1**

PC079	a.5:1	651ADED1-FF04-498A-968E-044147A25064	a.5:1	**SizeTol on Pattern Hole 5 on Repetitive Group 1.**
PC079	b.5:1	651ADED1-FF04-498A-968E-044147A25064	b.5:1	**DepthTol of Pattern Hole 5 on Repetitive Group 1.**
PC080	5:1	B44264BA-6A20-4409-9CD4-9E68F8EE27A1	5:1	**[Pos|0.01|A|B|C] on Pattern Hole 5 on Repetitive Group 1**
PC079	a.6:1	651ADED1-FF04-498A-968E-044147A25064	a.6:1	**SizeTol on Pattern Hole 6 on Repetitive Group 1.**
PC079	b.6:1	651ADED1-FF04-498A-968E-044147A25064	b.6:1	**DepthTol of Pattern Hole 6 on Repetitive Group 1.**
PC080	6:1	B44264BA-6A20-4409-9CD4-9E68F8EE27A1	6:1	**[Pos|0.01|A|B|C] on Pattern Hole 6 on Repetitive Group 1**
PC081	a:1	C3ECF0A7-718A-4A59-9BA5-E4DF2DF20140	a:1	**[ComProf|0.1|A|B|C] on Cam Pocket on Repetitive Group 1**
PC081	b:1	C3ECF0A7-718A-4A59-9BA5-E4DF2DF20140	b:1	**[ComProf|0.05|A|B] on Cam Pocket on Repetitive Group 1**
PC079	. . .	651ADED1-FF04-498A-968E-044147A25064	. . .	. . .
PC080	. . .	B44264BA-6A20-4409-9CD4-9E68F8EE27A1	. . .	. . .
PC081	. . .	C3ECF0A7-718A-4A59-9BA5-E4DF2DF20140	. . .	. . .
PC079	a.1:5	651ADED1-FF04-498A-968E-044147A25064	a.1:5	**SizeTol on Pattern Hole 1 on Repetitive Group 5.**
PC079	b.1:5	651ADED1-FF04-498A-968E-044147A25064	b.1:5	**DepthTol of Pattern Hole 1 on Repetitive Group 5.**
PC080	1:5	B44264BA-6A20-4409-9CD4-9E68F8EE27A1	1:5	**[Pos|0.01|A|B|C] on Pattern Hole 1 on Repetitive Group 5**
PC079	a.2:5	651ADED1-FF04-498A-968E-044147A25064	a.2:5	**SizeTol on Pattern Hole 2 on Repetitive Group 5.**
PC079	b.2:5	651ADED1-FF04-498A-968E-044147A25064	b.2:5	**DepthTol of Pattern Hole 2 on Repetitive Group 5.**
PC080	2:5	B44264BA-6A20-4409-9CD4-9E68F8EE27A1	2:5	**[Pos|0.01|A|B|C] on Pattern Hole 2 on Repetitive Group 5**
PC079	a.3:5	651ADED1-FF04-498A-968E-044147A25064	a.3:5	**SizeTol on Pattern Hole 3 on Repetitive Group 5.**
PC079	b.3:5	651ADED1-FF04-498A-968E-044147A25064	b.3:5	**DepthTol of Pattern Hole 3 on Repetitive Group 5.**
PC080	3:5	B44264BA-6A20-4409-9CD4-9E68F8EE27A1	3:5	**[Pos|0.01|A|B|C] on Pattern Hole 3 on Repetitive Group 5**
PC079	a.4:5	651ADED1-FF04-498A-968E-044147A25064	a.4:5	**SizeTol on Pattern Hole 4 on Repetitive Group 5.**
PC079	b.4:5	651ADED1-FF04-498A-968E-044147A25064	b.4:5	**DepthTol of Pattern Hole 4 on Repetitive Group 5.**
PC080	4:5	B44264BA-6A20-4409-9CD4-9E68F8EE27A1	4:5	**[Pos|0.01|A|B|C] on Pattern Hole 4 on Repetitive Group 5**
PC079	a.5:5	651ADED1-FF04-498A-968E-044147A25064	a.5:5	**SizeTol on Pattern Hole 5 on Repetitive Group 5.**
PC079	b.5:5	651ADED1-FF04-498A-968E-044147A25064	b.5:5	**DepthTol of Pattern Hole 5 on Repetitive Group 5.**
PC080	5:5	B44264BA-6A20-4409-9CD4-9E68F8EE27A1	5:5	**[Pos|0.01|A|B|C] on Pattern Hole 5 on Repetitive Group 5**
PC079	a.6:5	651ADED1-FF04-498A-968E-044147A25064	a.6:5	**SizeTol on Pattern Hole 6 on Repetitive Group 5.**
PC079	b.6:5	651ADED1-FF04-498A-968E-044147A25064	b.6:5	**DepthTol of Pattern Hole 6 on Repetitive Group 5.**

PC080	6:5	B44264BA-6A20-4409-9CD4-9E68F8EE27A1	6:5	**[Pos|0.01|A|B|C] on Pattern Hole 6 on Repetitive Group 5**
PC081	a:5	C3ECF0A7-718A-4A59-9BA5-E4DF2DF20140	a:5	**[ComProf|0.1|A|B|C] on Cam Pocket on Repetitive Group 5**
PC081	b:5	C3ECF0A7-718A-4A59-9BA5-E4DF2DF20140	b:5	**[ComProf|0.05|A|B] on Cam Pocket on Repetitive Group 5**

5.10.2 Repetitive All-Around Profile Group

A Composite Profile Tolerance applied all-around a profile group, which is repeated.

5.10.2.1 *Example*

A repetitive composite surface profile tolerance applied all-around, repeated in two places. The application of this profile tolerance against a profile group pocket provides several use-case examples for PC Instance Tags.

- Individual Profile Group
- Multiple Faces of Profile Group

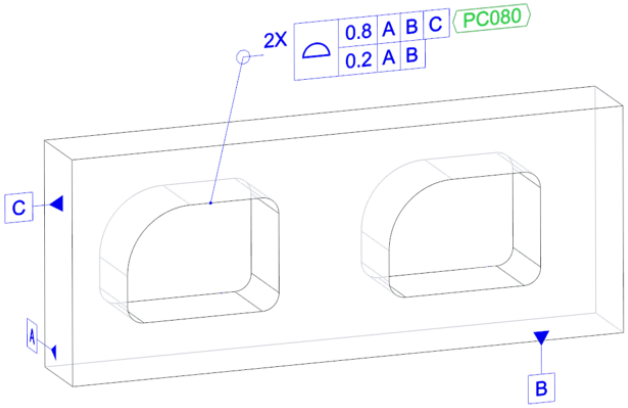


Figure 66: Repetition of All-Around Profile Group Example

5.10.2.2 *Individual Profile Group Application:*

The application of interest is of the one multi-faceted composite tolerance annotation identified as PC Reference Tag <PC080>. For this example, <PC080>has four permutations of PC Reference Tags for each <PC080> instance, which are applied to each repetitive pocket profile group.

- One Characteristic Reference Tag <PC080>
- A Two Multi-Faceted Composite Tolerance Annotation (a, b)
- Two Repetitive Group Numbers (1,2)

The PC Instance ID Table for multi-facet and repetitive group use case is shown below:

PC Base ID	PC Extension ID			PC Instance ID Example
	MFAI	MFRI	RGNI	
PC080	Y	n	Y	PC080a:1
PC080	Y	n	Y	PC080b:2

Following the format provided in section 5.4.7, each instance for the PC Reference Tag <PC080> is illustrated below.

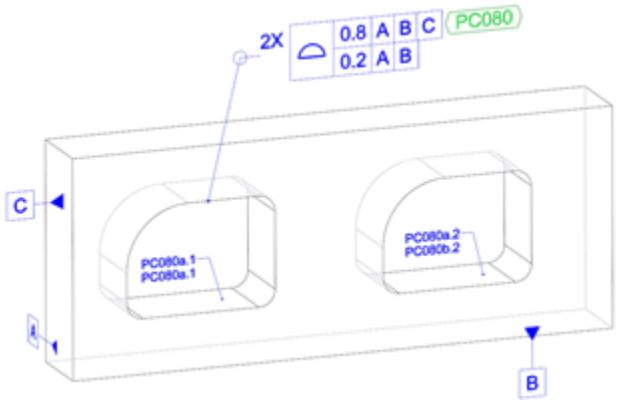


Figure 67: Repetition of All-Around Profile Example as Individual Profile Group Application

For this example, there are four permutations of PC Instance Tags for PC Reference Tag <PC080>. The first set of repetitive PC Instance Tags for the first pocket-profile group feature per the annotation's leader line are PC080a:1 and PC080b:1, whereas the last set of PC Instance Tags for the last repetitive pocket-profile group feature are PC080a:2 and PC080b:2. For clarity, the pocket-profile group feature is highlighted for PC Instance Tags PC080a:2 and PC080b:2.

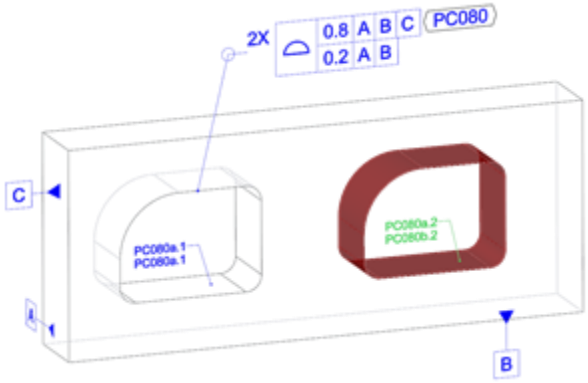


Figure 68: Query Highlighting of a Pocket Profile Group PC Instance Tags

5.10.2.3 *Individual Profile Group BOC Example*

Below is the BOC example listing PC Instance Tags with their multi-faceted and repetitive group extensions, PC instance UUIDs with their multi-faceted and repetitive group extensions and PC descriptions for the use case example shown above.

Tag (Base)	Tag (Ext)	UUID (Base)	UUID (Ext)	Description
PC080	a:1	351ADED1-FF04-498A-968E-044147A25064	a:1	**[CProf|0.8|A|B|C] on Profile Pocket1**
PC080	b.1	351ADED1-FF04-498A-968E-044147A25064	b.1	**[CProf|0.2|A|B] on Profile Pocket1**
PC080	a:2	351ADED1-FF04-498A-968E-044147A25064	a:2	**[CProf|0.8|A|B|C] on Profile Pocket2**

PC080	b:2	351ADED1-FF04-498A-968E-044147A25064	b:2	**[CProf|0.2|A|B] on Profile Pocket2**

5.10.2.4 *Multiple Faces of Profile Group Application:*

The application of interest is of the one multi-faceted composite tolerance annotation identified as PC Reference Tag <PC080>. For this example, <PC080> has 32 permutations of PC Reference Tags for each <PC080> instance, which are applied to each of the multiple faces of each repetitive pocket profile group.

- One Characteristic Reference Tag <PC080>
- A Two Multi-Faceted Composite Tolerance Annotation (a, b)
- Eight Multiple-Faced Pocket (1,2,3,4,5,6,7,8)
- Two Repetitive Group Numbers (1,2)

The PC Instance ID Table for multi-facet, multiple faces, and repetitive group use case is shown below:

PC Base ID	PC Extension ID			PC Instance ID Example
	MFAI	MFRI	RGNI	
PC080	Y	Y	Y	PC080a.1:1
PC080	Y	Y	Y	PC080b.8:2

Following the format provided in section 5.4.7, each instance for the PC Reference Tag <PC080> is illustrated below.

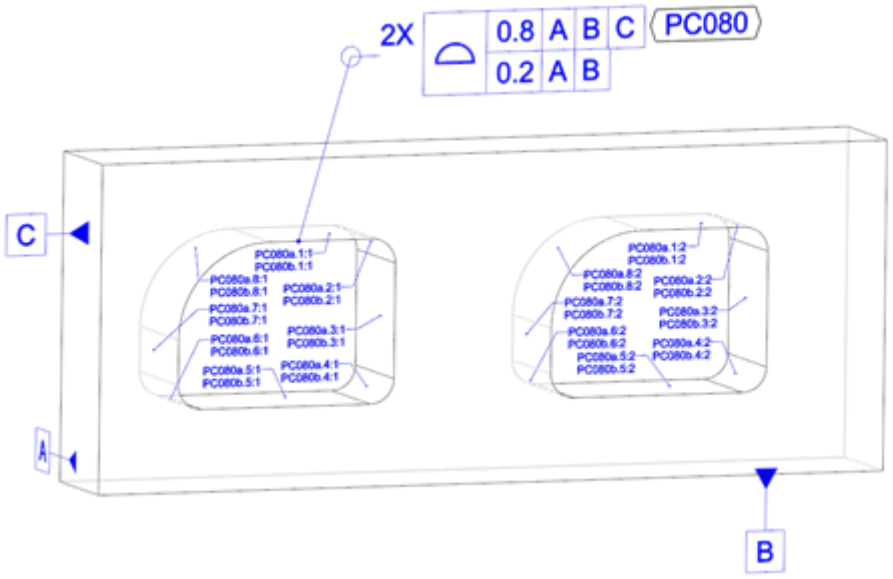


Figure 69: Repetition of All-Around Profile Example as Multiple Faces Profile Group Application

For this example, there are 32 permutations of PC Instance Tags for PC Reference Tag <PC080>. The first set of repetitive PC Instance Tags for the first pocket profile group feature per the annotation's leader line are PC080a.1:1 and PC080b.1:1, whereas the last

set of PC Instance Tags for the last repetitive pocket-profile group feature are PC080a.8:2 and PC080b.8:2. For clarity, the pocket-profile group feature is highlighted for PC Instance Tags PC080a.3:2 and PC080b.3:2.

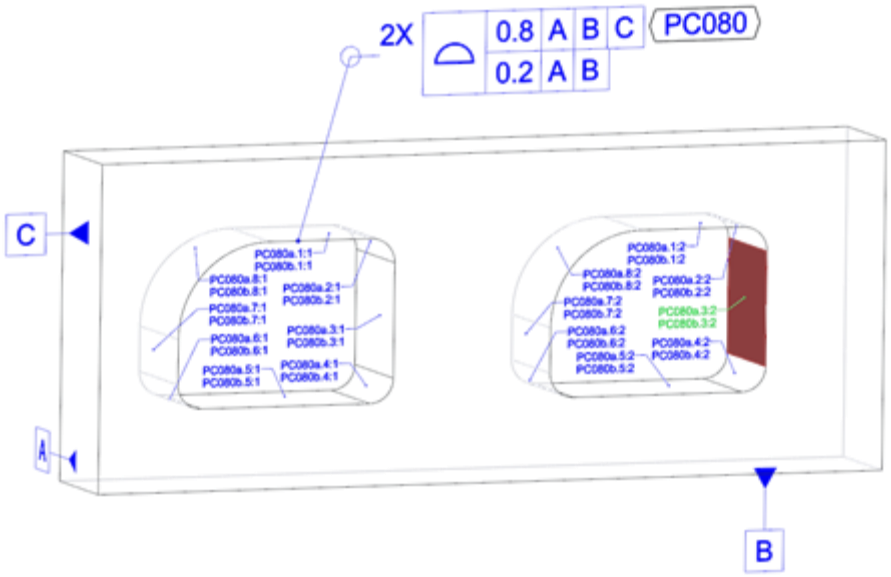


Figure 70: Query Highlighting of a Face on Pocket Profile Group for PC Instance Tags

5.10.2.5 *Multiple Faces of Profile BOC Example*

Below is the BOC example listing PC Instance Tags with their multi-faceted, multiple faces, and repetitive group extensions, PC instance UUIDs with their multi-faceted and repetitive group extensions and PC descriptions for the use case example shown above.

Tag (Base)	Tag (Ext)	UUID (Base)	UUID (Ext)	Description
PC080	a.1.1	351ADED1-FF04-498A-968E-044147A25064	a.1.1	**[CProf|0.8|A|B|C] Face1 on Pocket1**
PC080	b.1.1	351ADED1-FF04-498A-968E-044147A25064	b.1.1	**[CProf|0.2|A|B] Face1 on Pocket1**
PC080	a.2.1	351ADED1-FF04-498A-968E-044147A25064	a.2.1	**[CProf|0.8|A|B|C] Face2 on Pocket1**
PC080	b.2.1	351ADED1-FF04-498A-968E-044147A25064	b.2.1	**[CProf|0.2|A|B] Face2 on Pocket1**
PC080	a.3.1	351ADED1-FF04-498A-968E-044147A25064	a.3.1	**[CProf|0.8|A|B|C] Face3 on Pocket1**
PC080	b.3.1	351ADED1-FF04-498A-968E-044147A25064	b.3.1	**[CProf|0.2|A|B] Face3 on Pocket1**
PC080	a.4.1	351ADED1-FF04-498A-968E-044147A25064	a.4.1	**[CProf|0.8|A|B|C] Face4 on Pocket1**
PC080	b.4.1	351ADED1-FF04-498A-968E-044147A25064	b.4.1	**[CProf|0.2|A|B] Face4 on Pocket1**
PC080	a.5.1	351ADED1-FF04-498A-968E-044147A25064	a.5.1	**[CProf|0.8|A|B|C] Face5 on Pocket1**
PC080	b.5.1	351ADED1-FF04-498A-968E-044147A25064	b.5.1	**[CProf|0.2|A|B] Face5 on Pocket1**
PC080	a.6.1	351ADED1-FF04-498A-968E-044147A25064	a.6.1	**[CProf|0.8|A|B|C] Face6 on Pocket1**

PC080	b.6.1	351ADED1-FF04-498A-968E-044147A25064	b.6.1	**[CProf|0.2|A|B] Face6 on Pocket1**
PC080	a.7.1	351ADED1-FF04-498A-968E-044147A25064	a.7.1	**[CProf|0.8|A|B|C] Face7 on Pocket1**
PC080	b.7.1	351ADED1-FF04-498A-968E-044147A25064	b.7.1	**[CProf|0.2|A|B] Face7 on Pocket1**
PC080	a.8.1	351ADED1-FF04-498A-968E-044147A25064	a.8.1	**[CProf|0.8|A|B|C] Face8 on Pocket1**
PC080	b.8.1	351ADED1-FF04-498A-968E-044147A25064	b.8.1	**[CProf|0.2|A|B] Face8 on Pocket1**
PC080	a.1.2	351ADED1-FF04-498A-968E-044147A25064	a.1.2	**[CProf|0.8|A|B|C] Face1 on Pocket2**
PC080	b.1.2	351ADED1-FF04-498A-968E-044147A25064	b.1.2	**[CProf|0.2|A|B] Face1 on Pocket2**
PC080	a.2.2	351ADED1-FF04-498A-968E-044147A25064	a.2.2	**[CProf|0.8|A|B|C] Face2 on Pocket2**
PC080	b.2.2	351ADED1-FF04-498A-968E-044147A25064	b.2.2	**[CProf|0.2|A|B] Face2 on Pocket2**
PC080	a.3.2	351ADED1-FF04-498A-968E-044147A25064	a.3.2	**[CProf|0.8|A|B|C] Face3 on Pocket2**
PC080	b.3.2	351ADED1-FF04-498A-968E-044147A25064	b.3.2	**[CProf|0.2|A|B] Face3 on Pocket2**
PC080	a.4.2	351ADED1-FF04-498A-968E-044147A25064	a.4.2	**[CProf|0.8|A|B|C] Face4 on Pocket2**
PC080	b.4.2	351ADED1-FF04-498A-968E-044147A25064	b.4.2	**[CProf|0.2|A|B] Face4 on Pocket2**
PC080	a.5.2	351ADED1-FF04-498A-968E-044147A25064	a.5.2	**[CProf|0.8|A|B|C] Face5 on Pocket2**
PC080	b.5.2	351ADED1-FF04-498A-968E-044147A25064	b.5.2	**[CProf|0.2|A|B] Face5 on Pocket2**
PC080	a.6.2	351ADED1-FF04-498A-968E-044147A25064	a.6.2	**[CProf|0.8|A|B|C] Face6 on Pocket2**
PC080	b.6.2	351ADED1-FF04-498A-968E-044147A25064	b.6.2	**[CProf|0.2|A|B] Face6 on Pocket2**
PC080	a.7.2	351ADED1-FF04-498A-968E-044147A25064	a.7.2	**[CProf|0.8|A|B|C] Face7 on Pocket2**
PC080	b.7.2	351ADED1-FF04-498A-968E-044147A25064	b.7.2	**[CProf|0.2|A|B] Face7 on Pocket2**
PC080	a.8.2	351ADED1-FF04-498A-968E-044147A25064	a.8.2	**[CProf|0.8|A|B|C] Face8 on Pocket2**
PC080	b.8.2	351ADED1-FF04-498A-968E-044147A25064	b.8.2	**[CProf|0.2|A|B] Face8 on Pocket2**

5.11 Single Product Characteristic with Mixed Multiple Extension Applications

This section addresses a single PC Reference Tag, which has multiple extensions applications such as two or many of multi-faceted annotations, multiple feature reference, and repetitive group instance in which each permutation requires a PC Instance tag.

5.11.1 Multi-Facet Annotation Applied to Multiple Features/Faces

Composite tolerances may be applied to multiple features.

5.11.1.1 *Example*

This example shows three PC Reference Tags, in which PC Reference Tag <PC067> is a multi-faceted annotation applied to three features.

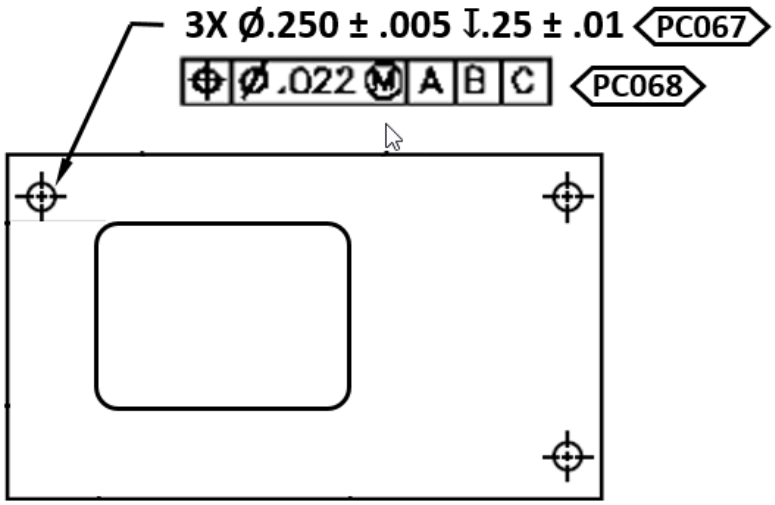


Figure 71: Multi-Faceted Annotation Applied to Multiple Features, PC Reference Tags

5.11.1.2 *Application:*

This composite tolerance example may be considered as one Product Characteristic that has multiple segments, which is applied within a common framework on a group of holes.

The PC Instance ID Table for multi-facet and multiple-feature use case is shown below:

PC Base ID	PC Extension ID			PC Instance ID Example
	MFAI	MFRI	RGNI	
PC067	Y	Y	n	PC067a.1
PC067	Y	Y	n	PC067b.3

For this example, there is one PC Reference Tag and PC Reference UUID. The reference is then extended with a multi-facet identifier for each composite segment such as PC024a and PC024b and then further extended with face/feature identifiers.

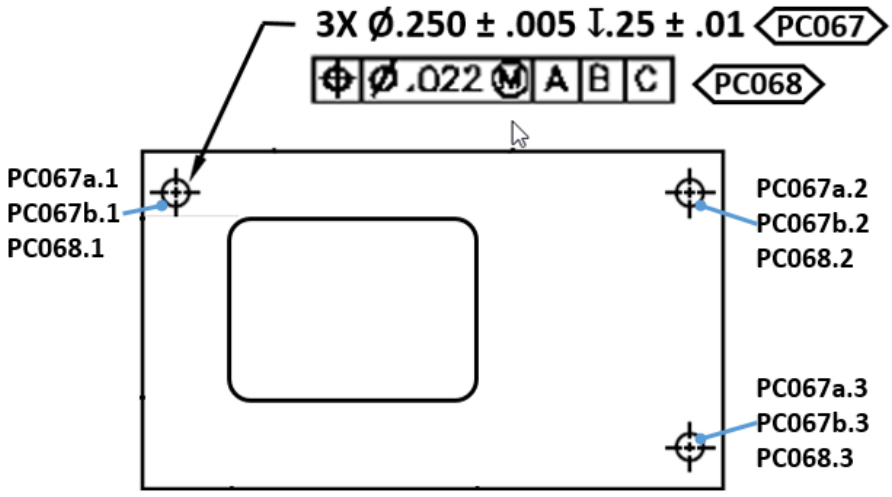


Figure 72: Multi-Faceted Annotation Applied to Multiple Features, PC Instance Tags

5.11.1.3 *BOC Example*

An example BoC is shown in the table below.

