

*Dimensional Metrology
Standards Consortium*

DMSC

Quality Information Framework (QIF) – An Integrated Model for Manufacturing Quality Information

Part 6: QIF Rules Information Model and XML Schema File Version 2.0



QIF Version 2.0

ANSI/QIF Part 6–2014

Dimensional Metrology Standards Consortium, Inc. (DMSC)

Prepared and Published by:
Dimensional Metrology Standards Consortium, Inc. (DMSC)
1350 Alsbury Blvd., #514
Burleson, Texas 76028
Tel: (817) 461-1092
email: support@qifstandards.org

www.QIFStandards.org

All rights reserved worldwide.

Copyright by the DMSC. Permission is hereby granted, free of charge, to make copies of this document and use the content in any manner as long as this copyright notice and permission notice are included in every copy.

Contents

Foreword.....	v
Introduction	vii
1 Scope	1
1.1 Contents of this document	1
1.2 QIF information model application architecture	2
2 Conformance	3
3 Normative references.....	4
4 Terms and definitions.....	5
4.1 General QIF terms referenced in the QIF Rules application area	5
4.1.1 feature	5
4.1.2 characteristic.....	5
4.2 Terms defined for the QIF Rules application area	5
4.2.1 Boolean condition	5
4.2.2 rule	5
4.2.3 point sampling strategy	5
4.2.4 sampling rigor	6
5 Symbols and abbreviated terms.....	7
6 The QIF Rules information model	9
6.1 Design principles of QIF Rules.....	9
6.2 QIF Rules schema files.....	10
6.3 QIF Rules data types	10
6.3.1 QIFRulesType.....	11
6.3.2 FeatureRulesType	11
6.3.3 IfThenElseFeatureRulesType	12
6.3.4 MaxFeatureRulesType.....	13
7 Data dictionary for QIF Rules information model.....	15
Annex A – Location of QIFRules.xsd.....	16

Annex B - Graphical conventions of the data dictionary	17
Annex C – Data dictionary for schema file QIFRules.xsd	20
Bibliography	99

Figures

Figure 1 – QIF information model application architecture 2

Figure 2 – *QIFRulesType*.....11

Figure 3 – *FeatureRulesType*11

Foreword

The Dimensional 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). The mission of the DMSC is to identify urgently needed standards in the field of dimensional metrology, and to promote, foster, and encourage the development and interoperability of these standards, along with related and supporting standards that will benefit the industry as a whole. More information about the DMSC can be found at www.dmsc-inc.org.

The Quality Information Framework (QIF) information model was developed by domain experts from the manufacturing quality (that is metrology) community representing a wide variety of industries and quality measurement needs. Specifically for the QIF Rules work, contributors include:

Main:

- Metrosage
- National Institute of Standards and Technology
- Mitutoyo America
- Honeywell Federal Manufacturing & Technologies

Support:

- Manufacturing Technology Centre
- University of North Carolina, Charlotte
- Lockheed Martin

More information about DMSC's QIF effort can be found at www.qifstandards.org.

The bulk of the work on this document was performed by the QIF Rules Working Group, approved and revised as needed by the Quality Information Framework (QIF) Working Group, and given final approval for ANSI balloting by the DMSC's Quality Measurement Standards (QMS) Committee.

Version 2.0 of QIF consists of the following parts, under the general title *Quality Information Framework (QIF) — An Integrated Model for Manufacturing Quality Information*:

Part 1: Overview and Fundamental Principles Version 2.0

Part 2: QIF Library Information Model and XML Schema Files Version 2.0

Part 3: QIF Model Based Design (MBD) Information Model and XML Schema File Version 2.0

Part 4: QIF Plans Information Model and XML Schema File Version 2.0

Part 5: QIF Resources Information Model and XML Schema File Version 2.0

Part 6: QIF Rules Information Model and XML Schema File Version 2.0

Part 7: QIF Results Information Model and XML Schema File Version 2.0

Part 8: QIF Statistics Information Model and XML Schema File Version 2.0

HTML-based data model viewer

The DMSC will make available an html-file based data dictionary for the entire QIF information model as an aid to understanding QIF. This data dictionary is non-normative material, but describes the normative content of the QIF information model. The html files facilitate viewing the complete information model, including all six application areas and Library content, using pictures and text. A user has the ability, through an internet browser, to follow navigation links forward and backward through the information model description using mouse clicks.