Tag (Base)	Tag (Ext)	UUID (Base)	UUID (Ext)	Description
PC067	a.1	5033CF76-C40F-4DDD-8E0C-34D922AB2B5D	Hole1Ref's UUID	Size .250 ± 005 on Hole 1
PC067	b.1	5033CF76-C40F-4DDD-8E0C-34D922AB2B5D	Hole1Ref's UUID	Depth .250 ± 005 on Hole 1
PC068	1	4DDDCF76-C40F-5033-8E0C-34D922AB2B5D	Hole1Ref's UUID	[Pos | .022 | A | B | C] on Hole1
PC067	a.2	5033CF76-C40F-4DDD-8E0C-34D922AB2B5D	Hole2Ref's UUID	Size .250 ± 005 on Hole 2
PC067	b.2	5033CF76-C40F-4DDD-8E0C-34D922AB2B5D	Hole2Ref's UUID	Depth .250 ± 005 on Hole 2
PC068	2	4DDDCF76-C40F-5033-8E0C-34D922AB2B5D	Hole2Ref's UUID	[Pos | .022 | A | B | C] on Hole2
PC067	a.3	5033CF76-C40F-4DDD-8E0C-34D922AB2B5D	Hole3Ref's UUID	Size .250 ± 005 on Hole 3
PC067	b.3	5033CF76-C40F-4DDD-8E0C-34D922AB2B5D	Hole3Ref's UUID	Depth .250 ± 005 on Hole 3
PC068	3	4DDDCF76-C40F-5033-8E0C-34D922AB2B5D	Hole3Ref's UUID	[Pos | .022 | A | B | C] on Hole3

5.12 Product Characteristics within an Assembly

Product end items are components that may be assemblies or individual piece-parts. Each component is a product defined with its own product identifier. Each product definition will have its own local set of unique product characteristics. For each component, the human-readable PC Reference Tags could start with the same initial tag number and be sequenced accordingly. Therefore, each component may share identical human-readable tags with other components within its assembly. Since, each component will have its own unique part identifier (e.g., 1A3456) and therefore the combination of the Part's Identifier with the PC Reference Tag (e.g., 1A3456; <PC007>) will uniquely identify the Product Characteristic. In addition, each PC Reference Tag should have a corresponding machine-readable PC Reference UUID, which will assure a unique identifier for each Product Characteristic.

Assembly components may be separable or inseparable assemblies.

5.12.1 Separable Assembly

Separable assembly typically contains an assembly component with multiple piece-part and/or sub-assembly components. Each component may have their own component unique PC Reference Tags.

5.12.1.1 *Example*

The example below shows a Lifting Block assembly product definition, which consists of multiple parts definitions. For this example, there is one top-level assembly, Lifting Block, 9A0814, consisting of multi-part components, in which piece-parts Shell Plate, 1A0518 and Sleeve, 1A4916 are each used twice.

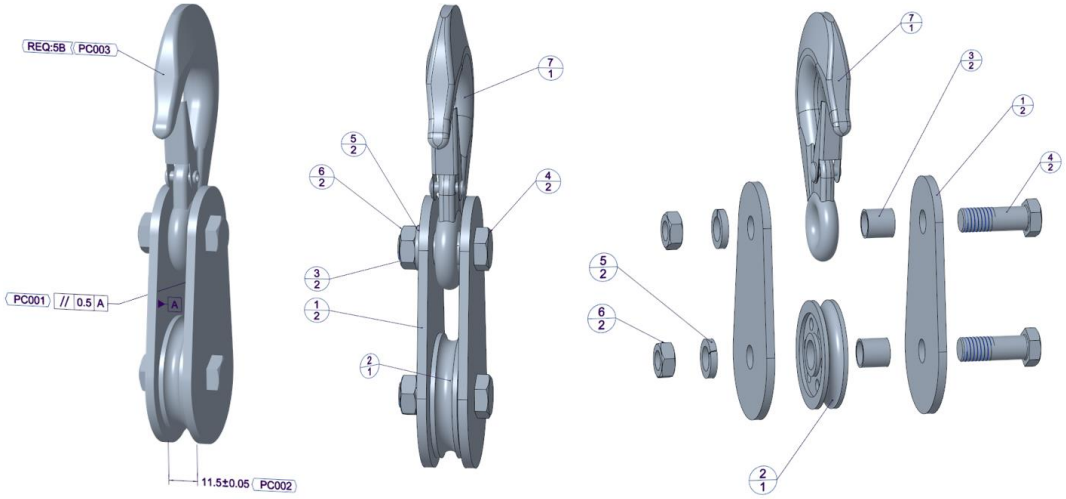


Figure 73: Lifting Block Assembly with Find Numbers

As shown in Table 2, the assembly item list is shown as well as each item's PC Reference Tags. Note that each item's PC Reference Tag list starts with a PC Base Tag identified as PC01.

Find Number	Quantity	Part Number	Description	Number of Prod Chars	Product Characteristics
		9A0814	LIFTING BLOCK ASSEMBLY	2	PC01 PC02
1	2	1A0518	SHELL PLATE	5	PC01 PC02 PC03 PC04 PC05
2	1	2A1025	SHEAVE	8	PC01 PC02 PC03 PC04 PC05 PC06 PC07 PC08
3	2	1A4916	SLEEVE	4	PC01 PC02 PC03 PC04
4	2	FS9-09-G8-S8	HEX BOLT M10-1.5 x 28	0	
5	2	HRSW-M10-Z	SPLIT WASHER M10	0	
6	2	HNN-M10-C8	HEX NUT M10-1.5	0	
7	1	3A1563	HOOK ASSEMBLY	1	PC01

Table 2: Lifting Block Assembly Item List with Product Characteristic Tags

Figure 74 shows the Hook Sub-Assembly which includes a Hook part, Latch Plate part, and Button Head Rivet commercial off the shelf part.

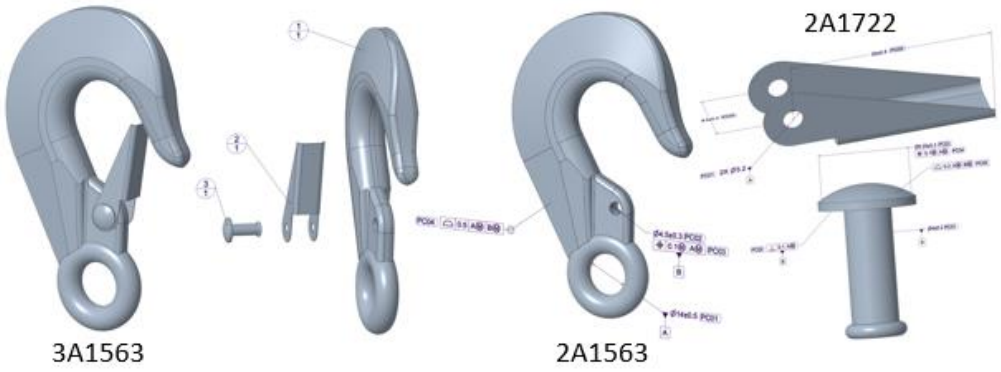


Figure 74: Hook Sub-Assembly with Part Identifiers

As shown in Table 3, the sub-assembly item list is shown as well as each item's PC Reference Tags. Note that each item's PC Reference Tag list starts with a PC Base Tag identified as PC01.

Find Number	Quantity	Part Number	Description	Number of Prod Chars	Product Characteristics
1	1	2A1563	HOOK	4	PC01 PC02 PC03 PC04
2	1	2A1722	LATCH PLATE	3	PC01 PC02 PC03
3	1	AA0531TF	RIVET, BUTTON HEAD 4MM	0	

Table 3: Hook Sub-Assembly Item List with Product Characteristics

5.12.1.2 *Application:*

The top-level separable assembly has a part defining identifier of 9A0814 and it has two PC Reference Tags of <PC001> and <PC002>, and each tag will have their own universally unique PC Reference UUID.

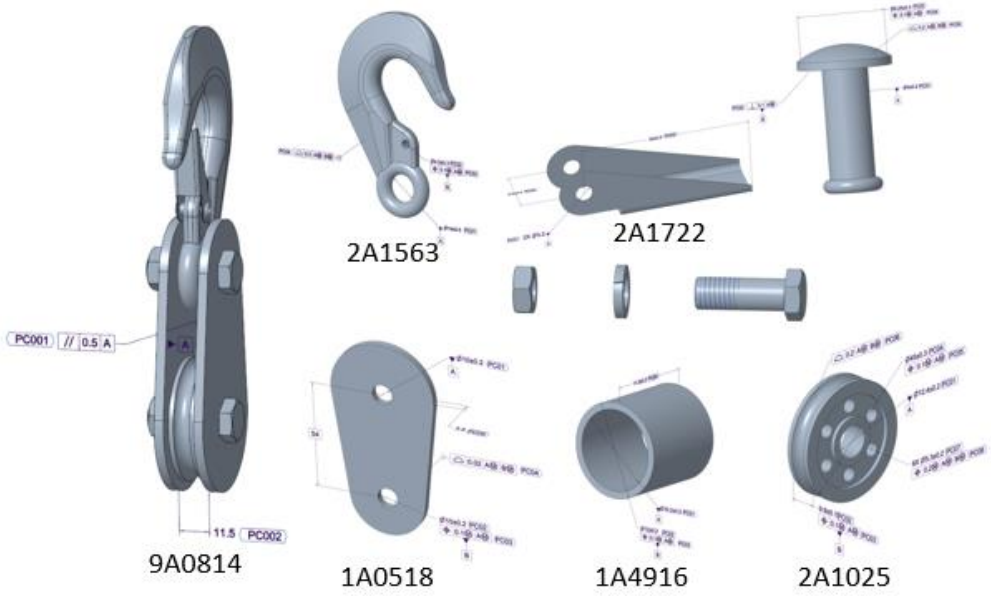


Figure 75: Lifting Block Assembly with Part Identifiers and Product Characteristic Tags

Notice that each component has its own local product characteristics, and the context of each of their PC Tags is within each individual component. Therefore, each component may have identical human-readable tags, for different purposes, with other components within its assembly. Because each component has its own unique part identifier (e.g., 1A3456) enables that the combination of the Part's Identifier with the PC Reference Tag (e.g., 1A3456; <PC007>) uniquely identifies each Product Characteristic. In addition, each PC Reference Tag should have a corresponding machine-readable PC Reference UUID, which will assure a universally unique identifier for each Product Characteristic.

5.12.1.3 *Assembly-Level BOC Example*

Table 4 below shows a BOC of a separable product assembly with Product Identification, 9A0814, which contains multiple components, in which four components, shown below, each have their own PC Reference Tags. Notice that components, 1A0518 and 3A1563 are used two times within the assembly. Notice that the human-readable PC Reference Tag values may be reused for each component; however, each PC Reference Tag has its own machine-readable PC Reference UUID.

Product Name	Product ID	Quantity of Item	PC Tag	PC UUID
Lifting Block	9A0814	N/A	PC01	1E034813-81BD-4D3A-BAB0-79EB6BC1E978
			PC02	3ED37270-CD83-47B2-88D9-573D9059085E
Shell Plate	1A0518	2	PC01	2C888A5E-1BFD-4886-8278-4ACD355B17BC
			PC02	4A9C0034-6D20-474A-AC11-D4F3117EE97D
			PC03	50DF7772-9456-42A1-B237-D7E3B79B84F5
			PC04	A4AFCDAA-4CC9-464B-85AD-942CE8E38260
			PC05	66BCE01C-D75D-4726-8884-57324562663A
Sheave	2A1025	1	PC01	304D0652-B477-4A2C-8934-562708E2099B
			PC02	602F4757-3E84-473E-B479-73A346A6C9CA
			PC03	3DAFDF9C-E708-4FEC-8A20-D90F27A3CF2C
			PC04	CECA88BE-04D4-4C94-9EC7-8FB1ABBD6E72
			PC05	56CB083F-9927-47BA-93CA-80439BF5CC0A
			PC06	FDEA19B3-9F20-4CDC-88D2-CED1DF02334F
			PC07	B0A6EC8A-539F-4C22-92D1-C9B207D143FD
			PC08	8573612F-894E-47EF-84EB-876272A61420
Sleeve	1A4916	2	PC01	CB23952F-0518-1961-0814-39020C268C13
			PC02	651ADED1-FF04-498A-968E-044147A25068
			PC03	0742FBF3-F393-49FE-8370-B6348C387CD5
			PC04	F50417CE-7D8E-4F12-9917-36E79D511D5C
Hook Assembly	3A1563	1	PC01	E7B9AB71-D42F-4AD3-AFFD-FD1DEB8FE460

Table 4: Assembly Level BOC with PC Tags and PC UUIDs

Table 5 below shows a BOC of a separable product sub-assembly with Product Identification, 3A1563, which contains two components, which have their own PC Reference Tags.

Product Name	Product ID	Quantity of Item	PC Tag	PC UUID
Hook Assembly	3A1563	N/A	PC01	9AE6C62D-E4E7-48AD-93C6-5F2B5597C653
Hook	2A1563	1	PC01	BA758934-95B7-4D4A-B29A-A1952BCE210D
			PC02	4700CC60-D002-468B-AE6B-4BCB4DAB985F
			PC03	E1A74099-A4E4-4523-B265-389DEE6FFFD5
			PC04	5FC8E27B-C969-4B6D-8A22-E54612862792
Latch Plate	2A1722	1	PC01	E7BF8961-AAAB-4403-AD6A-69E047B6D16D
			PC02	D73C9317-AC32-4EBE-BBED-82CB209B0976
			PC03	AF3C1B2B-F77B-4B8F-A7D5-B71530E8CDC5

Table 5: Subassembly Level BOC with PC Tags and PC UUIDs

5.12.1.4 *Record of Assembly (ROA) Example*

Table 6 below shows a record of assembly (ROA) example of a separable product assembly with Product Identification, 9A0814, which is built from multiple products, including two 1A0518, one 2A1025, two 1A4916, and one 3A1563. Commercially off the shelf parts was not recorded in this ROA example. Each product has its own set of product characteristics identifiers. After the product is physically realized, a product serial number is assigned. As a result, the human-readable Product Characteristic identification framework shall involve the part identifier, a serial number, and the Product Characteristic tag. Therefore, the verification/measurement result for PC Reference Tag, <PC01> of part 1A4916 with serial number 619 may be explicitly referenced. Similarly, the same verification/measurement result may be explicitly referenced via Part 1A4916, serial number 619 and PC Reference UUID of CB23952F-0518-1961-0814-39020C268C13.

Product Name	Product ID	Product Serial Number	PC Tag	PC UUID
Lifting Block	9A0814	001	PC01	1E034813-81BD-4D3A-BAB0-79EB6BC1E978
			PC02	3ED37270-CD83-47B2-88D9-573D9059085E
Shell Plate	1A0518	109	PC01	2C888A5E-1BFD-4886-8278-4ACD355B17BC
			PC02	4A9C0034-6D20-474A-AC11-D4F3117EE97D
			PC03	50DF7772-9456-42A1-B237-D7E3B79B84F5
			PC04	A4AFCDAA-4CC9-464B-85AD-942CE8E38260
			PC05	66BCE01C-D75D-4726-8884-57324562663A
		524	PC01	2C888A5E-1BFD-4886-8278-4ACD355B17BC
			PC02	4A9C0034-6D20-474A-AC11-D4F3117EE97D
			PC03	50DF7772-9456-42A1-B237-D7E3B79B84F5
			PC04	A4AFCDAA-4CC9-464B-85AD-942CE8E38260
			PC05	66BCE01C-D75D-4726-8884-57324562663A
Sheave	2A1025	814	PC01	304D0652-B477-4A2C-8934-562708E2099B
			PC02	602F4757-3E84-473E-B479-73A346A6C9CA
			PC03	3DAFDF9C-E708-4FEC-8A20-D90F27A3CF2C
			PC04	CECA88BE-04D4-4C94-9EC7-8FB1ABBD6E72

			PC05	56CB083F-9927-47BA-93CA-80439BF5CC0A
			PC06	FDEA19B3-9F20-4CDC-88D2-CED1DF02334F
			PC07	B0A6EC8A-539F-4C22-92D1-C9B207D143FD
			PC08	8573612F-894E-47EF-84EB-876272A61420
Sleeve	1A4916	619	PC01	CB23952F-0518-1961-0814-39020C268C13
			PC02	651ADED1-FF04-498A-968E-044147A25068
			PC03	0742FBF3-F393-49FE-8370-B6348C387CD5
			PC04	F50417CE-7D8E-4F12-9917-36E79D511D5C
		620	PC01	CB23952F-0518-1961-0814-39020C268C13
			PC02	651ADED1-FF04-498A-968E-044147A25068
			PC03	0742FBF3-F393-49FE-8370-B6348C387CD5
			PC04	F50417CE-7D8E-4F12-9917-36E79D511D5C
Hook Assembly	3A1563	319	PC01	E7B9AB71-D42F-4AD3-AFFD-FD1DEB8FE460

Table 6: Record of Assembly Example with PC Tags and PC UUIDs

5.12.2 Inseparable Assembly

PC Reference Tags within an inseparable assembly are typically addressed similarly to those of an individual piece-part, since one cannot separate the assembly into its constituents.

5.13 Product Characteristics in Support Documents (Text-based support artifacts)

Many organizations utilize one or more product specification documents containing verification requirements to complete their product definition dataset. As a result, verification requirements contained in these product definition support documents shall be identified with a Product Characteristic designator, be identified within the BoC, and require a quality verification plan.

5.13.1 Example

Often non-graphical product specifications are captured as verification requirements within supplemental product definition documents. These specifications are as important as those product characteristics that specify form, fit and function placed upon the drawing or model. Some examples of supplemental product definition documents that contain product verification requirements include:

- Weld Specifications
- Workmanship Specifications
- Marking Specifications
- Cleaning Specifications
- Special Acceptance Equipment definitions
- Record of Assembly Specification
- Packaging Specifications
- Weight Requirements

5.13.2 Application:

Some examples of verification requirements, and therefore product characteristics, which may be found within product definition support documents, could be:

- Verify workmanship per specification 9900000; Paragraph 3.3.4.
- Marking shall be per C&J's Etching specification 19610518 Revision B, Requirement: 18.
- Inspect Welds per 9912121.
- Verify five welds between gear Shaft and Clock-Plate per SS7A9944 (Laser Welding Requirement Paragraph 5.1 Visual Acceptance Criteria) using 7-20 X magnification, visually compared to sample.
- Clean parts per PES P1241234.
- Maximum dry weight shall be 3.25 lbs. per product specification PW1025.

5.13.3 BOC Example

Below is a BOC table example of product characteristics identified from verification requirements specified within supplemental product definition documents.

Spec Doc	Tag (Base)	Tag (Ext)	UUID (Base)	UUID (Ext)	Description
9900000	PC900	N/A	EF6C708F-0637-4D90-8594-1E9E59349B17	N/A	Verify workmanship per specification 9900000; Para 3.3.4
19610518	PC901	N/A	CA8EF89C-A3DE-4CE9-B917-00E70C91812B	N/A	Verify marking per specification 19610518 Rev. B, Requirement 18
9912121	PC902	N/A	5358EE09-D952-47CF-A99F-4754A2A05333	N/A	Inspect Weld per 9912121
SS7A9944	PC903	1	5C2B3DE4-B588-4409-96FF-4372E1EAC3C0	1	Verify weld 1 per SS7A9944; Para 5.1 using 7-20X magnification
SS7A9944	PC903	2	5C2B3DE4-B588-4409-96FF-4372E1EAC3C0	2	Verify weld 2 per SS7A9944; Para 5.1 using 7-20X magnification
SS7A9944	PC903	3	5C2B3DE4-B588-4409-96FF-4372E1EAC3C0	3	Verify weld 3 per SS7A9944; Para 5.1 using 7-20X magnification
SS7A9944	PC903	4	5C2B3DE4-B588-4409-96FF-4372E1EAC3C0	4	Verify weld 4 per SS7A9944; Para 5.1 using 7-20X magnification
SS7A9944	PC903	5	5C2B3DE4-B588-4409-96FF-4372E1EAC3C0	5	Verify weld 5 per SS7A9944; Para 5.1 using 7-20X magnification
P1241234	PC904	N/A	2C494A67-8320-4C77-A2FE-90D349EAAC52	N/A	Verify parts were cleaned per PES P1241234
PW1025	PC905	N/A	F89CCA8E-A3DE-B917-4CE9-0C91812B00E7	N/A	Verify part dry weight is less than 3.25 lbs. per PW1025 Req. 3.b

Table 7: BOC from Supplemental Product Definition Documents

5.14 Product Characteristics in System Information Models

Soon, systems or products may be described by the SysML. Requirements Diagrams provide modeling constructs to represent text-based requirements and relate them to other modeling elements. A Requirement specifies a capability or a condition that should be satisfied. Within SysML, one could identify a Product Requirement, which may lead to one-to-many Verification Requirements. Each Verification Requirement is satisfied by a Product Characteristic designator with its attributes and may have a criticality classification. Furthermore, the Verification Requirement may be verified by a quality verification plan, which helps determine if the system fulfills the requirement.

5.14.1 Example

Figure 76 below is an example of a SysML Requirements diagram that shows elements that directly relate to concepts within this standard.

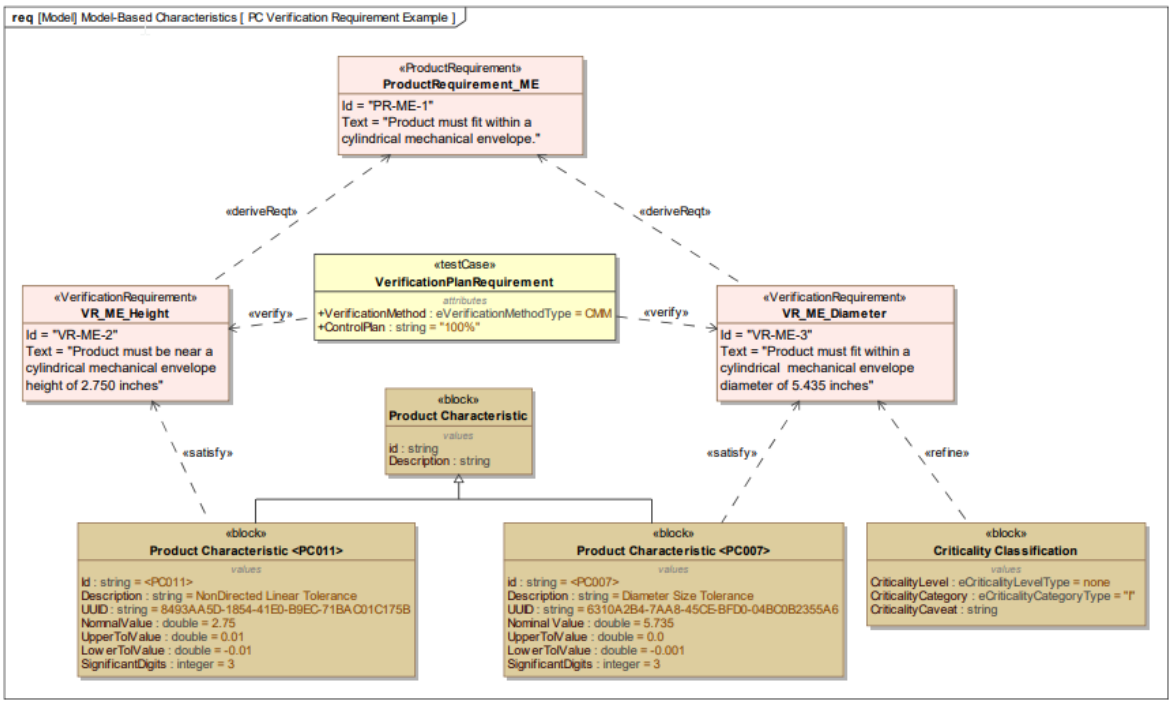


Figure 76: Requirements Diagram shown Requirements Satisfied identify by Product Characteristics

5.14.2 Application:

From the SysML Requirement diagram above, there is a Product Requirement for a mechanical envelope with identifier of PR-ME-1. Then there are two Verification Requirements, VR-ME-2, and VR-ME-3 respectively, which are derived from the Product Requirement. Each Verification Requirement contains a specific dimensional aspect, height, and diameter respectively of the Product Requirement. One Verification Requirement, VR-ME-3, has a Criticality Category of "I = Interface". Then each Verification Requirement may be satisfied by a Product Characteristic, <PC011> & <PC007> respectively. Each Product Characteristic contains specific attribute data including Description, UUID, Nominal Value, and Upper & Lower Tolerance Values. Finally, a Verification Plan Requirement of CMM with 100% sampling plan verifies each of the verification requirements. For this standard, we may interpret the Requirement diagram as:

- <PR-ME-1<<PC011>/CMM: 100%/
- <PR-ME-1<<PC007>/CMM: 100%/

5.15 Product Characteristics on Surrogate Surfaces

Product Characteristics may be applied to annotations that are associated with surrogate or consumable surfaces that may be found on a casting, forgings, moldings, additive manufacturing surfaces, or in-process shapes that are removed during post machining operations.

5.16 Product Characteristics on Representative Test Artifacts

Product Characteristics may be applied to annotations or verification requirements that are associated with representative test coupons. These test artifacts are intended to represent the actual product in processes such as additive manufacturing or welding.

5.17 Product Characteristics on Embellished Derivatives

Product Characteristics may be applied to annotations or verification requirements that are associated with an embellished derivative. These embellished derivatives are intended to represent the in-process or synthetic derivatives that support test or manufacturing.

5.18 Application to Serialized Parts which enables Digital Twins

The product realization process includes both designing a well-defined product and producing an acceptable product. The product realization process' design activity describes a product definition that includes verification requirements (aka product characteristics) that shall be satisfied for an acceptable product. The product realization process' production activity produces the parts and provides evidence that the part is acceptable according to the product-definition's verification requirements.

Upon the physical realization of each product, a serial number is typically assigned to each realized product.

5.18.1 Enables Digital Twins

Product end-items are identified with product identifiers (e.g., 1A3456-00, 1610518). Once the product definition gets physically realized as one or more actual parts, each physical part often receives a part serial number (e.g., SN 1025) and/or gets associated with a manufacturing lot identification (e.g., 20230305). Since this standard extends the product definition with Product Characteristic Tags and Product Characteristic UUIDs information and enables combining the Product Characteristic identification with the physical product identifier (e.g., serial number), the standard increases the precision and ease of connecting serialized digital twins.

5.18.2 Serialized Part Examples

The table below show an example of Product Identification, 1A3456, which has multiple PC Reference Tags and then was realized with multiple serial numbers, 101, 102, 103, and 104. The data in the Serial Number columns represent measurement results for each serial number of each PC Tag.

Product ID	PC Tag	PC UUID	Serial Number			
			101	102	103	104
1A3456	PC007	C83F6F4B-585F-4111-A986-3B61BE525412	.001	.001	.002	.002
	PC008	4CD78C49-A504-4D73-93CA-9B8642CF0C56	.010	.011	.012	.009
	PC009	D507C8DB-35F7-46C1-9AD6-0173D36FD6A7	T	T	F	T

5.19 Non-Product Characteristics, General Tag

Product Definitions will contain verification requirement annotations (e.g., tolerances, specifications) that are designated as Product Characteristics. However, Product Definitions may also contain other annotations (e.g., BASIC dimensions) that are NOT verification requirements, but may need to be designated for other purposes. They should have a different designation with a different prefix. A non-product characteristics identifier should use a General Tag "GT" prefix, as all that is required is a means to provide a reference to the annotation. One use case might be to explicitly designate a non-product characteristics annotation as a change control object toward

supporting a problem report or change request. Such as, “Request change the radius of <GT5026> from R 1.0 to R 1.5.

Format: <GT”numeric”>

Example: <GT500>

5.19.1 Example

Below shows the use case of two non-product characteristics identifiers, which are labeled as General Tag GT5025 and GT5026 for the BASIC dimensions of 30° and Ø4.500 respectively. These tags are highlighted in red in the figure below.

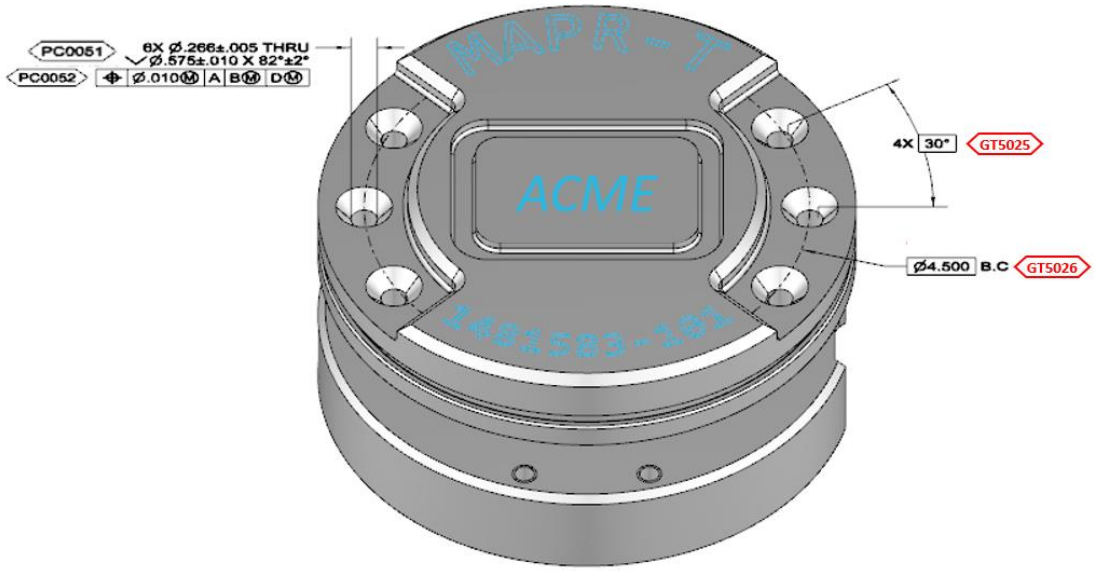


Figure 77: General Tags for Non-Product Characteristic Annotations

6 Criticality Classifications (CC) Augmentation

A Product Characteristic may be augmented by zero-to-many Criticality Classifications. Each Criticality Classification communicates a unique instance of a Criticality of interest for the Product Characteristic. Within the context that a Product Characteristic is augmented by a Criticality Classification, that Product Characteristic may be called a Critical Product Characteristic.

Criticality classifications are typically set based upon product requirements and are augmented during the prepare product definition phase.

6.1 Identification

The CriticalityClassification (CC) is identified by a CriticalityClassificationDesignator. The CriticalityClassificationDesignator is the identifying text that is optionally housed within the CriticalityClassificationDesignatorSymbol. The CriticalityClassificationDesignator is built from attributes values contained within CriticalityLevel, CriticalityCategory, and/or CriticalityCaveat.

6.2 Data Structure

The information model for an optional CriticalityClassification augmentation for a ProductCharacteristic is graphically shown below in Figure 78.

Each CriticalityClassification will have a CriticalityClassificationDesignator. The CriticalityClassificationDesignator is constructed from a CriticalityLevel and/or CriticalityClass with an optional CriticalityCaveat.

6.3 Data Taxonomy

The CriticalityClassification taxonomy consists of Levels, Categories, and Caveats along with a designator and symbol. The CriticalityClassification is identified by a CriticalityClassificationDesignator that should be shown within a CriticalityClassificationDesignatorSymbol.

The CriticalityClassification shall contain a CriticalityClassificationDesignator built from the CriticalityClass that is either a CriticalityLevel or a CriticalityCategory or both. The CriticalityLevel provides a level of severity for the category and is represented by an enumeration of CriticalityLevel types as discussed in section 6.4.5. The CriticalityCategory provides a type of criticality and is represented by an enumeration of CriticalityCategory types as discussed in section 6.4.6. In addition, the CriticalityCaveat extends the CriticalityClass with a free form specific reference, stipulations, conditions, or limitations.

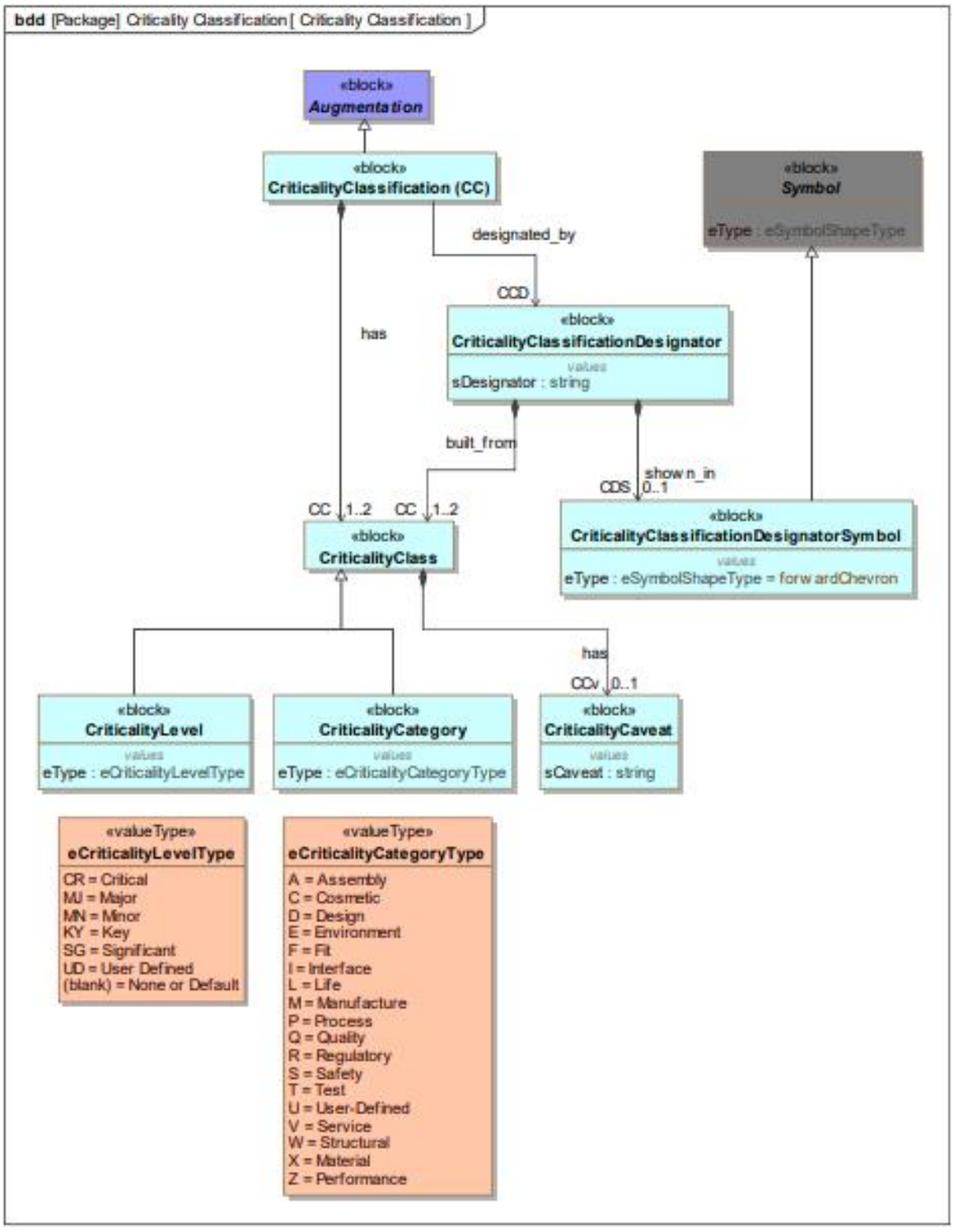


Figure 78: Block Diagram - Criticality Classifications Information Model

6.4 Data Objects

6.4.1 CriticalityClassification

The optional CriticalityClassification augmentation provides a Product Characteristic with a criticality designation that consists of CriticalityClass that allows for either or both a CriticalityLevel and/or CriticalityCategory, with an optional CriticalityCaveat.

6.4.2 CriticalityClassificationDesignator

The CriticalityClassificationDesignator element provides the textual identification for the CriticalityClassification, which is housed by the CriticalityClassificationDesignatorSymbol. The CriticalityClassificationDesignator consists of ordered information from CriticalityLevel, CriticalityCategory, and CriticalityCaveat, respectively. The CriticalityClassificationDesignator shall include either a CriticalityLevel or a CriticalityCategory or both.

If both the CriticalityLevel and CriticalityCategory are designated, then their type designators are separated by a colon, ":".

If a CriticalityCaveat modifies a CriticalityCategory, then the CriticalityClass type designator and CriticalityCaveat text are separated by a period, ".".

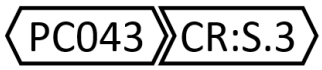
6.4.3 CriticalityClassificationDesignatorSymbol

The optional CriticalityClassificationDesignatorSymbol element defines a human-readable symbol for housing the CriticalityClassificationDesignator element.

The recommended symbol for housing a CriticalityClassificationDesignator (e.g., S) is with an elongated forward chevron. The elongated forward chevron symbol shall accommodate the CriticalityClassificationDesignator and provide visual connection to the ProductCharacteristicDesignator symbol.

A simple CriticalityClassificationDesignatorSymbol as a CriticalityClassificationDesignator is shown below.

An example of a simple CriticalityClassificationDesignatorSymbol associated with a ProductCharacteristicDesignatorSymbol is shown below. The CriticalityClassificationDesignatorSymbol should follow after and in-line with the ProductCharacteristicDesignatorSymbol.



6.4.4 CriticalityClass

The CriticalityClass element designates a CriticalityLevel and/or CriticalityCategory. The CriticalityClass has an optional CriticalityCaveat.

6.4.5 CriticalityLevel

The optional CriticalityLevel element is the designated criticality level for the CriticalityClassification. The CriticalityLevel has the following CriticalityLevel types:

- CR: Critical
- MJ: Major
- MN: Minor
- KY: Key
- SG: Significant
- UD: User-Defined
- Blank: None or default per organizational practices

In general, the Critical, Major, and Minor criticality groups are design criticalities that have an effect on product operations such that failure to satisfy the characteristic may cause failure of the item or mission. By contrast, the Key and Significant criticality group are process criticalities that will have a direct effect on costs, fit, properties, cosmetics, or manufacturing.

A criticality level may be associated with a criticality category.

6.4.5.1 *Critical (CR)*

A design characteristic of the material, part, or component whose potential deviation from the tolerance value may lead to a failure that has lethal consequences.

CR - Critical to compliance with government regulations or safety.

Where analysis indicates that if defective, it would:

- Create or increase a hazard to human safety
- Result in failure of a weapon, or
- Prevent a major system from performing a required mission.

6.4.5.2 *Major (MJ)*

A design characteristic, other than critical, that shall be met to avoid failure or material reduction of usability of the unit of product for its intended purpose.

MJ – Major effect on the operation of the product

Where analysis indicates that if defective, it would result in failure of an end item to perform a required mission.

6.4.5.3 *Minor (MN)*

A design characteristic, other than critical or major, whose departure from its specification requirement is not likely to reduce materially the usability of the unit of product for its intended purpose or whose departure from established standards has little bearing on the effective use or operation of the unit.

MN – Minor effect

Where analysis indicates that if defective, it would impair the mission performance of the item.

6.4.5.4 *Key (KY)*

The features of a material, part, or component whose potential variation from manufacturing process has a direct effect on product cost, fit, properties, cosmetics, or service life.

KY – Key

6.4.5.5 *Significant (SG)*

The features of a material, part, or component whose potential deviation from the tolerance valued may result in the product not functioning, or inability to continue production. In addition, any design features whose variation has the greatest impact on the fit, performance or service life of the finished product, from the perspective of the customer or the ability to process / build the product.

SG – Significant for customer satisfaction

6.4.5.6 *User-Defined (UD)*

A criticality level defined by the authoring organization.

UD – User-Defined

Format: UD:"String"

6.4.5.7 *Blank*

A criticality level that is left blank by the authoring organization.

Blank – None or defaults per organizational business practices

6.4.6 CriticalityCategory

The optional CriticalityCategory element is the designated criticality category for the CriticalityClassification. The CriticalityCategory list is shown below:

- A: Assembly
- C: Cosmetic
- D: Design
- E: Environmental
- F: Fit
- I: Interface
- L: Life
- M: Manufacture
- P: Process
- Q: Quality
- R: Regulatory
- S: Safety
- T: Test
- U: User-Defined
- V: Service
- W: Structural

- X: Material
- Z: Performance

A criticality category may be associated with a criticality level.

The CriticalityCategory has the following CriticalityCategory types:

6.4.6.1 *Assembly (A)*
A – A category related to an assembly aspect of the component.

6.4.6.2 *Cosmetic (C)*
C – A category related to a cosmetic aspect of the component.

6.4.6.3 *Design (D)*
D – A category related to a design aspect of the component.

6.4.6.4 *Environmental (E)*
E – A category related to an environmental aspect of the component.

6.4.6.5 *Fit (F)*
F – A category related to a fit aspect of the component.

6.4.6.6 *Interface (I)*
I – A category related to an interface aspect of the component.

6.4.6.7 *Life (L)*
L – A category related to the life of the component.

6.4.6.8 *Manufacture (M)*
M – A category related to a manufacturing aspect of the component.

6.4.6.9 *Process (P)*
P – A category related to a process aspect of the component.

6.4.6.10 *Quality (Q)*
Q – A category related to a quality aspect of the component.

6.4.6.11 *Regulatory (R)*
R – A category related to a regulatory aspect of the component.

6.4.6.12 *Safety (S)*
S – A category related to a safety aspect of the component.

6.4.6.13 *Test (T)*
T – A category related to a testing aspect of the component.

6.4.6.14 *User-Defined (U)*
U – A category defined by the user organization for the component.

Format: U:”string”

6.4.6.15 *Service (V)*

V – A category related to a service aspect of the component.

6.4.6.16 *Structural (W)*

W – A category related to a structural aspect of the component.

6.4.6.17 *Material (X)*

X – A category related to a material aspect of the component.

6.4.6.18 *Performance (Z)*

Z – A category related to a performance aspect of the component.

6.4.7 CriticalityCaveat

The optional CriticalityCaveat element modifies a CriticalityClass subtype of CriticalityLevel or CriticalityCategory with a free form referenceable specifier that adds additional information. For an example of >S.518> the "518" would be the CriticalityCaveat of CriticalityCategory "S" and the user would be able to look up and determine the information associated with "518" within the context of the product or company source.

6.4.8 CriticalityClassificationText

The recommended textual format for a CriticalityClassificationDesignator is with a "greater than" character preceding and a "greater than" character following the Designator. The general format is > Criticality Designation >. Some simple textual CriticalityClassificationDesignator are shown below:

Description	**Example**
CriticalityLevel and CriticalityCategory:	>CR: S>
CriticalityLevel only:	>CR>
CriticalityCategory only:	>S>
CriticalityCategory with CriticalityCaveat	>S.518>
CriticalityLevel and CriticalityCategory with CriticalityCaveat:	>CR:S.518>
User Defined CriticalityLevel and User Defined CriticalityCategory	>UD:KR.U:Thermo>
User Defined CriticalityLevel and User Defined CriticalityCategory	>UD:KR.U:"Thermo Queen">

Table 8: Textual Criticality Classification Designators

An example of a textual CriticalityClassificationDesignator, >S> associated with a ProductCharacteristicDesignator, <PC007> is shown below:

<PC007>>S>

7 Product Requirement Association (PRA) Augmentation

A Product Characteristic may be augmented by zero-to-many Product Requirement Associations. Each Product Requirement Association references a product requirement that drives this characteristic. Within the context that a Product Characteristic is augmented by a Product Requirement Association, that Product Characteristic may be called a Driven Characteristic.

A product requirement addresses a product's purpose, its features, functionalities, and behavior.

7.1 Identification

The ProductRequirementAssociation (PRA) is identified by a ProductRequirementAssociationDesignator. The ProductRequirementAssociationDesignator is the identifying text that is optionally housed within a ProductRequirementAssociationDesignatorSymbol. The ProductRequirementAssociationDesignator has a free-form designator string.

7.2 Data Structures

The information model for an optional ProductRequirementAssociation augmentation for a ProductCharacteristic is graphically shown below in Figure 79.

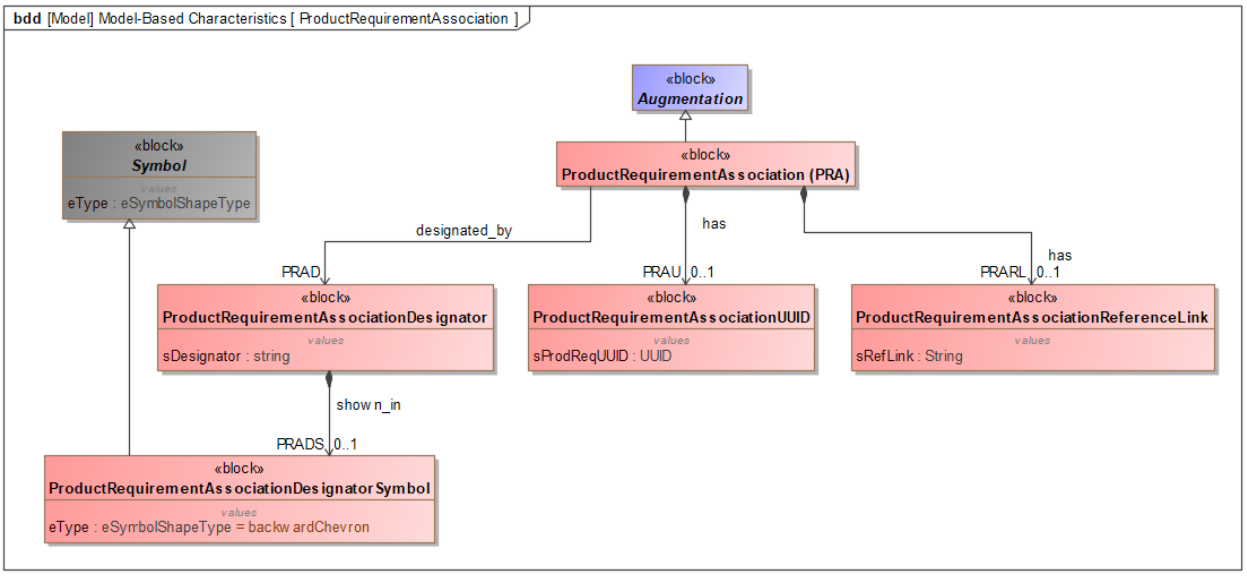


Figure 79: Block Diagram - Product Requirement Associations Information Model

7.3 Data Taxonomy

The ProductRequirementAssociation taxonomy consists of Reference Links and Universally Unique Identifiers along with a designator and symbol. The ProductRequirementAssociation is identified by a ProductRequirementAssociationDesignator that should be shown within a ProductRequirementAssociationDesignatorSymbol.

The ProductRequirementAssociation shall contain a ProductRequirementAssociationDesignator and either or both a ProductRequirementAssociationReferenceLink to the product requirement object or ProductRequirementAssociationUUID of the product requirement object. The ProductRequirementAssociationReferenceLink should be a direct link to the Product Requirement

object whereas the ProductRequirementAssociationUUID is the identifier that a user may use to search for a Product Requirement within a requirement management system.

7.4 Data Objects

7.4.1 ProductRequirementAssociation

The optional ProductRequirementAssociation augmentation provides a Product Characteristic an association with a product requirement object. The association is via a reference link and/or universally unique identifier.

7.4.2 ProductRequirementAssociationDesignator

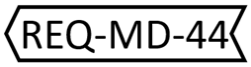
The ProductRequirementAssociationDesignator element provides the textual identification for the Product Requirement, which is housed by the ProductRequirementAssociationDesignator Symbol. The ProductRequirementAssociationDesignator has a string attribute that allows for a human-readable identifier of the associated Product Requirement.

7.4.3 ProductRequirementAssociationDesignatorSymbol

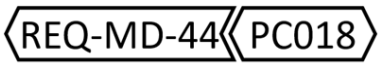
The optional ProductRequirementAssociationDesignatorSymbol element is a human-readable symbol for housing the ProductRequirementAssociationDesignator.

The recommended symbol for identifying a ProductRequirementAssociationDesignator is an elongated backward chevron. The elongated backward chevron symbol shall accommodate the ProductRequirementAssociationDesignator and provide a visual connection to the ProductCharacteristicDesignatorSymbol.

A simple ProductRequirementAssociationDesignatorSymbol with a ProductRequirementAssociationDesignator is shown below:



An example of a ProductRequirementAssociationDesignatorSymbol associated with a ProductCharacteristicDesignatorSymbol is shown below. The ProductRequirementAssociationDesignatorSymbol is a pre-augmentation symbol that precedes the ProductCharacteristicDesignatorSymbol.



7.4.4 ProductRequirementAssociation Text

The recommended textual format for a ProductRequirementAssociationDesignator is with a “less than” character preceding and a “less than” character following the Tag. The general format is < ProductRequirementAssociationDesignator <. A simple textual ProductRequirementAssociationDesignator is shown below:

< REQ-MD-44 <

An example of a textual ProductRequirementAssociationDesignator associated with a ProductCharacteristicDesignator symbol is shown below:

< REQ-MD-44 < < PC018>

An example of a textual ProductRequirementAssociationDesignator associated with a ProductCharacteristicDesignator symbol along with a CriticalityClassificationDesignator is shown below:

< REQ-MD-44 < < PC007> >S>

7.4.5 ProductRequirementAssociationUUID

The optional ProductRequirementAssociationUUID element defines a machine-readable universally unique identifier of the Product Requirement within the references source.

7.4.6 ProductRequirementAssociationReferenceLink

The optional ProductRequirementAssociationReferenceLink element defines a reference link (e.g., URL) to the Product Requirement within the references source.

8 Verification Plan Requirement (VPR) Augmentation

A Product Characteristic may be augmented by zero-to-many Verification Plan Requirements. The Verification Plan Requirement communicates product verification information for the Product Characteristic. Within the context that a Product Characteristic is augmented by a Verification Plan Requirement, that Product Characteristic may be called a Verification Characteristic. The VerificationPlanRequirement is usually determined during the quality engineering activity and is performed referencing the source product definition. Once completed, the source product definition has been embellished with verification plan requirements, resulting in an embellished derivative.

Furthermore, the quality engineering function typically embellishes the source product definition (e.g., MBD Model) with Verification Plan Requirements, which produces a quality-embellished derivative model.

8.1 Identification

A VerificationPlanRequirement is identified by a VerificationPlanRequirementDesignator. The VerificationPlanRequirementDesignator is the identifying text that is optionally housed within a VerificationPlanRequirementDesignatorSymbol.

8.2 Data Structures

The information model for an optional VerificationPlanRequirement augmentation for a ProductCharacteristic is graphically shown below in Figure 80.