Introduction

In an attempt to encourage uniformity in measurement strategies across a large corporation or supply chain, a common approach is to distribute a set of measurement rules (or sometimes called measurement templates). This will give the Coordinate Measuring Machine (CMM) (or any other Dimensional Measurement Equipment (DME)) inspection program creator a guideline for implementing programs that conform to best organizational practices.

QIF Rules is meant to give a CMM programmer (or CMM software package) a set of standard practices regarding how individual features of an arbitrary product should be measured based on a number of possible criteria (such as applicable characteristics, tolerance value, feature geometry type, feature size, etc.).

It is important to emphasize that the measurement rules described in this document are *not* an attempt to give DME programmers a precise set of instructions regarding how to carry out any given measurement on a given part. (Such a detailed set of instructions would be transmitted using QIF Part 4: QIF Plans.)

The QIF Rules model does not propose any specific set of rules. It only provides a format for rules files in order that sets of rules may be exchanged easily and unambiguously. Any company can use the QIF Rules model to model its own rules for the following items on a per feature basis:

- the number or density of CMM hit points to use
- the pattern in which the CMM hit points should be distributed (point sampling strategy)
- the feature fitting algorithm to use.

Quality Information Framework (QIF) — an integrated model for manufacturing quality information

Part 6: QIF Rules information model and XML schema file

1 Scope

1.1 Contents of this document

This Part of the standard defines a QIF application information model for a portion of the QIF manufacturing quality information model designated QIF Rules.

Part 1 of the standard gives an overview of QIF. As described in Part 1, an XML data file conforming to an XML schema model is called an instance file. The root of every QIF instance file is a QIFDocument, but different QIF instance files may focus on different QIF applications. A QIFDocument containing a Rules element may be regarded as a QIF Rules instance file.

A QIF Rules instance file will consist of a set of “Boolean Conditions” and a corresponding set of “Rules”. A Boolean Condition is a statement which can be unambiguously evaluated as true or false. For example: “Joe’s car is red” would evaluate to “true” if and only if the color of Joe’s car is red. Any given Boolean Condition may have one or more corresponding Rules that should be taken depending on the value of the Boolean Condition. For example, if “Joe’s car is red” evaluates to true, the corresponding Rule may be requested: “Borrow Joe’s car.” QIF Rules simply consists of a set of Boolean Conditions pertaining to the feature in question, and a set of measurement-related Rules that should be carried out depending on these conditions.

The information model consists of definitions for data types, elements, the logical relationships between them, and the semantics of the quality information. The information model, defined using the XML Schema definition (XSDL) language, is scoped to be a digital data exchange mechanism that can be easily incorporated in application software developed by commercial solution vendors that implement manufacturing quality systems.

The QIFRules.xsd schema file (see Annex A – Location of QIFRules.xsd) defines the top level information model for rules. QIF Rules also draws from elements in the QIF Library, particularly those in the GenericExpressions.xsd and Expressions.xsd schema files. The types and relationships of information contained in a QIF Rules instance file are governed by the schema. The way in which data is formatted for QIF Rules instance files is prescribed by the rules of XML and the rules for how instance files conform to schema files.

“Annex C – Data dictionary for schema file QIFRules.xsd” is a detailed data dictionary for QIFRules.xsd.

1.2 QIF information model application architecture

“Figure 1 – QIF information model application architecture” shows a high level view of the current QIF Applications information models architecture. The QIF foundation is its suite of information model definitions. At its core is the reusable QIF Library which contains components that are referenced throughout the comprehensive quality information model thereby ensuring interoperability and extensibility. Around the core are the QIF Applications. All of the applications shown in the figure other than QIF Execution are included in QIF version 2.0. This document describes the QIF Rules information model. The QIF Rules information model transfers Measurement Rules using information items in the QIF Library (for example, feature and characteristic).