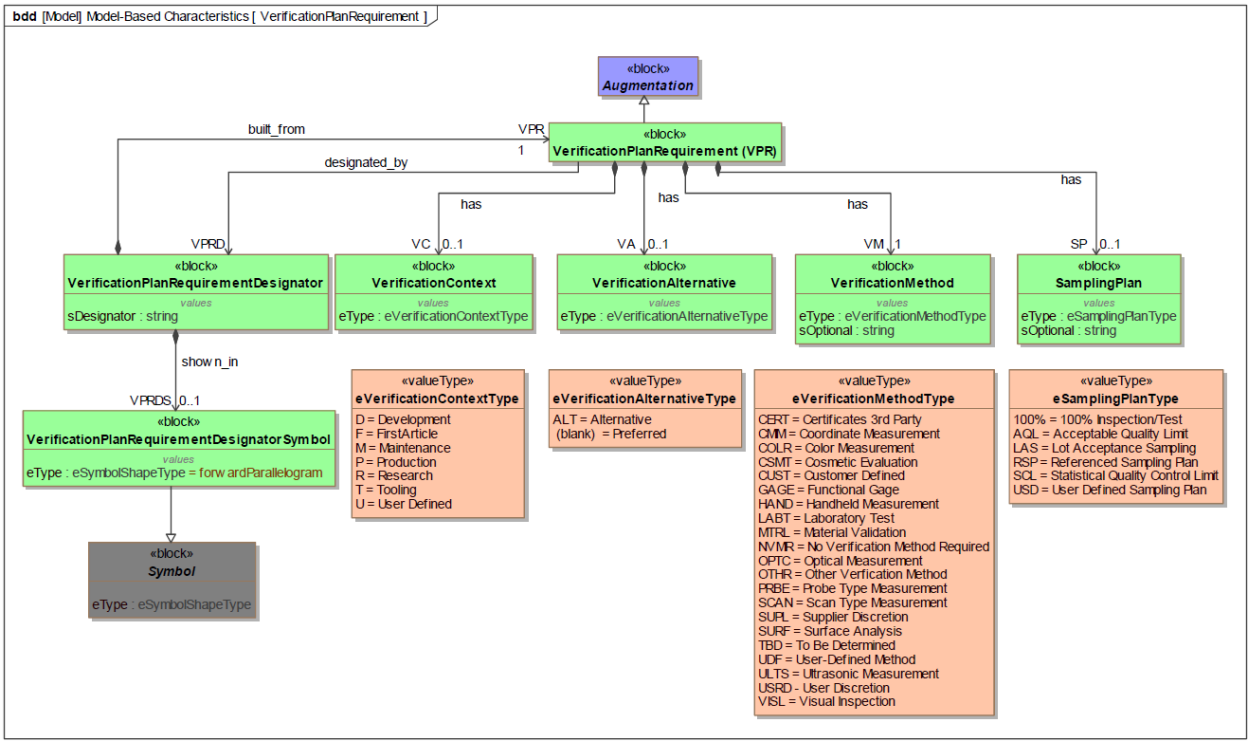


Figure 80: Block Diagram - Verification Plan Requirement Information Model

8.3 Data Taxonomy

The VerificationPlanRequirement taxonomy consists of a verification method, context, alternative, and sampling plan along with a designator and a symbol. The VerificationPlanRequirement is designated by a VerificationPlanRequirementDesignator that should be shown within a VerificationPlanRequirementDesignatorSymbol.

The VerificationPlanRequirement shall contain a VerificationMethod as part of the VerificationPlanRequirementDesignator. The VerificationMethod should allow an optional VerificationContext, an optional VerificationAlternative, and/or an optional SamplingPlan. The VerificationPlanRequirementDesignator will show the optional VerificationContext type, optional VerificationAlternative type, the required VerificationMethod type and the optional SamplingPlan type.

The VerificationMethod is represented by an enumeration of VerificationMethod types referenced in section 8.4.5. Likewise, the VerificationContext is represented by an enumeration of VerificationContext types as shown in section 8.4.6. Then the VerificationAlternative is blank for preferred or "ALT" as an alternate as discussed in section 8.4.7. In addition, the SamplingPlan is represented by an enumeration of SamplingPlan types as discussed in section 8.4.8.

8.4 Data Objects

8.4.1 VerificationPlanRequirement

The optional VerificationPlanRequirement augmentation provides a ProductCharacteristic with a verification plan designation.

8.4.2 VerificationPlanRequirementDesignator

The VerificationPlanRequirementDesignator element provides the textual identification for the VerificationPlanRequirement, which is housed by the VerificationPlanRequirementDesignatorSymbol. The VerificationPlanRequirementDesignator consists of ordered information from VerificationContext, VerificationAlternative, VerificationMethod, and SamplingPlan. The VerificationPlanRequirementDesignator shall include a VerificationMethod. The VerificationContext, VerificationAlternative, and SamplingPlan are optional.

The VerificationPlanRequirementDesignator has the following format between two forward slashes.

> Format: / opt{opt{*VerificationContext*}+opt{-*VerificationAlternative*}:}+ *VerificationMethod* +opt{:*SamplingPlan*}/

The format contains an optional *VerificationContext*, then an optional dash with VerificationAlternative and colon, then a required *VerificationMethod*, and then an optional colon *SamplingPlan*.

A VerificationPlanRequirementDesignator sample shown below, communicates that for a D = Development Verification Context, that an ALT = VerificationAlternative is the VerificationMethod of HAND = handheld measurement method with a sampling plan of 100%.

> /D-ALT: HAND: 100%/

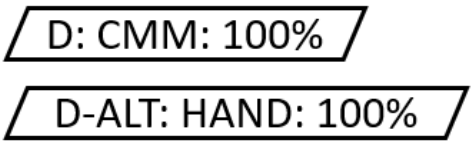
8.4.3 VerificationPlanRequirementDesignatorSymbol

The optional VerificationPlanRequirementDesignatorSymbol element is a human-readable symbol for housing the VerificationPlanRequirementDesignator element.

The recommended symbol for housing a VerificationPlanRequirementDesignator (e.g., CMM) is with an elongated forward parallelogram. The elongated forward parallelogram symbol shall accommodate the VerificationPlanRequirementDesignator and provide visual connection to the VerificationPlanRequirementDesignatorSymbol.

A simple VerificationPlanRequirementDesignatorSymbol with a VerificationPlanRequirementDesignator for only a VerificationMethod is shown below.

Additional examples of VerificationPlanRequirementDesignator and/or VerificationPlanRequirementDesignatorSymbol are:



8.4.4 VerificationPlanRequirementText

The recommended textual format for a VerificationPlanRequirementDesignator is with a “forward slash” character preceding and a “forward slash” character following the Designator. The general format is / VerificationPlanRequirementDesignator /. Some simple textual VerificationPlanRequirementDesignator examples are shown below:

- /CMM/
- /GAGE: ”CE53742”/
- /D: CMM: 100%/
- /D-ALT: HAND: 100%/
- /P: CMM: SP: 67%/
- /D: USRD:”GE1”: 100%/

8.4.5 VerificationMethod

The VerificationMethod for the VerificationPlanRequirement is the approach to verifying a product characteristic. Each verification method may have an optional string with more specifics. The format of the VerificationMethod is shown below. Whereas Method is one of the following verification method types, and the “string” is optional and freeform. The format of the Verification Plan Requirement with the required Verification Method is shown below.

Format: /*Method* opt{: “string”}/

The verification method shall be designated as one of the following verification method types:

8.4.5.1 *Certificates (CERT)*

CERT – Certificates: 3rd party documents

8.4.5.2 *Coordinate Measurement (CMM)*
CMM – Coordinate Measurement Machine

8.4.5.3 *Color Measurement (COLR)*
COLR – Color Measurement & Appearance Equipment

8.4.5.4 *Cosmetic Evaluation (CSMT)*
CSMT – Cosmetic Evaluation

8.4.5.5 *Customer Defined (CUST)*
CUST – Customer Defined

8.4.5.6 *Functional Gage (GAGE)*
GAGE – Functional Gage

8.4.5.7 *Handheld Measurement (HAND)*
HAND – Handheld Measurement Tools

8.4.5.8 *Laboratory Test (LABT)*
LABT – Laboratory Test

8.4.5.9 *Material Validation (MTRL)*
MTRL – Material Validation

8.4.5.10 *No Verification Method Required (NVMR)*
NVMR – No Verification Method Required

8.4.5.11 *Optical Measurement (OPTC)*
OPTC – Optical Measurement Equipment

8.4.5.12 *Other Verification Method (OTHR)*
OTHR – Other Verification Method

8.4.5.13 *Probing Measurement (PRBE)*
PRBE – Probe Type Measurement Equipment

8.4.5.14 *Scanning Measurement (SCAN)*
SCAN – Scan Type Measurement Equipment

8.4.5.15 *Supplier Discretion (SUPL)*
SUPL – Supplier Discretion

8.4.5.16 *Surface Analysis Measurement (SURF)*
SURF – Surface Analysis Equipment

8.4.5.17 *To Be Determined (TBD)*
TBD – To Be Determined

8.4.5.18 *User-Defined Method (UDF)*
UDF – User Defined Verification Method; Always includes a user defined “string” in quotes to designate that it is a free-form string. (i.e., “Joe’s Tester”

Format: UDF: ”string”

8.4.5.19 *Ultrasonic Measurement (ULTS)*

ULTS – Ultrasonic Measurement Equipment

8.4.5.20 *User Discretion/ Standard Practice (URSD)*

USRD – User Discretion

8.4.5.21 *Visual Inspection (VISL)*

VISL– Visual Inspection

8.4.5.22 *Verification Method Example*

Examples of Verification Plan Requirements with different Verification Methods are shown below. The optional free-form string method examples include a GAGE method, which specifies a specific functional gage GA53742, a UDF method that designates the ACME Manufacturing’s Wile-E Method, and a SCAN method that designates CT (computerized tomography) scanning.

/CMM/

/CMM: “CMM4449”/

/TBD/

/GAGE: ”GA53742”/

/UDF: ”ACME_Wile-E_Method”/

/SCAN: ”CT”/

8.4.6 VerificationContext

A verification method may be applied to a specific context (e.g., D, F, P, R, and T). The format of the Verification Plan Requirement with the required Verification Method and a Verification Context is shown below.

Format: /***Context***: *Method* /

The following are the verification context types.

8.4.6.1 *Development (D)*

D = Development

8.4.6.2 *First Article (F)*

F = First Article

8.4.6.3 *Maintenance (M)*

M = Maintenance

8.4.6.4 *Production (P)*

P = Production

8.4.6.5 *Research (R)*

R = Research

8.4.6.6 *Tooling (T)*

T = Tooling

8.4.6.7 *User-Defined (U)*

U = User-Defined
Format: U:"string"

8.4.6.8 *Verification Context Example*

Examples of Verification Plan Requirements with different Verification Context are shown below. First, the Development Verification Context is by the HAND - Handheld Measurement Tools method with 100% sampling plan, then the Production Verification Context is by a CMM - Coordinate Measurement Machine method with a User-Defined sampling plan of "Each lot, 60%", and then a Maintenance Verification Context is by the HAND method at 100% sampling.

/**D**: HAND: 100%//**P**: CMM: USR: "Each lot, 60%"//**M**: HAND: 100%/

8.4.7 VerificationAlternative

A VerificationPlanRequirementDesignator may designate a preferred verification method and optionally designate an alternative verification method. Note: when designating the preferred or only VerificationMethod, no additional designation is needed, and it should be left "blank". The format of the Verification Plan Requirement with the required Verification Method and a VerificationAlternative is shown below.

Format: /opt{ALT:} Method: 100%/

8.4.7.1 *Preferred Method*

{blank} Preferred

8.4.7.2 *Alternative Method (ALT)*

ALT: Alternate

8.4.7.3 *Verification Alternative Example*

An example of a Verification Plan Requirement with both a preferred and an alternate method is shown below. The preferred approach uses a CMM - Coordinate Measurement Machine with 100% sampling plan and the alternative uses HAND - Handheld Measurement Tools approach with 100% sampling plan.

Example: /CMM: 100%//**ALT**: HAND: 100%/

8.4.8 SamplingPlan

The Verification Plan Requirement has an optional sampling plan that governs the application of the method. The format of the Verification Plan Requirement with the required Verification Method and a Sampling Plan is shown below.

Format: /*Method*:***SamplingPlan/***

The optional *SamplingPlan* shall be one of the following choices:

- **100%:** 100% Sampling Plan

- **AQL**: Acceptable Quality Limit
- **LAS**: Lot Acceptance Sampling Plan
- **RSP**: Referenced Sampling Plan
- **SCL**: Statistical Quality Control Limit
- **USR**: User Defined Sampling Plan

Examples of Verification Plan Requirements with different Sampling Plans are shown below. The example shows a 100% sampling plan using the CMM VerificationMethod, a Lot Acceptance Sampling of a 45 sample size with one allowed defective sample using the CMM VerificationMethod , and then a Statistical Quality Control Limit of "Cpk >= 1.33" using the CMM VerificationMethod.

Example:

/ CMM: 100%/

/ CMM: LAS: 45,1 /

/ CMM: SCL: "Cpk >= 1.33" /

8.4.8.1 *100% Inspection/Test (100%)*

The Sampling Plan is to be performed at 100% inspection or test of the available product on the product characteristic. The format of the sampling plan within a Verification Plan Requirement is shown below

Format: /*Method*: 100%/

Example of Verification Plan Requirement with 100% Sampling Plan using a handheld measurement device is shown below.

Example: /HAND: **100%**/

8.4.8.2 *Acceptable Quality Limit (AQL)*

The Sampling Plan has a lot sampling plan based upon the acceptable quality limit (AQL) as defined per ANSI/ASQ Z1.4. Per ANSI/ASQ Z1.4, the **acceptable quality limit** (**AQL**) is the worst tolerable process *average* (*mean*) in percentage or ratio that is still considered acceptable; that is, it is at an **acceptable quality level**. The AQL approach to product quality inspection addresses: How many units should be inspected/tested within a given shipment and how many defective products are allowable before the inspection fails?

The AQL is based upon the Defect Category, Acceptable Inspection Level, and the AQL Limits that are appropriate for a product's customer(s). The format of the Verification Plan Requirement with the required Verification Method and an Acceptable Quality Limit is shown below.

Format: /*Method*: AQL: "*acceptable quality limit(s)*"/

Where:

Acceptable quality limit is:

Format: "*sampling level*"," *defect category*":"*AQL limit*"

Sampling level is multiple alphanumeric per section 8.4.8.2.1.

Defect Category is an alphanumeric per section 8.4.8.2.2.

AQL Limit is a numeric per section 8.4.8.2.3.

8.4.8.2.1 Sampling Levels

Acceptable Inspection levels are General Sampling Levels of GI (Reduced), GII (Normal), GIII (Tightened), or Special Sampling Levels of S1, S2, S3, S4. Different sample levels will command different numbers of samples to inspect. Per ANSI/ASQ Z1.4, the accepted Sampling Levels are:

- GI = General Reduced Sampling Level
- GII = General Normal Sampling Level,
- GIII = General Tightened Sampling Level,
- S1 = Special Sampling Level 1,
- S2 = Special Sampling Level 2,
- S3 = Special Sampling Level 3,
- S4 = Special Sampling Level 4

8.4.8.2.2 Defect Categories

Per ANSI/ASQ Z1.4, the Defect Categories are:

- CD = Critical Defects - Accepted defects could harm customers.
- MD = Major Defects - Defects that are not acceptable to the end users since defects are likely to make the product fail.
- mD = Minor Defects - Defects that do not reduce the product's functionality or purpose; end users will still buy these products.

8.4.8.2.3 AQL Limits

Per ANSI/ASQ Z1.4, the AQL Limits are:

- 0.0,
- 0.065,
- 0.10,
- 0.15,
- 0.25,
- 0.40,
- 0.65,
- 1.0,
- 1.5,
- 2.5,
- 4.0,
- 6.5

8.4.8.2.4 Acceptable Quality Limit Example

The format and an example of a Verification Plan Requirement with an AQL of a Normal General sampling plan with three defect categories and AQL Limits are shown below:

Example: /CMM: **AQL**: GII, CD: 0.0; GII, MD: 2.5; GII, mD: 4.0/

8.4.8.3 *Lot Acceptance Sampling (LAS)*

A Sampling Plan may apply a lot acceptance sampling plan, which is a sampling scheme and a set of rules for making decision. The decision is based upon counting the number of defectives in a sample, can be to accept the lot or reject the lot. A common LAS plan is the single LAS plan, which involves selecting at random from a lot and the disposition of the lot is determined from the resulting information. The single LAS plan is specified by the pair of numbers (*n, c*). The sample size is *n*, and the lot is rejected if there are more than *c* defectives in the sample; otherwise, the lot is accepted. The format of the Verification Plan Requirement with the required Verification Method and a Lot Acceptance Sampling is shown below.

Format: /*Method*: **LAS**: *n*, *c* /

Where:

n is the sample size number

c is the allowed defectives in the sample

Example of Verification Plan Requirement with a Lot Acceptance Sample Plan using a random lot sample of 15 and allowed defects of zero, using a HAND verification method is shown below. If there are more than zero defects, then the entire lot is rejected, else the entire lot is accepted

Examples: /HAND: **LAS**: 15, 0"/

8.4.8.4 *Referenced Sampling Plan (RSP)*

A Sampling Plan may apply a referenced sampling plan on the Product Characteristic followed by a specified reference or citation of a sanctioned document that may communicate how many artifacts out of the lot will be "sampled". The format of the Verification Plan Requirement with the required Verification Method and a Referenced Sampling Plan is shown below.

Format: /*Method*: **RSP**:"*string*"/

Example of Verification Plan Requirement with a Referenced Sampling Plan that references Table B1 within a reference identified as T518, using a CMM verification method, is shown below.

Example: /CMM: **RSP**:"T518: Table B1"/

8.4.8.5 *Statistical Quality Control Limits (SCL)*

A Sampling Plan may apply Statistical Quality Control Limits, which shall be added to any Verification Method. Control limits are used to detect signals from in-process data that indicate that a process is not in control and therefore not operating predictably. The format of the optional control limit is via a "control limit equation". The format of the Verification Plan Requirement with the required Verification Method and a Statistical Quality Control Limit is shown below.

Format: /*Method*: **SCL**: "*control limit equation*"/

Where:

Control limit equation is:

Format: "*control limit*" "*inequality expression symbol*" "*limit number*"

Inequality expression symbol shall be:

- > "greater than",
- < "less than",
- >= "greater than or equal to", or
- <= "less than or equal to"

Limit Number shall be any non-negative real number.

Control limit shall be one of the following:

- *ACL*
- *AOQ*
- *AOQL*
- *Cp*
- *Cpk*
- *CpkL*
- *CpkU*
- *LCL*
- *LCU*
- *LQL*
- *Pm*
- *Pmk*
- *PmkL*
- *PmkU*
- *Pp*
- *Ppk*
- *PpkL*
- *PpkU*
- *QK*
- *UCL*
- *UDL*

8.4.8.5.1 Acceptance Control Limit (ACL)

ACL = Acceptance Control Limit

8.4.8.5.2 Average Outgoing Quality (AOQ)

AOQ = Average Outgoing Quality

8.4.8.5.3 Average Outgoing Quality Limit (AOQL)

AOQL = Average Outgoing Quality Limit

8.4.8.5.4 Process Capability Index (Cp)

Cp = Process Capability Index

8.4.8.5.5 Process Capability Index, Minimum (Cpk)

Cpk = Process Capability Index, Minimum

8.4.8.5.6 Process Capability Index, Lower (CpkL)

CpkL = Process Capability Index, Lower

8.4.8.5.7 Process Capability Index, Upper (CpkU)

CpkU = Process Capability Index, Upper

8.4.8.5.8 Control Limit, Lower (LCL)

LCL = Control Limit, Lower

8.4.8.5.9 Control Limit, Upper (LCU)

LCU = Control Limit, Upper

8.4.8.5.10 Limiting Quality Level (LQL)

LQL = Limiting Quality Level

8.4.8.5.11 Machine performance Index (Pm)

Pm = Machine Performance Index

8.4.8.5.12 Machine performance Index, Minimum (Pmk)

Pmk = Machine Performance Index, Minimum

8.4.8.5.13 Machine performance Index, Lower (PmkL)

PmkL = Machine Performance Index, Lower

8.4.8.5.14 Machine performance Index, Upper (PmkU)

PmkU = Machine Performance Index, Upper

8.4.8.5.15 Process Performance Index (Pp)

Pp = Process Performance Index

8.4.8.5.16 Minimum Process Performance Index, Minimum (Ppk)

Ppk = Process Performance Index, Minimum

8.4.8.5.17 Process Performance Index, Lower (PpkL)

PpkL = Process Performance Index, Lower

8.4.8.5.18 Process Performance Index, Upper (PpkU)

PpkU = Process Performance Index, Upper

8.4.8.5.19 Process Variation Index (Qk)

Qk = Process Variation Index

8.4.8.5.20 Upper Control Limit (UCL)

UCL = Upper Control Limit

8.4.8.5.21 User Defined Limit (UDL)

UDL = User Defined Limit

Format: UDL:” string”

8.4.8.5.22 Statistical Quality Control Limit Example

Examples of Verification Plan Requirements with various Statistical Control Limits, using a CMM verification method, is shown below.

/CMM: **SCL:** "Cpk >= 1.33"/

/CMM: **SCL:** "LCL > .100"/

/CMM: **SCL:** "UDL:"HON18" <= 10"/

8.4.8.6 *User Defined Sampling Plan (USR)*

A Sampling Plan may apply a user defined sampling plan other than a standard sampling plan on the product characteristic. A user defined sampling plan cannot be specified by reference and shall be fully specified. The format of the Verification Plan Requirement with the required Verification Method and a User-Defined Sampling Plan is shown below.

Format: /*Method*: **USR**:"*string*" /

Example of Verification Plan Requirement with User-Defined Sampling Plan, using a CMM verification method, is shown below.

Examples: /CMM: **USR**: "Each lot, 60%"/

9 MBC Structure Information Model

The entire information model for the contents of this standard is shown below:

9.1 Product Characteristics with Augmentations Information Model

This Model-Based Characteristics block diagram is described in section 1.5.

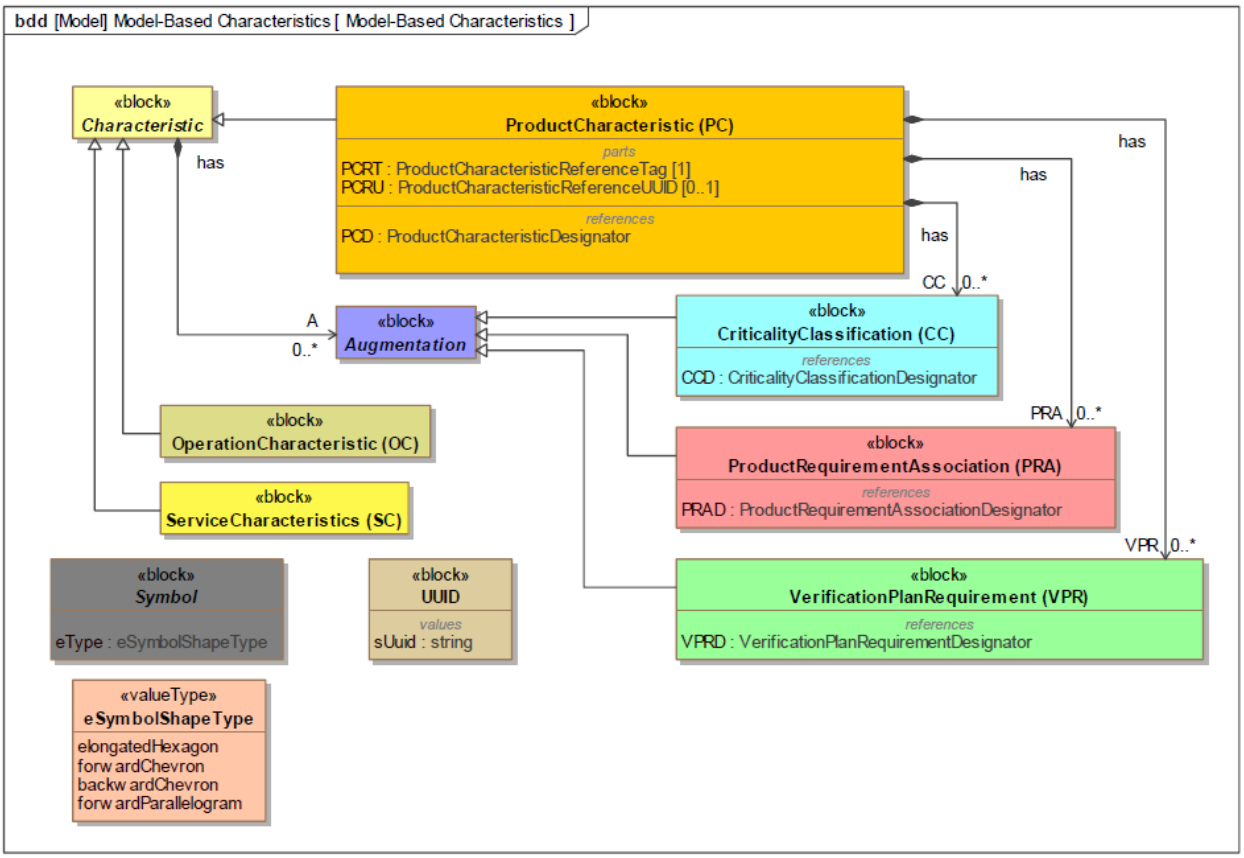


Figure 81: Block Diagram - Product Characteristic with Augmentations Information Model

9.2 Product Characteristics Identification Framework Data Model

This Product Characteristics Identifiers block diagram is described in section 5.1.1.

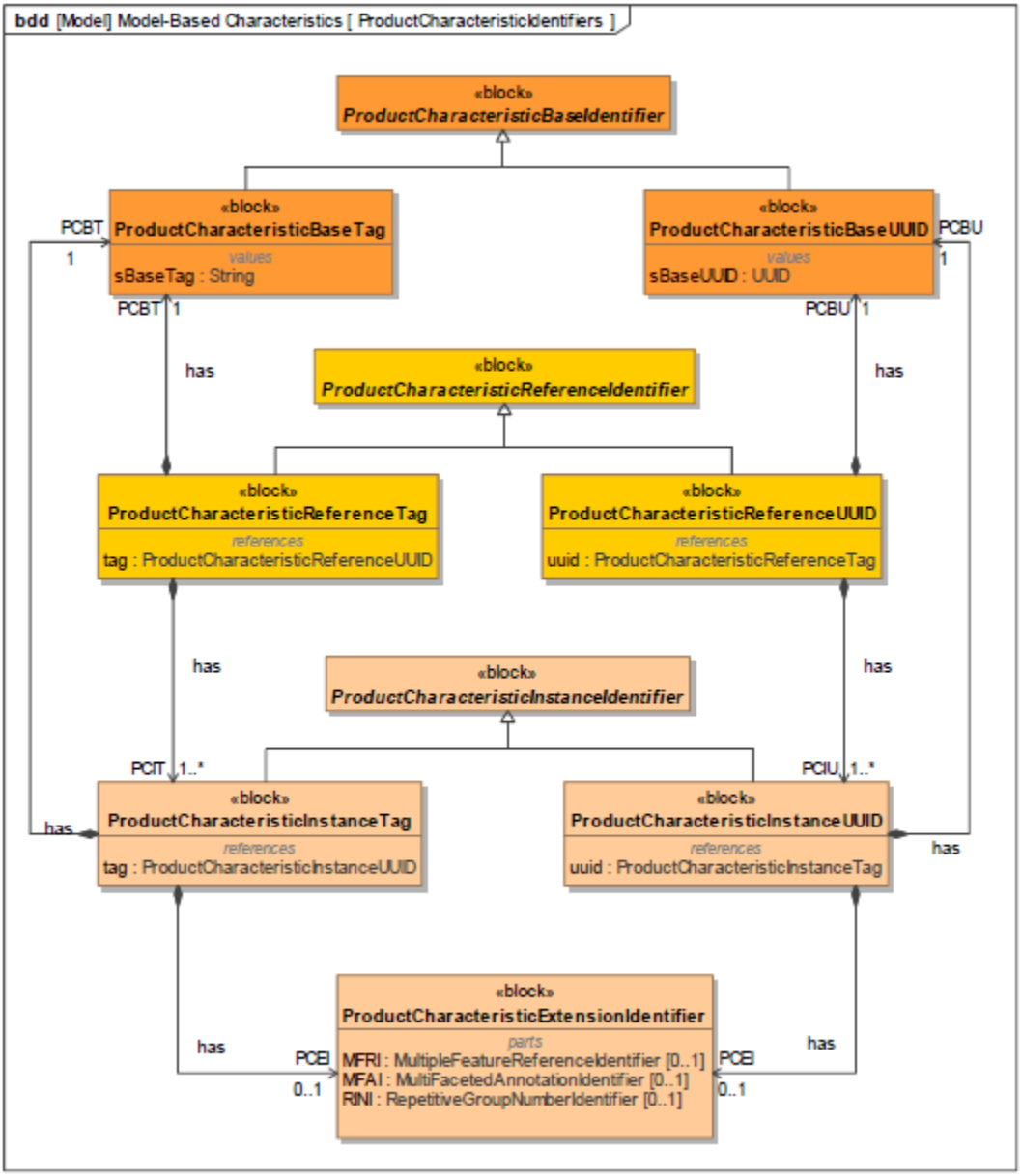


Figure 82: Block Diagram - Product Characteristic Identification Framework

9.3 Product Characteristics Extension Identifier Data Model

This Product Characteristics extended identifier block diagram is described in section **Error! Reference source not found.**.

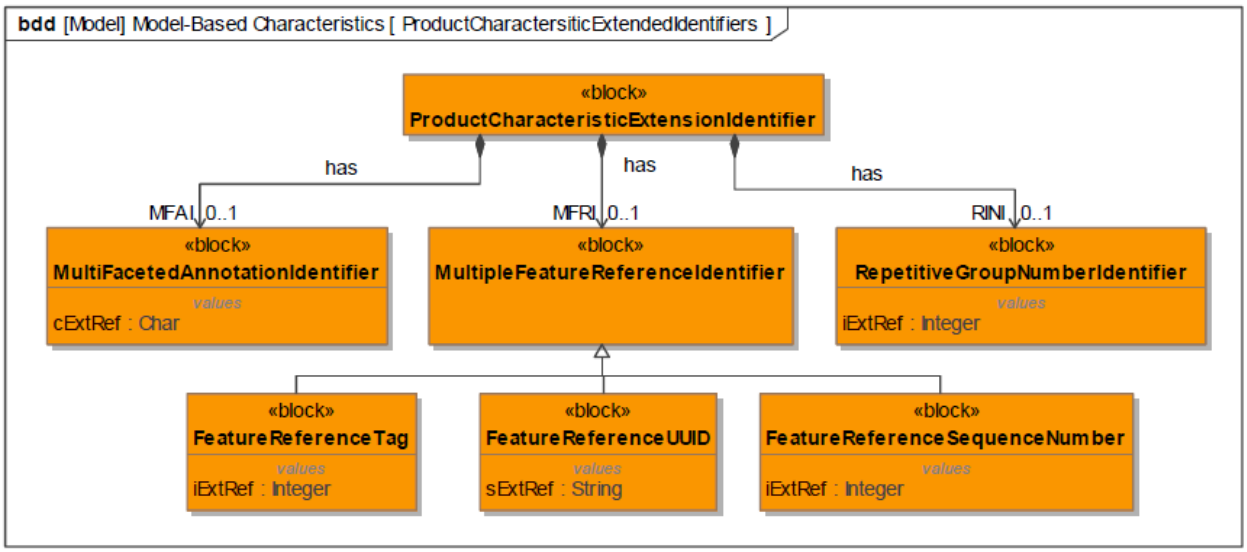


Figure 83: Block Diagram - Product Characteristic Extension Identifiers

9.4 Product Characteristic Data Model

This Product Characteristics block diagram is described in section 5.2.

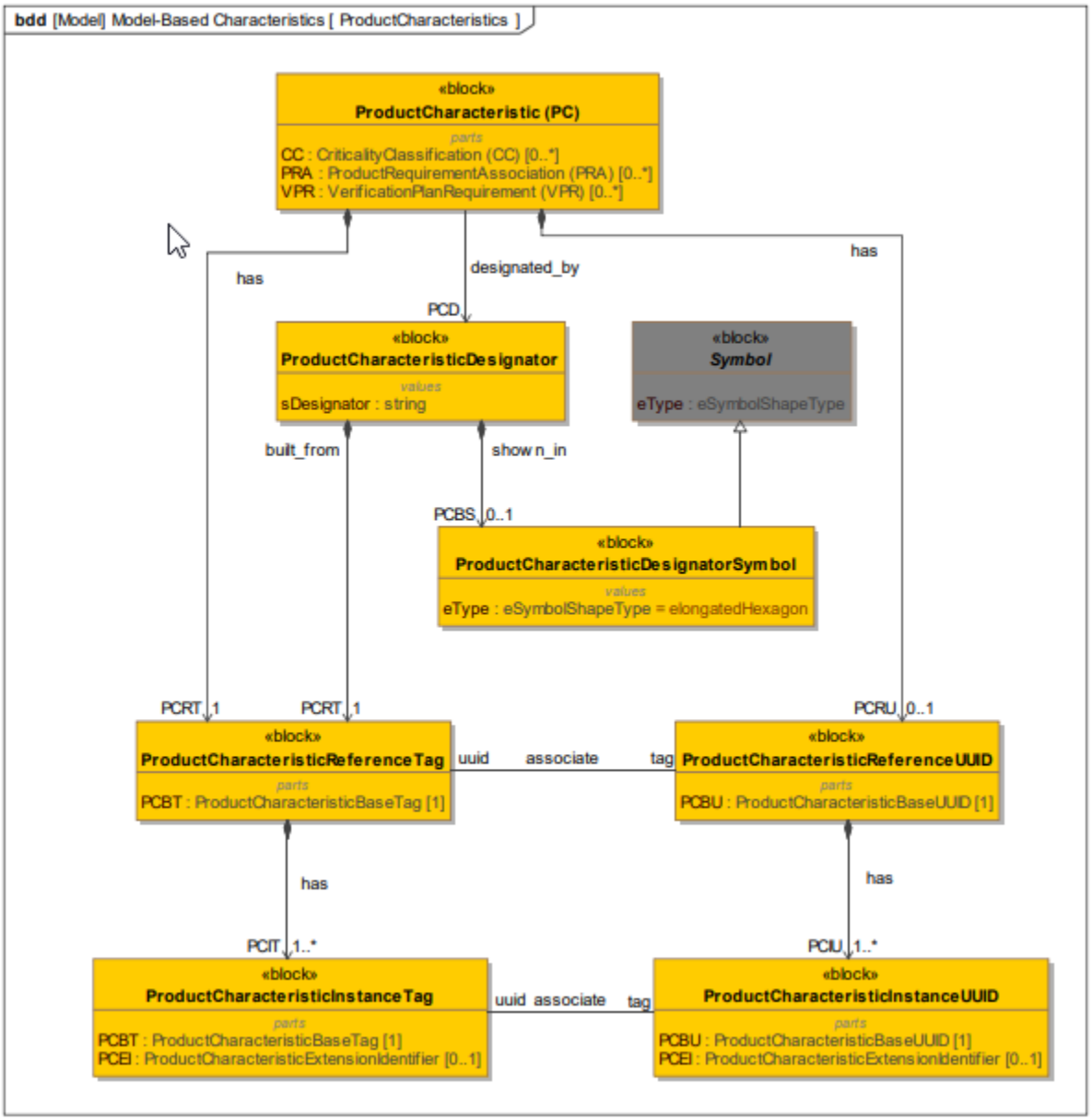


Figure 84: Block Diagram - Product Characteristic Information Model

9.5 Criticality Classification Augmentation Data Model

This Critically Classification block diagram is described in section 6.2

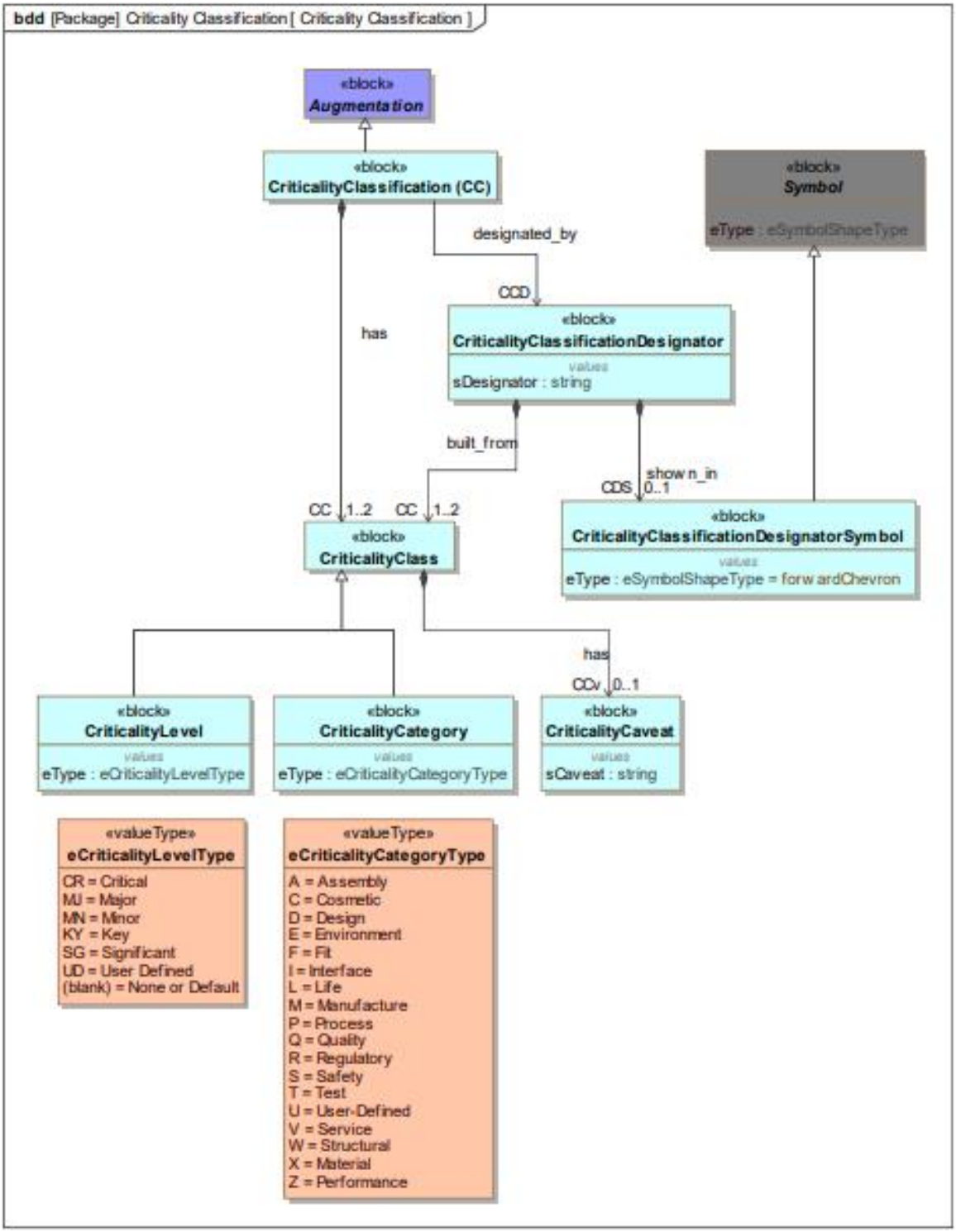


Figure 85: Block Diagram - Criticality Classifications Information Model

9.6 Product Requirement Association Augmentation Data Model

This Product Requirement Association block diagram is described in section 7.2

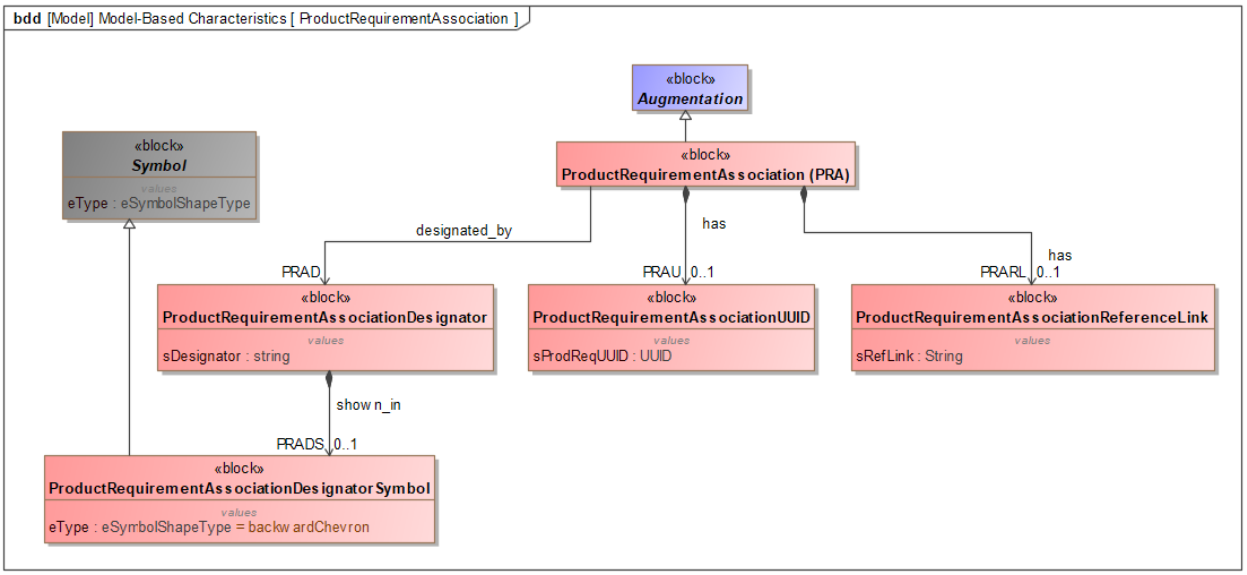


Figure 86: Block Diagram - Product Requirement Associations Information Model

9.7 Verification Plan Requirement Augmentation Data Model

This Verification Plan Requirement block diagram is described in section 8.2.

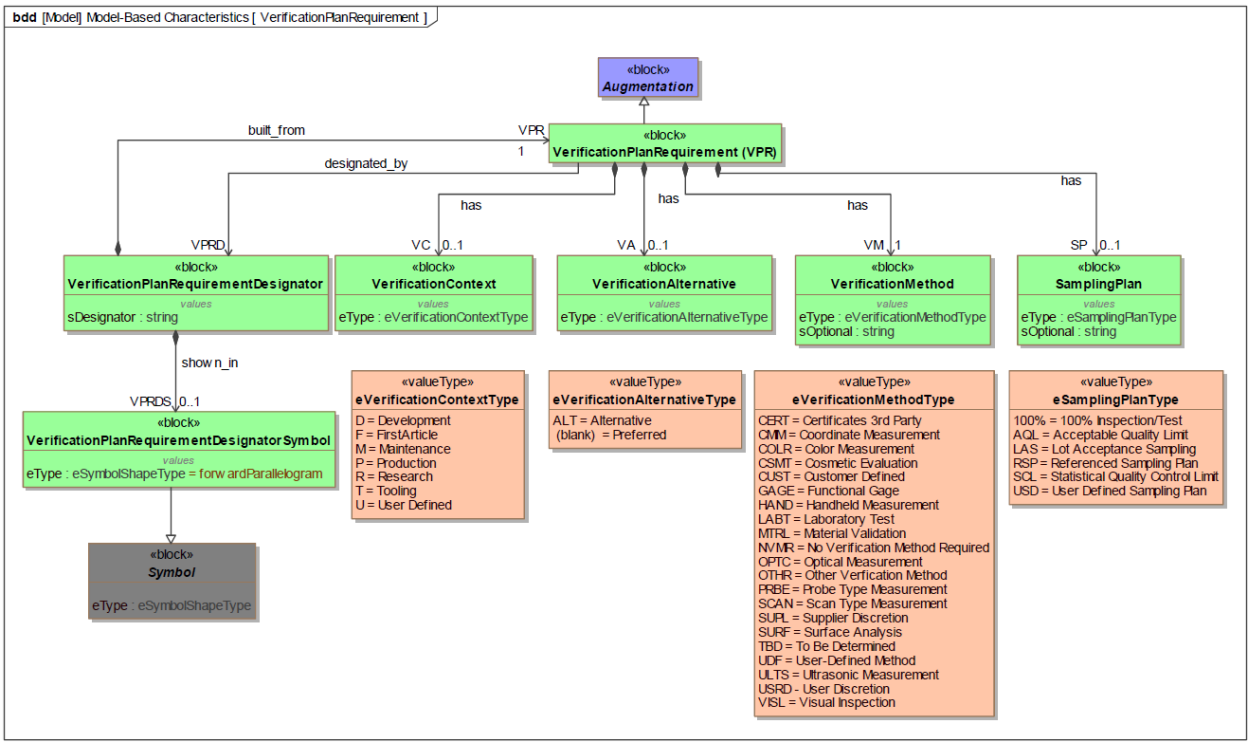


Figure 87: Block Diagram - Verification Plan Requirement Information Model

10 Informative Appendix

The following annexes are provide for information only to help implementation and usage of this standard as well as to obtain comments for future work.

10.1 Annex A – Form & Proportion of Symbols

(Informative)

The following illustrate the preferred form and proportion for the Product Characteristic symbols and Augmentation Symbols. See ASME Y14.2 for line weights, letter heights, and arrow proportions.

10.1.1 Preferred Form and Proportion

Product Characteristic and Augmentation symbol proportions are given in a factor of *h*, where *h* is the letter height selected for use within the enclosed symbols.

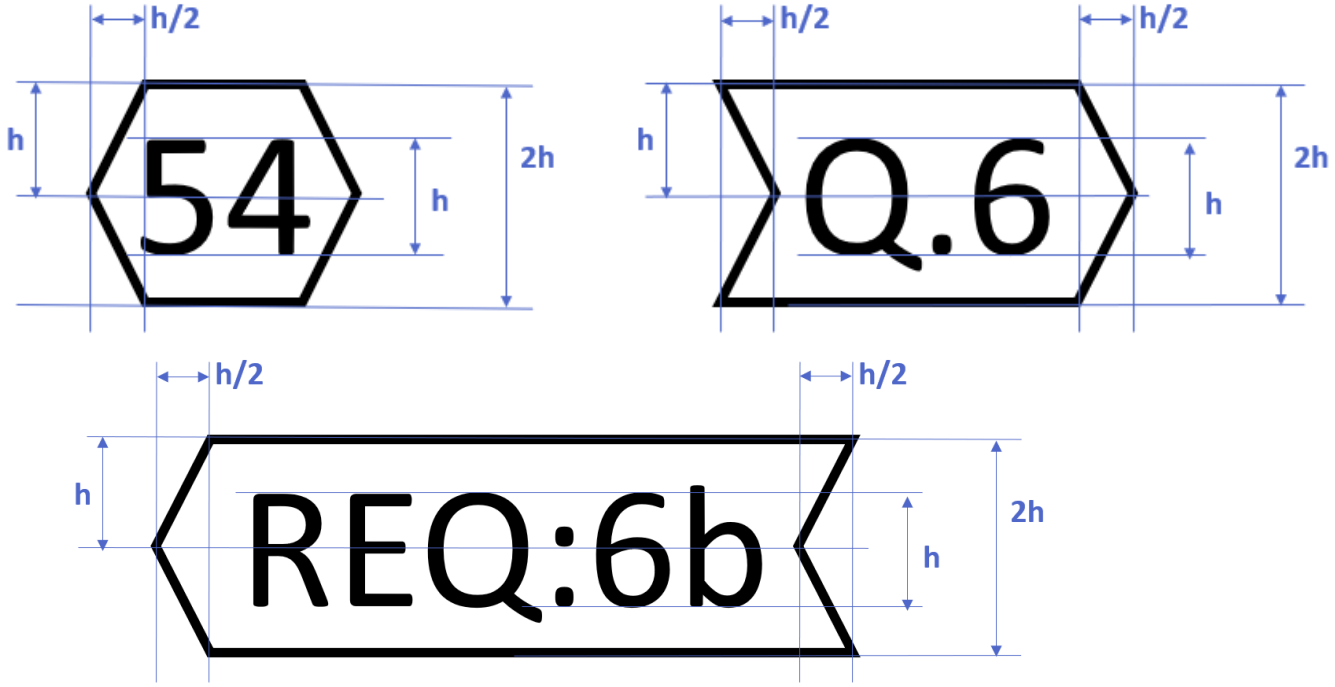


Figure A- 1 Preferred Form and Proportion for Symbols

10.1.2 Compound Symbol Spacing

For compound symbols, there should be adequate spacing between symbols for legibility.

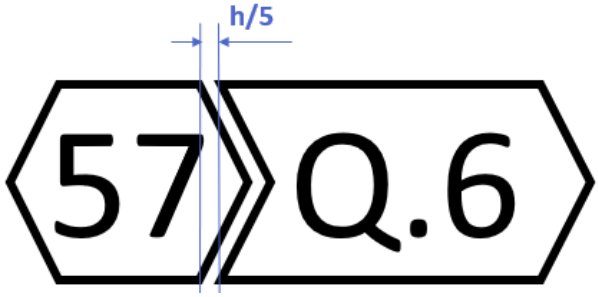


Figure A- 2 Compound Symbols with Adequate Spacing

10.1.3 Symbol Alignment

Augmentation symbols shall be placed in alignment with the Product Characteristic Symbol.

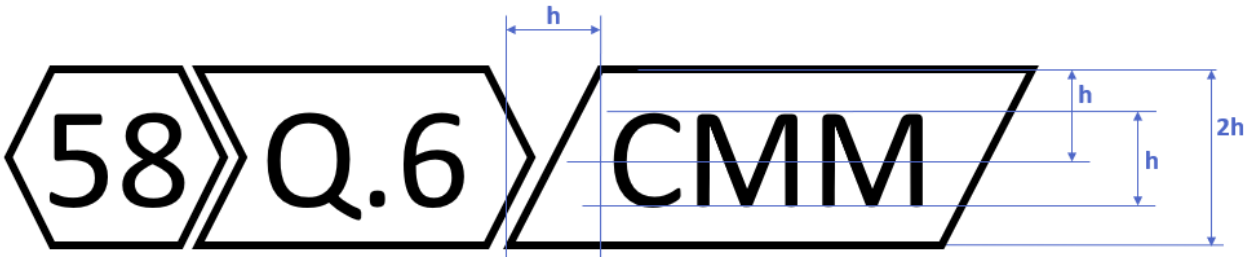


Figure A- 3 Augmentation Symbols Alignment

Additional symbols for data reporting are presented in an isosceles trapezoid form with a longer bottom base.