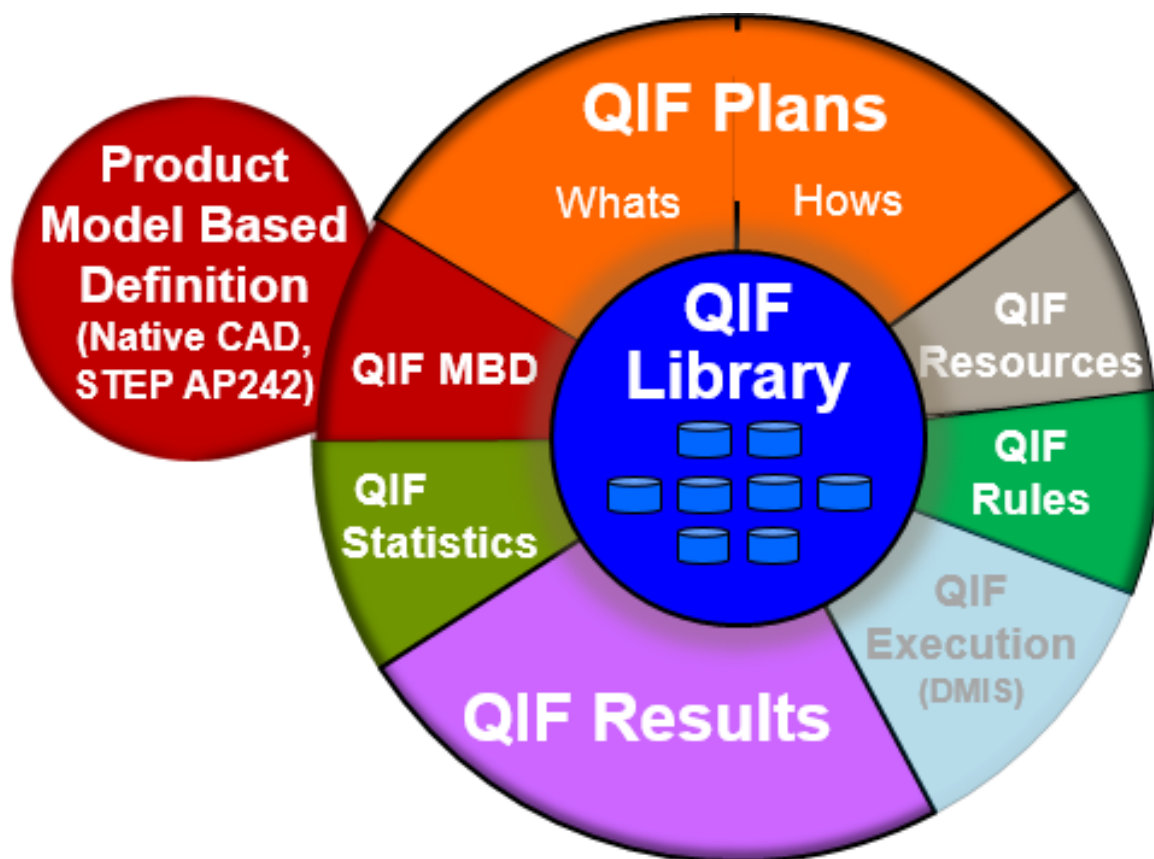


Figure 1 – QIF information model application architecture

2 Conformance

Software products that implement this specification to write QIF Rules instance files must:

- follow the rules of XML when writing QIF Document instance files
- generate instance files that validate against the QIF Document schema
- employ semantics of the information written that complies with the referenced standards and with the data dictionary in this specification.

Software products that implement this specification to read QIF Rules XML instance files must:

- be able to read any valid QIF Rules XML instance file and extract all numerical and semantic data correctly.

3 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ANSI/DMIS 105.2, Part 1-2009, *Dimensional Measuring Interface Standard, DMIS 5.2 Standard, Part 1*. Also available as ISO 22093:2011 *Industrial automation systems and integration -- Physical device control -- Dimensional Measuring Interface Standard (DMIS)*

ASME B1.7 - 2006, *Screw Threads: Nomenclature, Definitions, and Letter Symbols*

ASME Y14.36 - 1996, *Surface Texture Symbols*

ASME Y14.6 - 2001, *Screw Thread Representation*

ASME Y14.5M-1994 (reaffirmed 2004), *Dimensioning and Tolerancing - Engineering Drawing and Related Documentation Practices*

ASME Y14.5-2009, *Dimensioning and Tolerancing - Engineering Drawing and Related Documentation Practices*

ASME Y14.41 (2003), *Digital Product Definition Data Practices*

Extensible Markup Language (XML) 1.0 (Fifth Edition), W3C Recommendation 26 November 2008

ISO/IEC 9834-8:2008. *Information technology -- Open Systems Interconnection -- Procedures for the operation of OSI Registration Authorities: Generation and registration of Universally Unique Identifiers (UUIDs) and their use as ASN.1 Object Identifier components*

ISO/IEC 11578:1996: "Information technology - Open System Interconnection - Remote Procedure Call (RPC)"

ISO/IEC Guide 99:2007 (E/F) – *International vocabulary of metrology – Basic and general concepts and associated terms* (VIM)

XML Schema Part 1: Structures Second Edition, W3C Recommendation 28 October 2004

XML Schema Part 2: Datatypes Second Edition, W3C Recommendation 28 October 2004

ISO 14406:2010, *Geometrical product specifications (GPS) – Extraction*

4 Terms and definitions

For the purposes of this document, the following terms and definitions apply. The terms and definitions used in this document are given in Part 1 of this standard. The first group is general QIF terms. The second group is terms defined to describe the information in the QIF Rules application area.