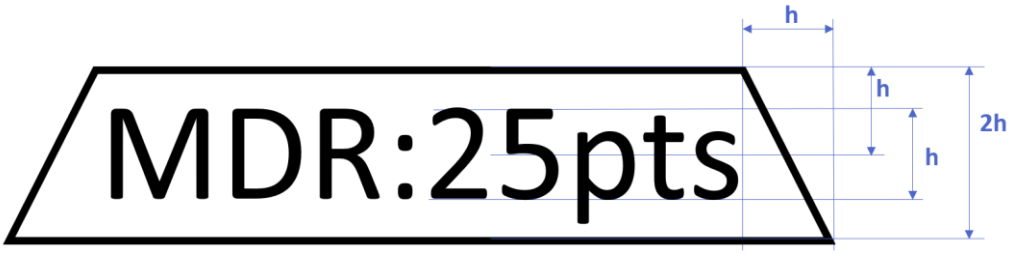


Figure A- 4 Augmentation Symbols for Data Reporting

10.2 Annex B – QIF Characteristic Designator Type

(Informative)

The QIF v3.0 schema browser may be found at the following link:
https://qualityinformationframework.github.io/qif3-browser/qif3.html

10.2.1 Characteristics

Within QIF, the Characteristic object contains three lists of characteristics aspects, definition, nominal, and item. Each of those, CharacteristicDefinitionBaseType, CharacteristicNominalBaseType, and CharacteristicItemBaseType, defines a CharacteristicDesignator as an optional element. CharacteristicMeasurementBaseType is defined elsewhere in the QIF model.

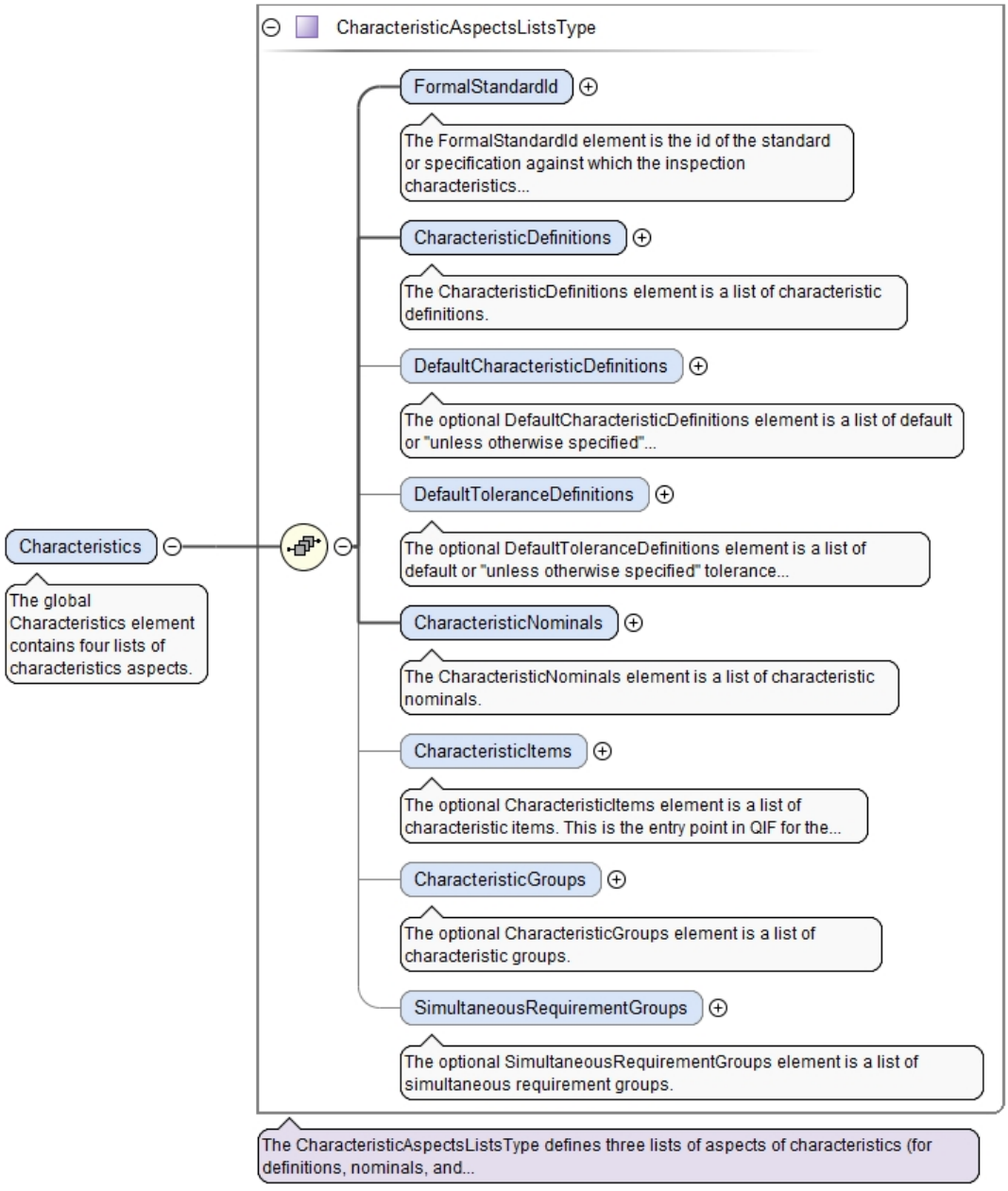


Figure B- 1 QIF Characteristic XSD Schema

10.2.2 CharacteristicDefinitionBaseType / Characteristic Designator

The CharacteristicDefinitionBaseType is the abstract base type within QIF that defines information that can be common to more than one characteristic. The CharacteristicDefinitionBaseType has an optional CharacteristicDesignator element.

10.2.3 CharacteristicNominalBaseType / Characteristic Designator

The CharacteristicNominalBaseType is the abstract base type that defines a unique characteristic nominal. The CharacteristicNominalBaseType has an optional CharacteristicDesignator element.

10.2.4 CharacteristicItemBaseType / Characteristic Designator

The CharacteristicItemBaseType is the abstract base type within QIF that defines information that can be common to all characteristic items. The CharacteristicDefinitionBaseType has an optional CharacteristicDesignator element.

10.2.5 Characteristic Designator

The Characteristic Designator is the optional designator of a characteristic together with an optional level of criticality and optional UUID. The designator element of the CharacteristicDesignator will typically be at the most general level when used in this CharacteristicXxxxxxxxBaseType. For example, an instance of CharacteristicDefinitionBaseType might have a CharacteristicDesignator element with the Designator K. Then there might be three instances of CharacteristicItemBaseType referencing the CharacteristicDefinitionBaseType instance and having CharacteristicDesignator elements using Designators K-1, K-2, and K-3.

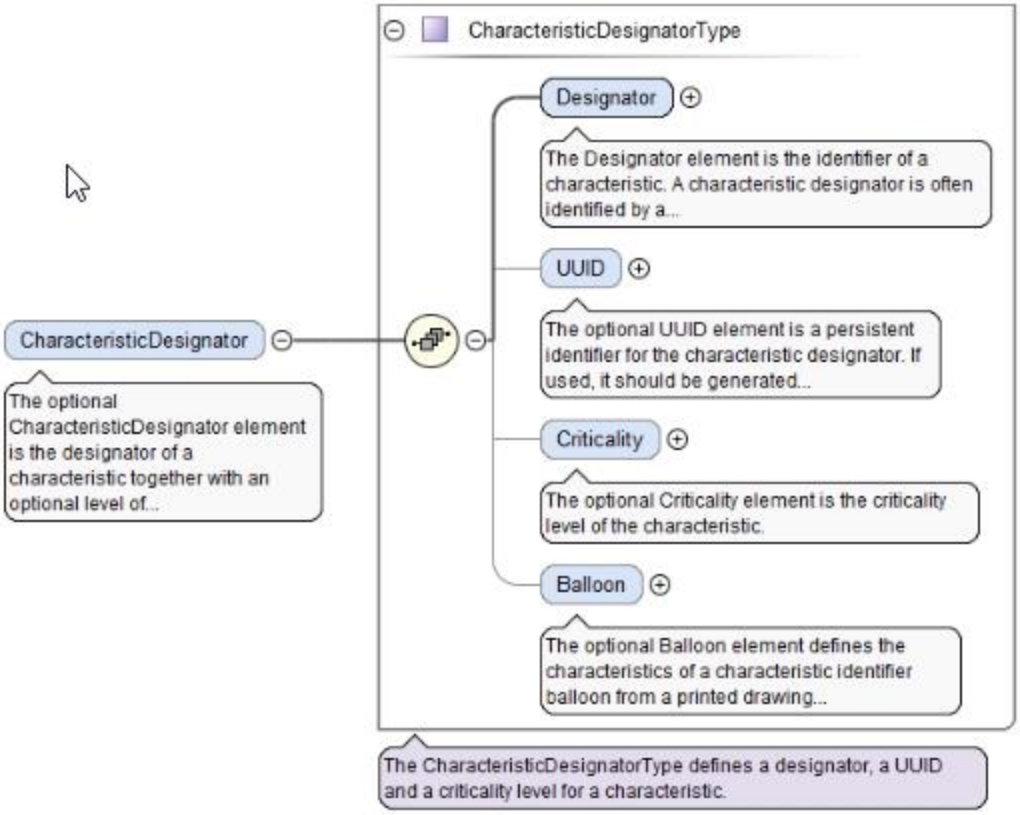


Figure B- 2 QIF CharacteristicDesignator XSD Schema

10.2.6 Characteristic Designator Type

The figures above and below may be found at the link:
https://qualityinformationframework.github.io/qif3-browser/Characteristics_xsd_Complex_Type_CharacteristicDesignatorType.html#CharacteristicDesignatorType_Designator

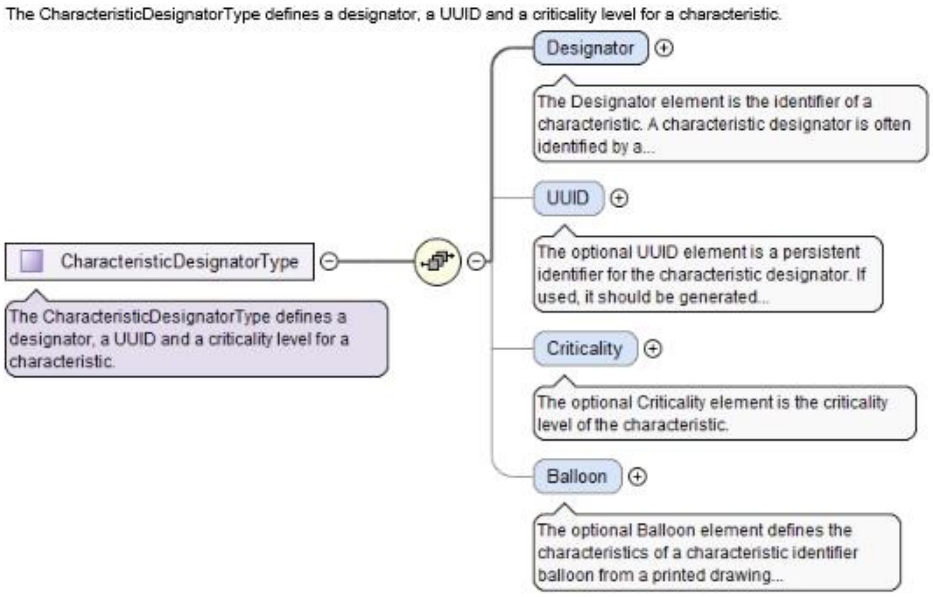


Figure B- 3 QIF CharacteristicDesignatorType XSD Schema

10.2.6.1 *Characteristic Designator Type / Designator*

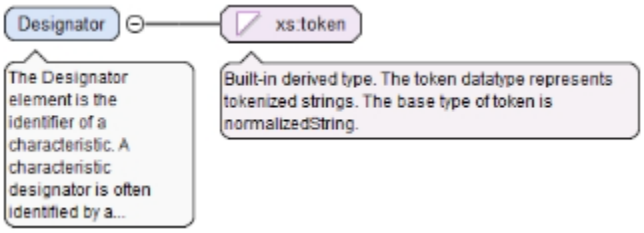


Figure B- 4 QIF CharacteristicDesignatorType / Designator XSD Schema

10.2.6.2 *Characteristic Designator Type / UUID*

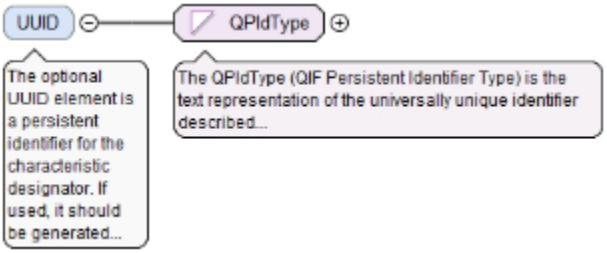


Figure B- 5 QIF CharacteristicDesignatorType / UUID XSD Schema

10.2.6.3 *Characteristic Designator Type / Criticality*

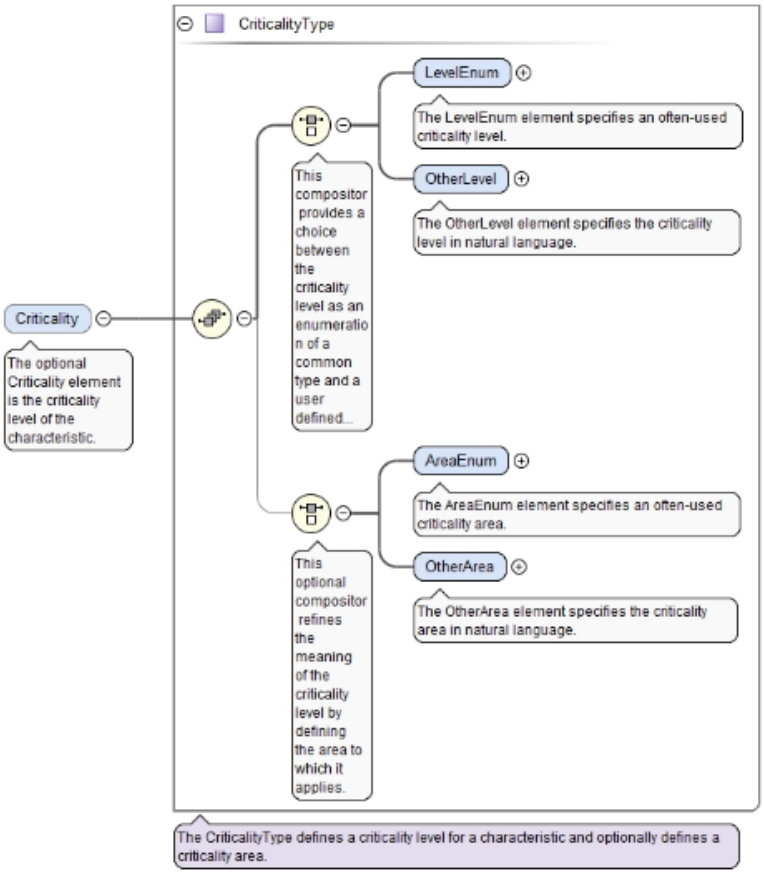


Figure B- 6 QIF CharacteristicDesignatorType / Criticality XSD Schema

10.2.6.4 *Criticality Type / Level Enum*

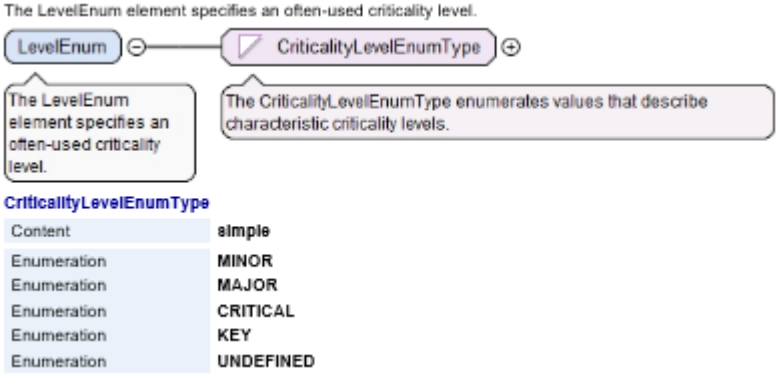


Figure B- 7 QIF CharacteristicDesignatorType / Level Enum XSD Schema

10.2.6.5 *Criticality Type / Other Level*

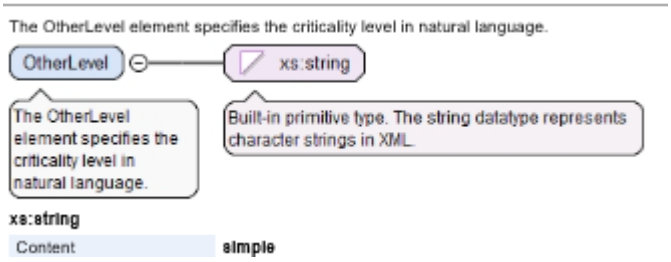


Figure B- 8 QIF CharacteristicDesignatorType / Other Level XSD Schema

10.2.6.6 *Criticality Type / Area Enum*

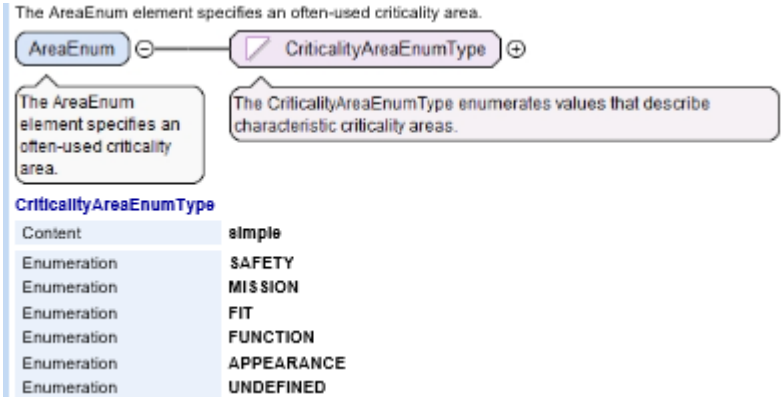


Figure B- 9 QIF CharacteristicDesignatorType / Area Enum XSD Schema

10.2.6.7 *Criticality Type / Other Area*

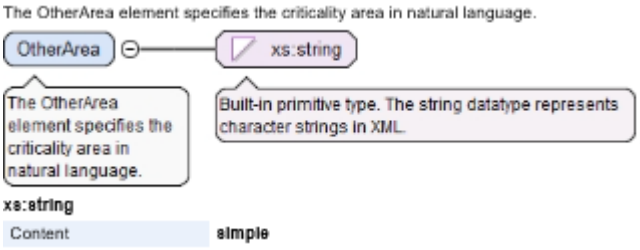


Figure B- 10 QIF CharacteristicDesignatorType / Other Area XSD Schema

10.2.7 Characteristic Designator Type / Balloon

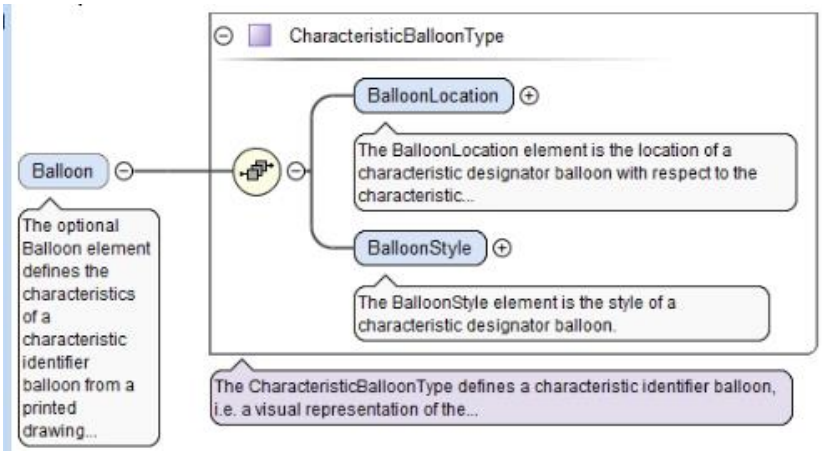


Figure B- 11 QIF CharacteristicDesignatorType / Balloon XSD Schema

10.3 Annex C – Selection Criteria for a Characteristic Designator Symbol

(Informative)

The following criteria were defined and used towards the recommended selection for the Characteristic Designator Symbol. This review also included how the Criticality symbol and the Requirement Association symbols would visually be associated with the Characteristic Designator symbol.

C1. Symbol must be a recognizable unique shape.

C2. Symbol must be easily creatable using existing office/CAD tools.

C3. Symbol must be able to enclose a set of alphanumeric identifiers.

C4. Symbol must not conflict with other regularly used symbols in related ASME / ISO standards.

C5. Symbol can be easily associated with an annotation (e.g., Dimensional Tolerance, Geometric Tolerance, Surface Texture, General Note, and Flagged Note).

C6. Symbol must be able to accommodate Criticality symbol(s) before or after.

C7. Symbol can be chained with one or more Product Requirement symbol(s).

C8. Symbol must be easily created as a textual field.

C9. Symbol must be applicable for 2D drawings, 3D drawings, and 3D MBDs.

C10. Symbol can be both human-readable, and when implemented within a digital MBD, be machine-readable.

C11. The digital implementation of the symbol instance should be digitally associated with the respective annotation.

C12. The digital implementation of the symbol should digitally associate with a persistent universally unique identifier for the symbol instance.

10.4 Annex D – MBC Attributes Names for Applications

(Informative)

Some recommended attribute names for applications (e.g., MBD, CAI) to consider toward supporting this standard are the following:

10.4.1 MBC_PC (Product Characteristic)

Attribute Name	Attribute Type	Attribute Value
MBC_PC_TAG	String	Tag of Product Characteristic
MBC_PC_UUID	UUID or String	UUID of Product Characteristic
MBC_PC_ANNOTATION_ID	TBD	Application's Identifier of the Product Characteristic's related Annotation

10.4.2 MBC_CC (Critical Classification)

Attribute Name	Attribute Type	Attribute Value
MBC_CC_PC_TAG	String	Tag of CC's Product Characteristic
MBC_CC_PC _UUID	UUID or String	UUID of CC's Product Characteristic
MBC_CC_DESIGNATION	String	String content within the CC Symbol
MBC_CC_LEVEL	String or Enum	Criticality Level Enum or string
MBC_CC_CATEGORY	String or Enum	Criticality Category Enum or string
MBC_CC_CAVEAT	String	Criticality Caveat string

10.4.3 MBC_PRA (Product Requirement Association)

Attribute Name	Attribute Type	Attribute Value
MBC_PRA_PC_TAG	String	Tag of PRA's Product Characteristic
MBC_PRA_PC _UUID	UUID or String	UUID of PRA's Product Characteristic
MBC_PRA_DESIGNATION	String	String content within the PRA Symbol
MBC_PRA_LINK	URL or String	URL or String of PRA
MBC_PRA_UUID	UUID or String	UUID of PRA

10.4.4 MBC_VPR (Verification Plan Requirement)

Attribute Name	Attribute Type	Attribute Value
MBC_VPR_PC_TAG	String	Tag of VPR's Product Characteristic
MBC_VPR_PC _UUID	UUID or String	UUID of VPR's Product Characteristic
MBC_VPR_DESIGNATION	String	String content within the VPR Symbol
MBC_VPR_CONTEXT	String or Enum	Verification Context's Enum or string
MBC_VPR_ALT	Boolean	Verification Alternative Boolean (default is false)
MBC_VPR_METHOD	String or Enum	Verification Method's Enum or string
MBC_VPR_SAMPLEPLAN	String or Enum	Sampling Plan's Enum or string

10.4.5 MBC_GT (General Tag)

Attribute Name	Attribute Type	Attribute Value
MBC_GT_TAG	String	Tag of Reference Tag
MBC_GT_UUID	UUID or String	UUID of Reference Tag
MBC_GT_ANNOTATION_ID	TBD	Application's Identifier of the Reference Tag's related Annotation

10.5 Annex E – Acronyms & Abbreviations

(Informative)

10.5.1 Characteristic Tags

Characteristic Tags context reference section 4.1 and 5.19

- **PC:** Product Characteristic
- **OC:** Operation Characteristic
- **SC:** Service Characteristic
- **MBC:** Model-Based Characteristic
- **DBC:** Drawing-Based Characteristic
- **GT:** General Tag

10.5.2 PC Augmentations

Product Characteristic Augmentation context reference section 4.3

- **CC**: Criticality Classifications
- **PRA**: Product Requirement Associations
- **VPR**: Verification Plan Requirements

10.5.3 Criticality Levels

Criticality Level context reference section 6.4.5

- **CR**: Critical
- **MJ**: Major
- **MN**: Minor
- **KY**: Key
- **SG**: Significant
- **UD**: User-Defined
- (Blank) None or default per organizational practices

10.5.4 Criticality Category

Criticality Category context reference section 6.4.6

- **A**: Assembly
- **C**: Cosmetic
- **D**: Design
- **E**: Environmental
- **F**: Fit
- **I**: Interface
- **L**: Life
- **M**: Manufacture
- **P**: Process
- **Q**: Quality
- **R**: Regulatory
- **S**: Safety
- **T**: Test
- **U**: User-Defined
- **V**: Service
- **W**: Structural
- **X**: Material

Z: Performance

10.5.5 Verification Method

Verification Method context reference section 8.4.5 also Annex G

CERT Certificates: 3rd party documents
CMM Coordinate Measurement Machine
COLR Color Measurement & Appearance Equipment
CSMT Cosmetic Evaluation
CUST Customer Defined
GAGE Functional Gage
HAND Handheld Measurement Tools
LABT Laboratory Test
MTRL Material Validation
NVMR No Verification Method Required
OPTC Optical Measurement Equipment
OTHR Other Verification Method
PRBE Probe Type Measurement Equipment
SCAN Scan Type Measurement Equipment
SUPL Supplier Discretion
SURF Surface Analysis Equipment
TBD To Be Determined
ULTS Ultrasonic Measurement Equipment
UDF User Defined Verification Method
USRD User Discretion
VISL Visual Inspection

10.5.6 Verification Context

Verification Context reference section 8.4.6

D Development
F First Article
M Maintenance
P Production
R Research
T Tooling
U User Defined

10.5.7 Verification Alternative

Verification Alternative reference section 8.4.7

ALT Alternative
{blank} Preferred

10.5.8 Sampling Plan

Sampling Plan reference section 8.4.8

100% First Article
AQL Acceptable Quality Limit
LAS Lot Acceptance Sampling Plan
RSP Reference Sampling Plan

SCL Statistical Quality Control Limit
USR User Defined Sampling Plan

10.5.9 Statistical Control Limit Variables

Statistical Control Limit Variables reference section 8.4.8.5

ACL Acceptance Control Limit
AOQ Average Outgoing Quality
AOQL Average Outgoing Quality Limit
Cp Process Capability Index
Cpk Process Capability Index, minimum
CpkL Process Capability Index, lower
CpkU Process Capability Index, upper
LCL Control Limit, Lower
LCU Control Limit, Upper
LQL Limiting Quality Level
Pm Machine performance index
Pmk Machine performance index, minimum
PmkL Machine performance index, lower
PmkU Machine performance index, upper
Pp Process Performance Index
Ppk Minimum Process Performance Index
PpkL Process Performance Index, lower
PpkU Process Performance Index, upper
QK Process Variation Index
UCL Upper Control Limit
UDL User Defined Limit

10.5.10 Product Characteristic Extension Identifiers

Product Characteristic Instance Extension Identifiers reference section 5.4.7

MFAI Multi-Faceted Annotations Identifiers
MFRI Multiple Feature Reference Identifiers
RGNI Repetitive Group Number Identifiers
FRT Feature Reference Tag
FRU Feature Reference UUID
FRSN Feature Reference Sequence Number

10.5.11 Additional Acronyms

Additional Acronyms context:

ANSI American National Standards Institute
ASCII American Standard Code for Information Interchange
ASME American Society of Mechanical Engineers
BoC Bill of Characteristics reference section 1.3.3
CMM Coordinate Measuring Machine
DMSC Digital Metrology Standards Consortium
DRF Datum Reference Frame
FEAT Feature

GD&T	Geometric Dimensioning and Tolerancing
ISO	International Organization for Standardization
MBC	Model-Based Characteristic
MBD	Model-Based Definition
MBE	Model-Based Enterprise
PC	Product Characteristic
PMI	Product and Manufacturing Information
QIF	Quality Information Framework
QPId	QIF Persistent Identifier
REQ	Requirement
UUID	Universally Unique Identifier

10.6 Annex F – System Modeling Language (SysML) Block Diagram Modeling

(Informative)

System Modeling Language (SysML) is a graphical modeling language used to design, analyze, and specify complex systems. It is an extension of Unified Modeling Language (UML) and is designed specifically for systems engineering applications. SysML provides a set of graphical notations and constructs for modeling different aspects of a system, including its structure, behavior, requirements, and constraints.

10.6.1 Block Diagrams

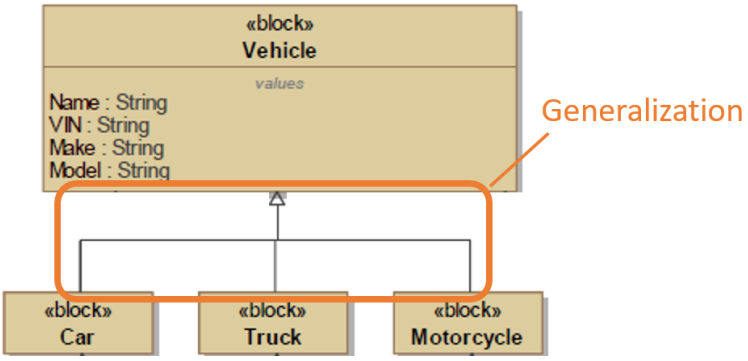
A SysML Block Definition Diagram (BDD) is a graphical representation that shows the logical and physical structure of a system. It is used to define and specify the system's components, their properties, and the relationships between them. The BDD is the primary diagram used in SysML for defining system components and their relationships. It provides an overview of the system being modeled and includes blocks that represent the system's components, and how the components relate with each other. Overall, a SysML BDD provides a high-level view of a system's structure and helps stakeholders understand how the system's components fit together to achieve its intended functionality.

10.6.2 Relating Blocks

Blocks are connected by using the main relationships as follows.

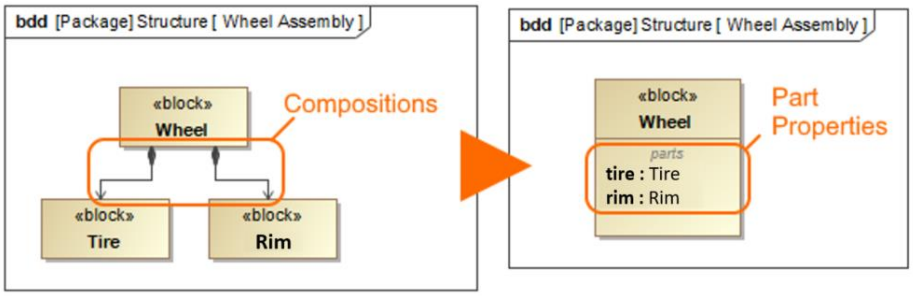
10.6.2.1 *Generalization or Inheritance*

In SysML (Systems Modeling Language), the Generalization relationship is used to represent an "is-a" relationship between two model elements, where one element is a more specialized version of the other. In other words, the Generalization relationship is used to model inheritance or specialization in the SysML model. The more specialized element inherits the attributes, operations, and relationships of the more general element and may also have additional attributes, operations, and relationships that are specific to it. This relationship is represented graphically using an arrow that points from the specialized element to the more general element, with an open arrowhead at the end of the arrow. The specialized element is typically shown as a child of the more general element in a diagram. The figure below shows that a *Cars*, *Trucks*, and *Motorcycles* are types of *Vehicles*. It means that all the subtypes (*Cars*, *Trucks*, and *Motorcycles*) require all the characteristics of *Vehicles* but add their own specialized characteristics as well.



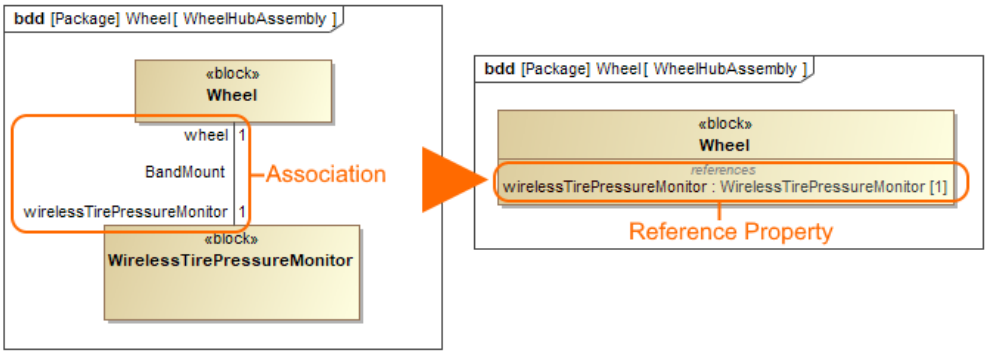
10.6.2.2 *Directed Composition or Composition*

The Direct Composition or Composition relationships convey a structural decomposition of Blocks. The notation is a solid line between two Blocks with a solid diamond on the composite end. The following figure shows that the *Wheel* is composed of *Tire* and *Brake*. The Composition can be represented in two different ways: by showing Composition relations between Blocks or by showing Part Properties on decomposed Block compartment.



10.6.2.3 *Direct Association or Association*

The Direct Association or Association convey that a connection can exist between those Blocks. Those Blocks can access each other for some purpose across the connection. The Direct Association notation is a solid line. The figure below illustrates the Association between *Wheel* and *WirelessTirePressureMonitor* Blocks. The name of the Association is *BandMount*, which describes the type of connection that could exist between the wheel and the wireless tire pressure monitor. An association can be represented in two different ways: by showing Association relation between Blocks or by showing a Reference Property on a decomposed Block compartment.



10.7 Annex G – Verification Method Types Detail

(Informative)

The following table is a taxonomy of verification method types, equipment, processes, software, and tools. This classification structure may be used to help specify the Verification Method types and data elements used within section 8.4.5 and 10.5.5. This listing is a classification system that may be used as a reference guide.

1		**Handheld Measurement Tools:** (manual)
2		**Linear Gauges**
	2A	Micrometers
	2B	Scan Micrometer
	2C	Calipers
	2D	Depth Gages
	2E	Indicators
	2F	Feeler Gages
	2G	Center Gages
	2H	Height Gages
	2J	Holes -Pin Gages -Bore Gages -Inside Micrometers
	2K	Master Spheres
	2L	Thickness Gage
	2M	Measure Tapes & Rules
	2N	Gage Blocks
3		**Angle Measurement**
4		**Chamfer Gages**
5		**Radius Gages**
6		**Thread Gages**
	6A	Go-NoGo
	6B	Feeler Gage Threads
7		**TBD**
8		**TBD**
9		**Specialty**
	9A	Gears
10		**Functional Gages:** Fixtures & Functional Gages:
11		**Profile gage**
12		**Go- NoGo gages**
13		**MMC & MMB Gages**
14		**RFS & RMB gages**
15		**Flexible Product Gages**
16		**Instrumented gage:** gage fixture with one or more indicators or features
17		**Air Gages**
18		**Specialty Gages**
20		**Probe Type Inspection/Measurement**
21		**CMM** Coordinate Measuring Machines: Bridge, Horizontal Arm, Cantilever,
	21A	**Gantry CMM** large size bed CMM (greater than 2m x 2m)

22 **Probe Arm** Portable Articulated CMMs
23 **Probe Comparator** Parallel Link CMM type

30 **Non-Contact Inspection/Measurement**
31 **Laser Scanner Handheld -** handheld blue light
32 **Laser Scan Machines 3D-** CNC type , robot manipulated, cabinet based
32A **Laser Scan Machines 2D Measurement** desktop, micrometer accuracy, flat or cylindrical parts,
33 **Laser Scan Arm** - portable measurement arm
34 **Laser Tracker**
35 **Theodolite** - tripod mount for very large objects (airplanes or buildings)
36 **X-Ray 2D Image** X-ray Scanning
37 **CT 3D Representation** X-Ray 3D Computed Tomography
38 **Ultrasonic Measurement**

40 **Optical Measurement Equip**:
41 **Visual Inspection**
42 **Optical Comparators**
43 **Optical Measurement 2D**
43A **Optical Measurement 2D+ additional Axis**
44 **Microscopes & Enlarged Digital Images**
45 **Cameras & Digital Images**
46 **Vision Systems**

50 **Surface Analysis Measurement** : Roughness, waviness, flatness,
51 **Thickness of Plating & Coatings -**Electroplating and Metals Thickness, Paint, Powder Coating, & Anodizing
52 **Color Measurement** Equipment: Pigmentation, color, fluorescence, & gloss
53 **Cosmetic Evaluation:** Cosmetic finishes evaluation: imperfections: scratches, blemishes, gouges

60 **Material Analysis**
61 **Metal Composition** - Metallurgy using X-Ray Fluorescence (XRF), Optical Emission Spectroscopy (OES), Laser Induced Breakdown Spectroscopy (LIBS).
62 **Polymer Material Characterization** - Plastic Material Analysis
63 **Composite Material Evaluation** - Prepreg and dry fabrics for design, thermoset system in prepreg, liquid resin & sheet form, sheet molding compounds (SMC)
66 **Materials Test**: Torque, fatigue, hardness, impact, adhesion, force, strain, tensile, compression,
66A **Tensile testing strain, force, compression**
66B **Hardness test**
66C **Impact**
63D **Fatigue**
66E **Porosity**
66F **Adhesion**
66G **Galvanic level**
66H **Outgassing**

70 **General Verification Options**

71 **Certificate of Verification:** 3rd party documents**:** material, heat treat, anodize

72 **Visual Inspection:** miscellaneous, surface lay, no paint area, plating

73 **Supplier Discretion**

74 **TBD:** To Be Determined

75 **Other Verification Method:** tool or compliance category not listed**,**

80 **Assembly & Process Verification**

81 **Torque:** Torque wrench

82 **Cure Cycle**

85 **Laser Profiler**

90 **System Performance Test measurement equipment**

91 **Time Measurement -** Clock, Stopwatch

91A **Frequency Counters**

91B **Rotation Angle**

92 **Velocity**

92A **Fluid Flow**

92B **Tachometers** - RPM

93 **Acceleration**

93A **Vibration**

93B **Shock**

93C **Noise and Sound**

93D **Acoustic & Audio functional test**

94 **Load, Pressure, Force, Strain**

94A Weight, load, force

94B Pressure

94C Stress & Strain

95A **Temperature**

95B **Heat Flow**

96A **Optics - Color**

96B **Light & luminance, photometric**

97 **Energy & Power**

98 **Magnetics**

99 **Radioactivity**

100 **Electrical Test Equipment**:

101 **Voltage**

102 **Current**

103 **Resistance**

104 **Capacitance**

105 **Digital I/O**

106 **Frequency Counters**

107 **Waveform Generators**

108 **Battery Test Systems**

109 **ADAS and autonomous driving perception algorithms**

110 **Electrical component**
111 **Transistor**

200 **Environmental Test**:
Leak testing, thermal, thermal-vacuum, humidity, vibration, test chambers

700 **Gage Management & Calibration Gage R&R**

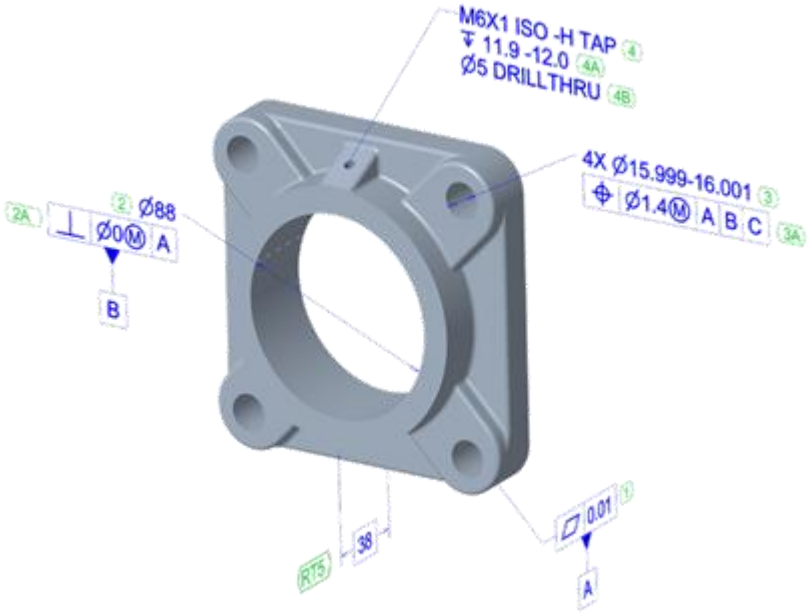
800 **Data Collection/Acquisition**:
Data collectors, line analyzers, data collection software, chart recorders, data loggers, data acquisition, bar-coding equipment

900 **Quality Systems & Services**

10.8 Annex H – Alternate Numerical Approach for Product Characteristic Tags

(Informative)

An alternative to use the numerical numbering <4> or the <PC004> number technique as described in section 1.3.1 would be to use a hybrid with a number and a letter. In the example figure, each group of product characteristics has a primary numeric designator and related facets or associated characteristics have alphabetical enumerations in order to logically group these items for reporting.



10.9 Annex I –Data Reporting Requirements (DRR) Augmentation

(Informative)

For future scope to this standard, a proposed Product Characteristic augmentation for Data Reporting Requirements is described here as informative for comment.

A Product Characteristic can be augmented by zero-to-many Data Reporting Requirements. Each Data Reporting Requirement references a data reporting method that reports this characteristic. Within the context that a Product Characteristic is augmented by a Data Reporting Requirement, that Product Characteristic can be called a Reporting Product Characteristic.

A Data Reporting Requirement addresses some product data reporting needs.

10.9.1 Identification

A DataReportingRequirement is identified by a DataReportingRequirementDesignator. The DataReportingRequirementDesignator is the identifying text that is optionally housed within a DataReportingRequirementDesignatorSymbol.

10.9.2 Data Structures

The information model for an optional DataReportingRequirement augmentation for a ProductCharacteristic is graphically shown below in …

TBD SysML Figure

10.9.3 Data Taxonomy

The DataReportingRequirement taxonomy consists of DataReportingRequirement types along with a designator and symbol. The DataReportingRequirement is identified by a DataReportingRequirementDesignator that can be shown within a DataReportingRequirementDesignatorSymbol.

The DataReportingRequirement must contain a DataReportingRequirementDesignator.

10.9.4 Data Objects

10.9.4.1 *DataReportingRequirement*

The optional DataReportingRequirement augmentation provides a Product Characteristic with a data reporting designation that consists of a DataReportingRequirement types with attributes that help control the reporting.

10.9.4.2 *DataReportingRequirementDesignator*

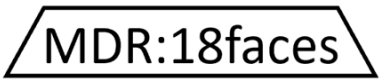
The DataReportingRequirementDesignator element provides the textual identification for the Data Reporting Requirement, which is housed by the DataReportingRequirementDesignatorSymbol. This designator has an alphanumeric string attribute that allows for a human-readable identifier of the intended DataReportingRequirement.

10.9.4.3 *DataReportingRequirementDesignatorSymbol*

The optional DataReportingRequirementDesignatorSymbol element is a human-readable symbol for housing the DataReportingRequirementDesignator.

The recommended symbol for identifying a DataReportingRequirementDesignator is an elongated trapezoid with its top side shorter than its bottom side. The elongated trapezoid symbol can accommodate the DataReportingRequirementDesignator and provide a visual connection to the ProductCharacteristicDesignator symbol.

A simple DataReportingRequirementDesignatorSymbol with a DataReportingRequirementDesignator is shown below:



An example of a DataReportingRequirementDesignatorSymbol associated with a ProductCharacteristicDesignatorSymbol is shown below.

<67>/MDR:6pts\

10.9.4.4 *DataReportingRequirement Text*

The recommended textual format for a DataReportingRequirementDesignator is with a "forward slash" character preceding and a "backward slash" character following the Designator. The general format is / DataReportingRequirementDesignator \. A simple textual DataReportingRequirementDesignator example is shown below:

/ MDR: X=15, X=30 \

An example of a textual DataReportingRequirementDesignator associated with a ProductCharacteristicDesignatorSymbol is shown below:

<PC019>/MDR:X=15,X=30\

10.9.4.5 *MeasurementDataReportingLocation*

The ASME Y14.45 has the capability of identifying the locations for reporting measurement results. This standard associates this measurement data reporting (MDR) location with a specific PC Reference Tag as part of the VPR. The designated report locations for measurement results are described as method C per ASME Y14.45. Reporting locations are one or more freeform strings per ASME Y14.45 Method C.

Format: / **MDR**:"*Reporting location"*\

Where

Reporting location is a string of a list of alphanumerics delimited by a comma ","

For profile of locations along an axis

Example: / **MDR**:"X=5, X=15"/

10.9.4.5.1 Reporting of Directional Deviations at Specific Locations

The example below shows a positional tolerance with PC Reference Tag <4> that communicates a MeasurementDataReportingLocation via ASME Y14.45 Method C, which designates measurement reporting locations at X=15 and X=30. This is called out using the trapezoidal symbol with the Measurement Data Reporting (MDR) callout and values of X=15 and X=30 which would be displayed as /MDR:X15,X30\ or using the trapezoidal symbol and supplemental geometry as shown in figure I-1. The measure reporting locations are designated 4-01 (for X=15) and 4-02 (for X=30).

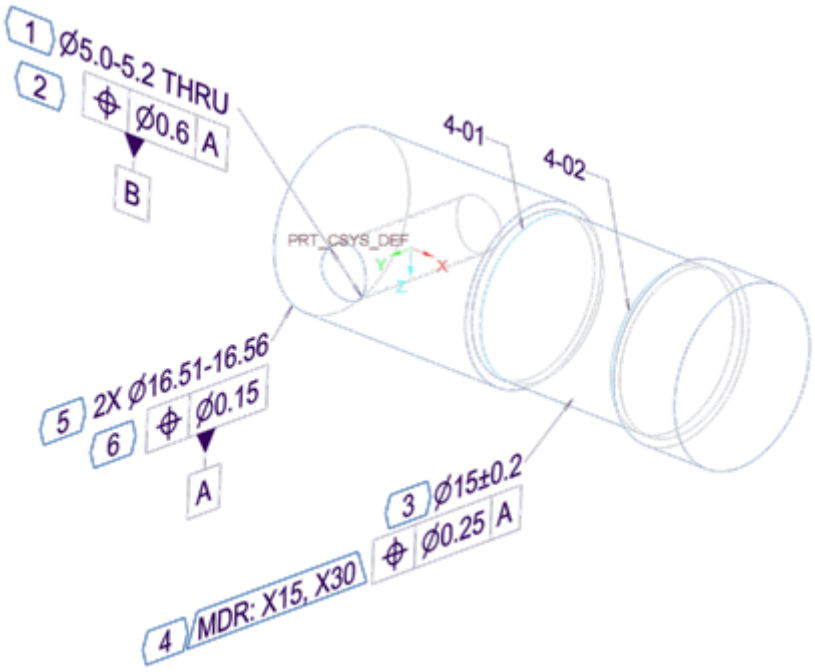


Figure I-1 Position Measurement Data Reporting Locations with Circle Geometry

These measurement data reporting locations 4-01 and 4-02 will display in the Data Report resolved position components in the Y and Z directions. They will be labeled 4-01Y, 4-01Z for the first location and 4-02Y, 4-02Z for the second location as shown in the section of the report below.

ASME Y14.45 Single Part Data Report Example												
Part #:			Drawing #:		Drawing Revision:		3D CAD Model #:					3D CAD Model Revision:
Part Name:					Part Serial #:		Inspection Plan #:					Report #:
Characteristic Identifier	Characteristic Type	*ASME Y14.45 Method	Reference Location	Characteristic Designator	Specification		**Calculated Acceptance Limit(s)	Reported Value	Accept (Y or N)	Tooling/Equipment	Nonconformance #	Comments
4	POS	B		4	Position | Ø0.25 | A		0.250	0.234	Y			
4-01Y		C		4	Location component for position			-0.040				Y location at X = 15, datum feature B used to stop rotation
4-01Z		C		4	Location component for position			0.110				Z location at X = 15, datum feature B used to stop rotation
4-02Y		C		4	Location component for position			0.050				Y location at X = 30, datum feature B used to stop rotation
4-02Z		C		4	Location component for position			-0.020				Z location at X = 30, datum feature B used to stop rotation
* ASME Y14.45, Measurement Data Reporting, includes methods A, B, and C as data categories that may be specified on a measurement plan or other document: Method A is attribute (pass/fail) data. Method B is variable data such as a size, profile, or position value. Method C is variable data to provide additional information, such as profile surface deviations or position location components.												
**Calculated acceptance limit(s) may include guard banding and/or bonus tolerance considerations.												

Figure I-2 Position Measurement Data Report Example

10.9.4.5.2 Reporting of Profile Deviations for Feature instances

The example below shows a profile tolerance with PC Reference Tag <PC080>.

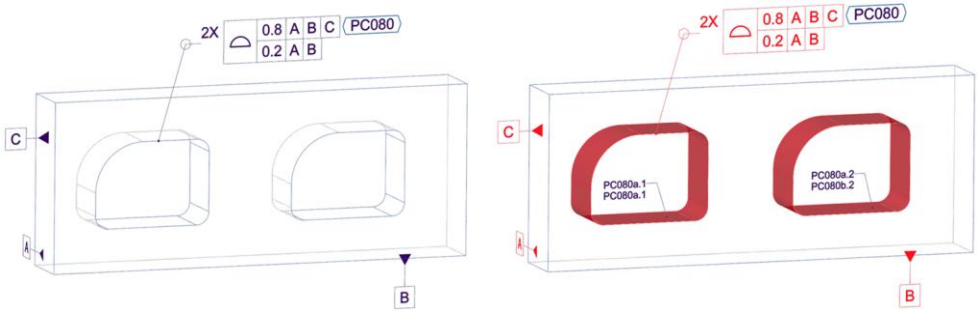


Figure I-2 Profile-Group Feature per Y1445C Measurement Data Reporting

10.9.4.5.3 Reporting of Profile Deviations at Face Locations

The example below shows a maximum profile deviation to be reported for each face within each pattern instance for PC Reference Tag <PC080>. The highlighted face refers to face 3 in group 2.

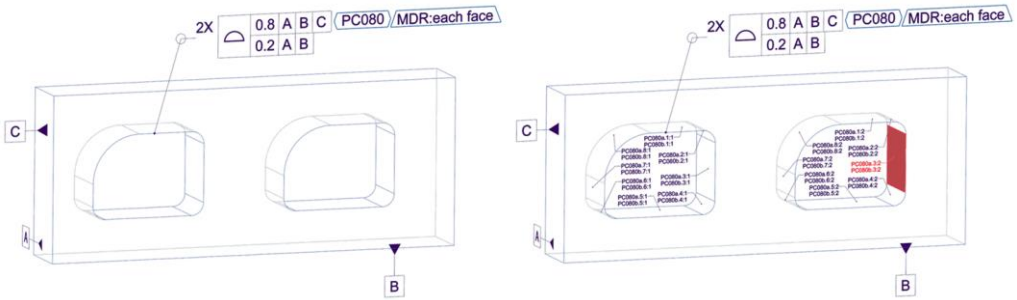


Figure I-3 Face Location Profile-Group per Y1445C Measurement Data Reporting

10.9.4.5.4 Reporting of Profile Deviations at Point Locations

The example below shows a profile tolerance with PC Reference Tag <PC080> that communicates a MeasurementDataReportingLocation via ASME Y14.45 Method C, which designates 15pts on each profile group feature. Some suggested location identifiers are identified in the figure below as PC080a.1-01 thru PC080a.1-15 on profile group 1 and PC080a.2-01 thru PC080a.2-15 on profile group 2.

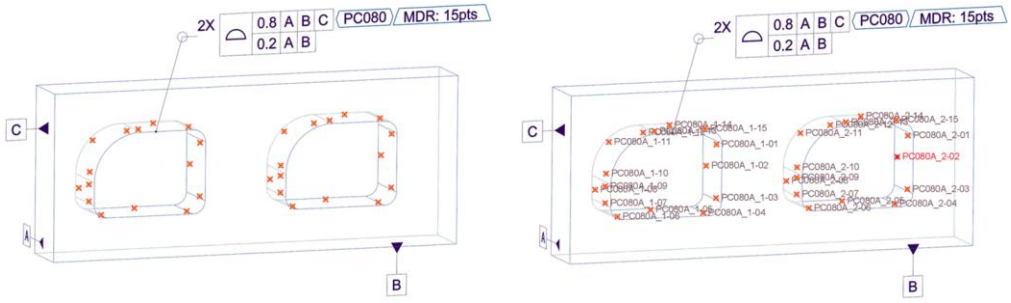


Figure I-4 Point-Location Profile-Group per Y1445C Measurement Data Reporting

10.9.4.5.5 Reporting of Profile Deviations at Point Locations on complex grouped patterns

The example below shows a complex group of patterned features. There is a combination of hole's diameters, depths, locations and other feature profiles repeated in five groups.

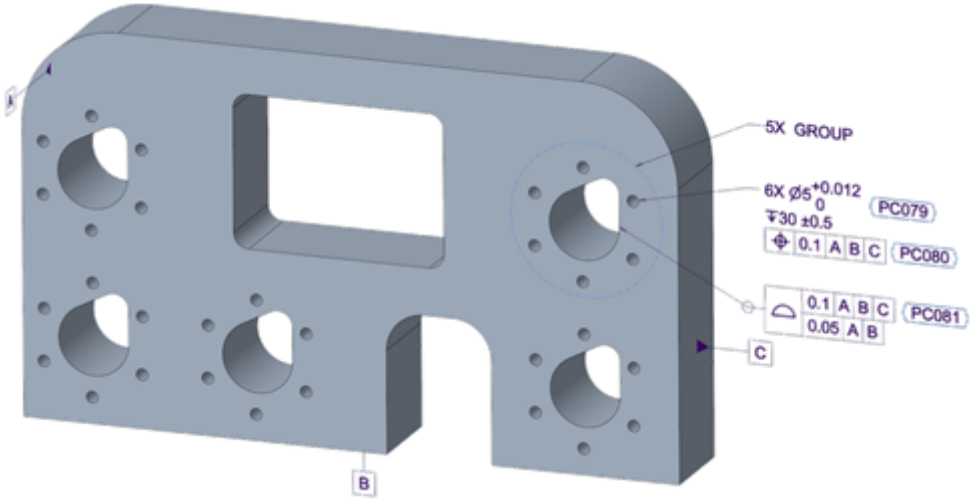


Figure I-5 Repetitive Feature Groups per Y1445C Measurement Data Reporting

Semantic query of the five-group callout shows the 35 separate features to be verified.

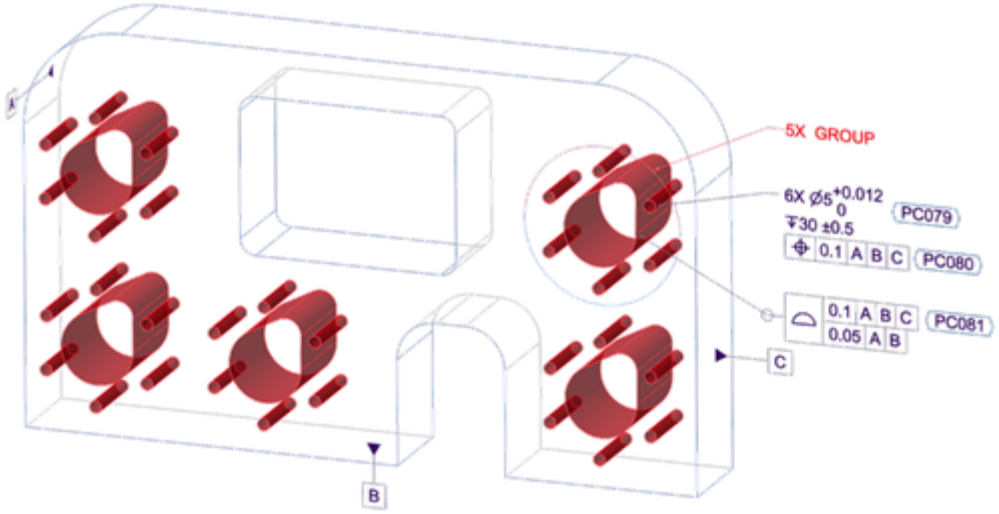


Figure I-6 Query Repetitive Feature Groups per Y1445C Measurement Data Reporting

The measurement data reporting is requested for the front and back surfaces shown which correspond to depths along the Z-axis of 40 (front) and 0 zero (back).

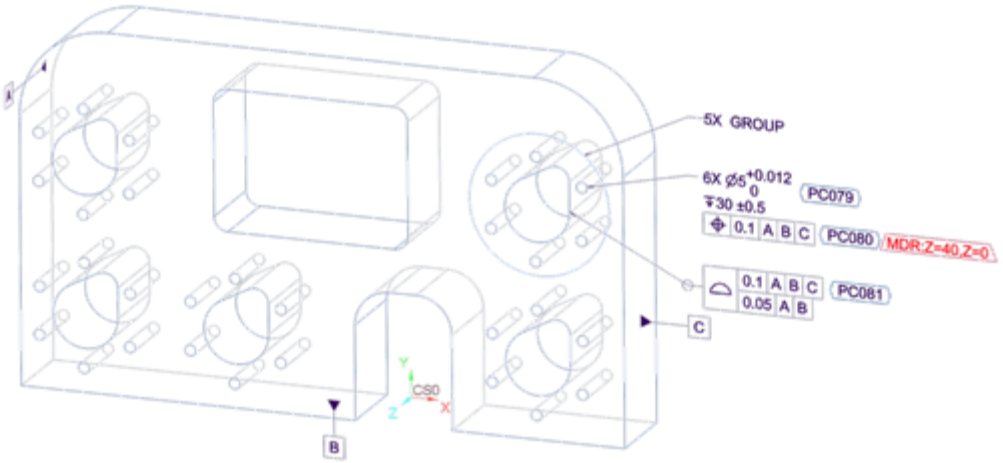


Figure I-7 Data Reporting Location of Hole per Y1445C Measurement Data Reporting

The front surface group of locations at Z=40 will be labeled -01.

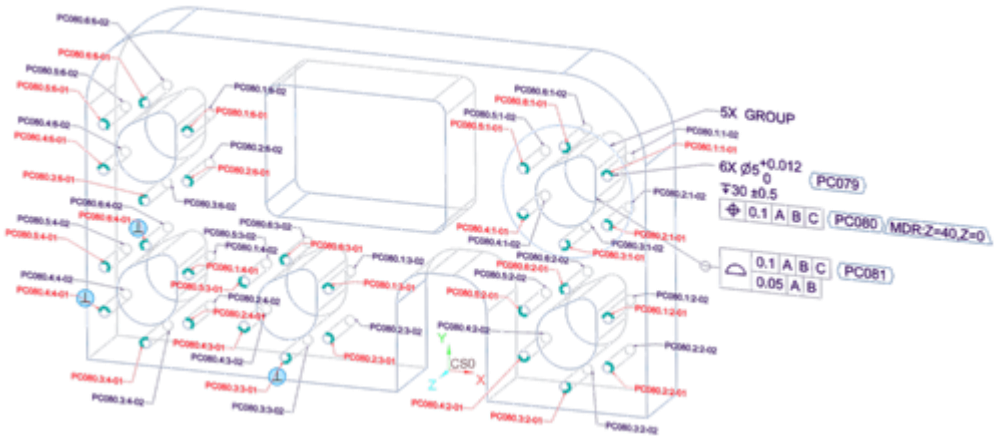


Figure I-8 Frontside Circle Locations of Holes per Y1445C Measurement Data Reporting

The backside surface group of locations at Z=0 will be labeled -02.

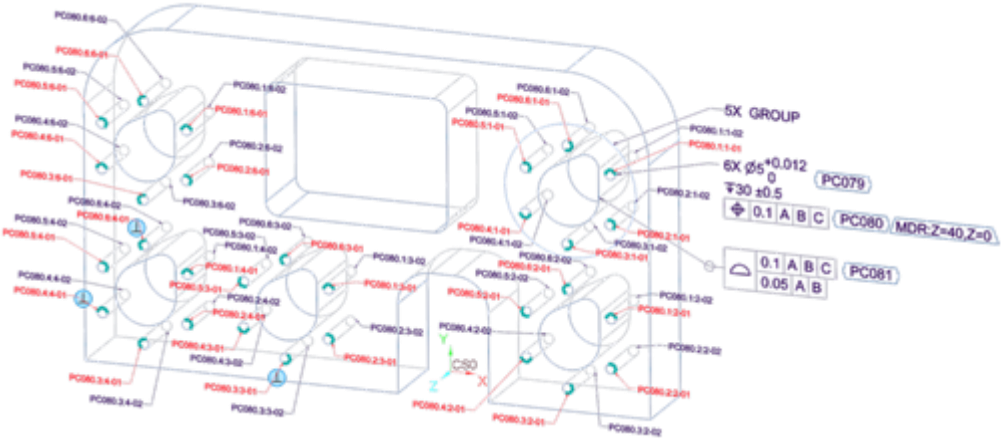


Figure I-9 Backside Circle Locations of Holes per Y1445C Measurement Data Reporting

An enlarged hidden line view shows the combination of front and back reporting groups -01 and -02 (red).

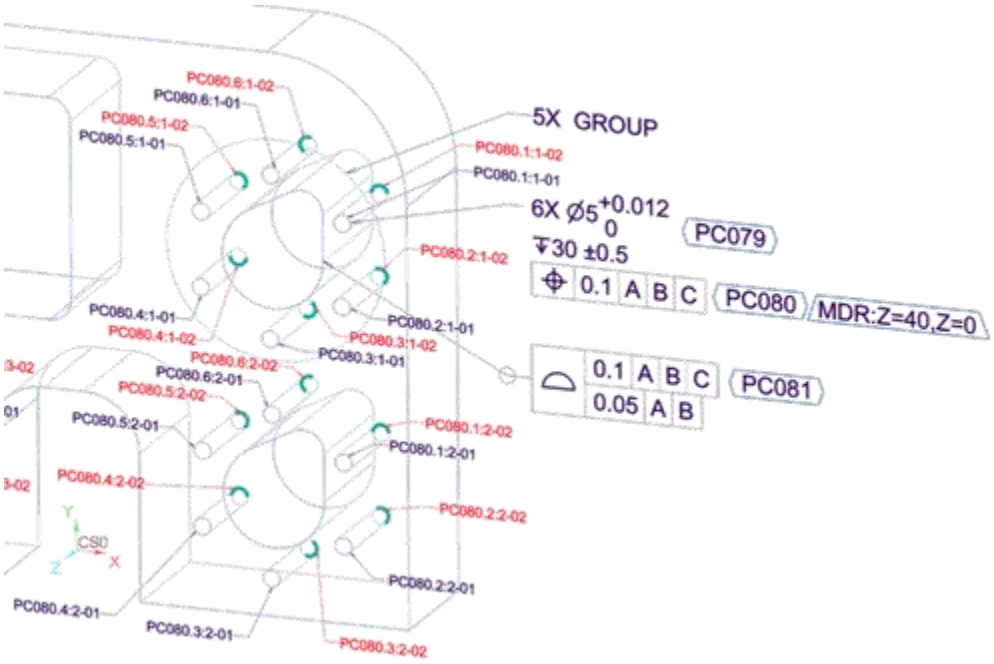


Figure I-10 Detail Circle Locations of Holes per Y1445C Measurement Data Reporting

10.9.4.6 *Risk Priority Number (RPN)*

As part of a product design, release, manufacturing, verification, maintenance, or approval process it may be desirable to report or attribute specific components with model-based characteristics related to a numerical assessment of the risk.

Common forms of assessment are risk priority number (RPN), a numerical assessment of the risk assigned to a failure mode when conducting a Failure Modes and Effects Analysis (FMEA) or Automotive Safety Integrity Level (ASIL), which is a risk classification system defined by the ISO 26262 standard for the functional safety of road vehicles.

The assessed value or component values can be placed in the product characteristic augmentation for data reporting which is a trapezoidal shape. The examples below show an analysis case number followed by the designation values that are derived through another process.

The data reporting augmentation shall be placed in-line with the characteristic designator. There may be additional augmentations before the data reporting augmentation.

Example of Automotive Safety Integrity Level (ASIL) data augmentation

Example of Risk Priority Number (RPN) data augmentation

Format: / opt{“case i*dentifier*” -} “*risk data form type***”:***”risk data string”*\

Where:

Case i*dentifier* is the *reference to the specific analysis being reported*

Risk data form type is the reference for the risk data information and they are:

- **RPN** =Risk Priority Number per FEMA
- **SIL** = Safety Integrity Level per ASIL

Risk data string is a string of alphanumerics

Example: /Case 44 - **RPN**:”S3 07 D4TBD”/

A simple DataReportingRequirementDesignatorSymbol with a DataReportingRequirementDesignator for RiskPriorityNumber is shown below:

An example of a DataReportingRequirementDesignatorSymbol associated with a ProductCharacteristicDesignatorSymbol for RiskPriorityNumber is shown below.

s

10.10 Annex J– QIF 3.0 Mapping / Comparisons

(Informative)

Mapping / comparison of MBC 1.0 elements with QIF 3.0 elements. The expectation is that the next version of QIF will incorporate many of the MBC elements.

MBC 1.0	Example	QIF 3.0
PRODUCT CHARACTERISTICS		
ProductCharacteristic		Characteristic
ProductCharacteristicBaseTag	PC007	CharacteristicDefinitionBase / CharacteristicDesignator / Designator
ProductCharacteristicBaseUUID	A UUID	CharacteristicDefinitionBase / CharacteristicDesignator / UUID
ProductCharacteristicReferenceTag	<PC007>	CharacteristicNominalBase / CharacteristicDesignator / Designator
ProductCharacteristicReferenceUUID	A UUID	CharacteristicNominalBase / CharacteristicDesignator / UUID
ProductCharacteristicInstanceTag	PC007a.1	CharacteristicItemBase / CharacteristicDesignator / Designator
ProductCharacteristicInstanceUUID	A UUID	CharacteristicItemBase / CharacteristicDesignator / UUID
		CharacteristicDesignator
ProductCharacteristicBaseUUID		CharacteristicDesignator / UUID
ProductCharacteristicDesignatorSymbol		CharacteristicDesignator / Balloon
		Balloon / BalloonLocation
		Balloon / BalloonStyle
eSymbolShapeType		CharacteristicBalloonType / BalloonStyle
eSymbolShapeType = elongatedHexagon	Symbol; Test: < >	CharacteristicBalloonType = ELONGATED_HEXAGON
FeatureReferenceID	1961; Cir4	FeatureItemBase / FeatureName
FeatureReferenceUUID	A UUID	FeatureItemBase / UUID
FeatureReferenceSequenceNumber		
CRITICALITY CLASSIFICATION		
CriticalityClassification		CharacteristicDesignator / Criticality
CriticalityClassificationDesignator		
CriticalityClassificationDesignatorSymbol		
eSymbolShapeType = forwardChevron	Symbol; Test: > >	CharacteristicBalloonType = FORWARD_CHEVRON
CriticalityLevel		
CriticalityLevelEnum		Criticality / LevelEnum
CriticalityLevelEnum = CR: Critical	CR	CriticalityLevelEnum = CRITICAL
CriticalityLevelEnum = MJ: Major	MJ	CriticalityLevelEnum = MAJOR
CriticalityLevelEnum = MN: Minor	MN	CriticalityLevelEnum = MINOR
CriticalityLevelEnum = KY: Key	KY	CriticalityLevelEnum = KEY
CriticalityLevelEnum = SG: Significant	SG	
CriticalityLevelEnum = UD User-Defined	UD	Criticality / OtherLevel

CriticalityLevelEnum = (Blank)		CriticalityLevelEnum = UNDEFINED
CriticalityCategory		Criticality / AreaEnum
CriticalityCategoryEnum = A: Assembly	A	
CriticalityCategoryEnum = C: Cosmetic	C	CriticalityAreaEnum = APPEARANCE
CriticalityCategoryEnum = D: Design	D	
CriticalityCategoryEnum = E: Environmental	E	
CriticalityCategoryEnum = F: Fit	F	CriticalityAreaEnum = FIT
CriticalityCategoryEnum = I: Interface	I	
CriticalityCategoryEnum = L: Life	L	
CriticalityCategoryEnum = M: Manufacture	M	
CriticalityCategoryEnum = P: Process	P	
CriticalityCategoryEnum = Q: Quality	Q	
CriticalityCategoryEnum = R: Regulatory	R	
CriticalityCategoryEnum = S: Safety	S	CriticalityAreaEnum = SAFETY
CriticalityCategoryEnum = T: Test	T	
CriticalityCategoryEnum = U: User-Defined	U	Criticality / OtherArea
CriticalityCategoryEnum = V: Service	V	
CriticalityCategoryEnum = W: Structural	W	
CriticalityCategoryEnum = X: Material	X	
CriticalityCategoryEnum = Z: Performance	Z	
Mission is classified via Criticality Levels		CriticalityAreaEnum = MISSION
Function is classified via Criticality Levels		CriticalityAreaEnum = FUNCTION
CriticalityCaveat		
PRODUCT REQUIREMENT ASSOCIATION		
ProductRequirementAssociationDesignator		
ProductRequirementAssociationDesignatorSymbol		
eSymbolShapeType = backwardChevron	Symbol; Test: < <	CharacteristicBalloonType = BACKWARD_CHEVRON
VERIFICATION PLAN REQUIREMENT		
VerificationPlanRequirementDesignator		
VerificationPlanRequirementDesignatorSymbol		
eSymbolShapeType = forwardParallelogram	Symbol; Test: / /	
VerificationMethod		MeasureFeatureMethodBaseType
VerificationMethodEnum = CERT	CERT	
VerificationMethodEnum = CMM	CMM	CoordinateMeasureFeatureMethodType,
		AutocollimatorMeasureFeatureMethodType,
		LaserRadarMeasureFeatureMethodType, ,
		UniversalLengthMeasureFeatureMethodType
VerificationMethodEnum = COLR	COLR	
VerificationMethodEnum = CSMT	CSMT	
VerificationMethodEnum = GAGE	GAGE	GageMeasureFeatureMethodType,
VerificationMethodEnum = HAND	HAND	ManualMeasureFeatureMethodType,
VerificationMethodEnum = LABT	LABT	

VerificationMethodEnum = MTRL	MTRL	
VerificationMethodEnum = NVMR	NVMR	
VerificationMethodEnum = OPTC	OPTC	MicroscopeMeasureFeatureMethodType, ProfileProjectorMeasureFeatureMethodType,
VerificationMethodEnum = OTHR	OTHR	OtherMeasureFeatureMethodType,
VerificationMethodEnum = PRBE	PRBE	CalibratedComparatorMeasureFeatureMethodType,
VerificationMethodEnum = SCAN	SCAN	ComputedTomographyMeasureFeatureMethodType, LaserTrackerMeasureFeatureMethodType TheodoliteMeasureFeatureMethodType,
VerificationMethodEnum = SUPL	SUPL	
VerificationMethodEnum = SURF	SURF	
VerificationMethodEnum = TBD	TBD	
VerificationMethodEnum = ULTS	ULTS	
VerificationMethodEnum = UDF	UDF	ExternalReferenceMeasureFeatureMethodType,
VerificationMethodEnum = URSD	URSD	
VerificationMethodEnum = VISL	VISL	

10.11 Annex K – Control Limit Mapping to ISO 3534-2 and ISO 22514-3 Symbols

(Informative)

ISO 3534-2	Section	Variable	Term	
	2.4.7	ACL	acceptance control limit	control limits for an acceptance control chart which permit some assignable shift in process level based on specified requirements, provided subgroup variability is subject to only random causes under statistical control
	4.7.1	AOQ	average outgoing quality	average quality level of outgoing product for a given value of incoming product quality
	4.7.2	AOQL	average outgoing quality limit	maximum average quality level over all possible values of incoming product quality level
	2.7.2	Cp	process capability index	index describing process capability in relation to specified tolerance
	2.6.5	Cpk	process capability index, minimum	smaller of upper process performance index and lower process performance index
	2.7.3	CpkL	process capability index, lower	index describing process capability in relation to the lower specification limit
	2.7.4	CpkU	process capability index, upper	index describing process capability in relation to the upper specification limit
	2.4.9	LCL	control limit, lower	control limit that defines the lower control boundary
	4.6.14	LQL	limiting quality level	quality level which, for the purposes of acceptance sampling inspection is the limit of an unsatisfactory process average when a continuing series of lots is considered
	2.6.2	Pp	process performance index	index describing process performance in relation to specified tolerance
	2.6.5	Ppk	process performance index, minimum	smaller of upper process performance index and lower process performance index
	2.6.3	PpkL	process performance index, lower	index describing process performance in relation to the lower specification limit

	2.6.4	PpkU	process performance index, upper	index describing process performance in relation to the upper specification limit
	2.7.7	Qk	process variation index	measure of variation expressed in terms of the target value
	2.4.8	UCL	control limit, upper	control limit that defines the upper control boundary
ISO 22514-3	**Section**	**Variable**	**Term**	
	7.6.2.1	Pm	machine performance index	index describing the level of uncertainty existing with a point estimate of a machine performance index based on a small sample of data
	7.6.2.2	Pmk	machine performance index, minimum	smaller of upper machine performance index and lower machine performance index
	7.6.2.2	PmkU	machine performance index, upper	index describing machine performance in relation to the upper specification limit
	7.6.2.2	PmkL	machine performance index, lower	index describing machine performance in relation to the lower specification limit

10.12 Annex L – Variations in implementation

(Informative)

Software applications implementing this MBC standard may use common user-interface features beyond the scope of this standard without affecting its meaning. These may include, but are not limited to:

- Arranging data in a tree control
- Arranging data in a list control
- Arranging data in a grid control
- Arranging data in a control which is some hybrid of the above
- Dynamic data linking to a secondary application such as a spreadsheet
- Identifying aspects of the standard with icons in command controls or display controls
- Arranging data on named levels, layers, or views
- Pop-up information from pointer hovering, right/left click, right/left double click, gesturing, etc.
- Pop-up information from touch screen tap, double-tab, two finger expansion, gesturing, etc.
- Bolding, style, color
- Virtual reality display
- Augmented reality display

Similarly, physical world implementations beyond printed graphical views will not affect the standard's meaning such as, but not limited to:

- Hand-written on part
- Printed on part
- Label affixed to part
- Display projected on part
- Augmented reality display

Note: This annex establishes as prior art any number of trivial innovations that may be patented and cause damage to or hinder the industry acceptance of this standard.

10.13 Annex M – MBC Feedback Request

(Informative)

The DMSC requests user comment and suggestions on the content of this this important standard, both from the normative sections and/or the informative annexes.

Please provide feedback to email address: Director@QIFStandards.org.