4.1 General QIF terms referenced in the QIF Rules application area

The following general terms are referenced by the QIF Rules application area and defined in the QIF Part 1 document. The terms are repeated here for the convenience of the reader.

4.1.1 feature

a physical portion of a part such as a surface, pin, hole, or slot or its representation on drawings, in models, or in digital data files [ASME Y14.5]

4.1.2 characteristic

a control placed on an element of a feature such as its size, location or form and may be a specification limit, a nominal with tolerance, a feature control frame, or some other numerical or non-numerical control

4.2 Terms defined for the QIF Rules application area

The following terms are introduced by the QIF Rules application area.

4.2.1 Boolean condition

a statement which can be unambiguously evaluated as true or false

EXAMPLE “The feature is a cylinder” would evaluate to “true” if and only if the feature in question is a cylinder feature.

4.2.2 rule

in the context of QIF Rules, a specification of the type of measurement activity that should be carried out, given a Boolean condition context

NOTE Any given Boolean condition will have a corresponding rule that should be taken if the Boolean condition evaluates to true. For example, if “the feature is a cylinder” evaluates to true, the corresponding rule may be requested: “measure 13 points”.

4.2.3 point sampling strategy

the geometric pattern that is used to distribute the measurement points on a given feature

NOTE The range of possible values is taken from ISO-14406:2010. Examples include: an orthogonal grid, a helix, a “birdcage”, etc.

4.2.4 sampling rigor

the level of stringency that will be used to measure a given feature

NOTE The exact meaning of each level of rigor and the number of possible levels is meant to be defined and evaluated by the user.

5 Symbols and abbreviated terms

ANSI	American National Standards Institute
ASCII	American Standard Code for Information Interchange
ASME	American Society of Mechanical Engineers
CAD	Computer-Aided Design
CAIPP	Computer-Aided Inspection Process Planning
CAM	Computer-Aided Machining, or Computer-Aided Manufacturing
CMM	Coordinate Measuring Machine
COTS	Commercial Off-The-Shelf
DME	Dimensional Measuring Equipment
DMIS	Dimensional Measuring Interface Standard
DMSC	Dimensional Metrology Standards Consortium
DRF	Datum Reference Frame
ERP	Enterprise Resource Planning
GD&T	Geometric Dimensioning and Tolerancing
GPS	Geometrical Product Specifications
GUID	Globally Unique Identifier
ISO	International Organization for Standardization
MES	Manufacturing Execution Systems
MRI	Measurement Resources Information
MRP	Materials Resource Planning
MSA	Measurement Systems Analysis
PDPMI	Product Definition with Product Manufacturing Information
PMI	Product Manufacturing Information
QIF	Quality Information Framework
QMS	Quality Measurement Standards (a DMSC committee)

QPId	QIF Persistent Identifier
R&R	Repeatability and Reproducibility
SI	The International Systems of Units
SPC	Statistical Process Control
SQC	Statistical Quality Control
STEP	Standard for the Exchange of Product model data (ISO 10303)
UUID	Universally Unique Identifier
XML	eXtensible Markup Language
XSDL	XML Schema Definition Language

6 The QIF Rules information model

6.1 Design principles of QIF Rules

This section describes the highest level elements of the QIF Rules information model.

QIF Rules consists of a set of “Boolean Conditions” and a corresponding set of “Rules”, as in the following pseudocode model:

```
If ("Boolean Condition") then
{
    Corresponding "Rule"
}
```

Here is an example, also in pseudocode:

```
If (FeatureType is Cylinder) then
{
    Number of Points = 64
}
```

This if/then clause could optionally be followed by a series of “else if” and finally an “else” statement. These “else if” and “else” statements are modeled after the commonly seen “if/else if/else” statements found in applications of computer science.

Because of the nature of XML, the actual implementation of the “if/else if/else” statements can be counter-intuitive at first sight to a developer who is used to more traditional, structured languages like C.

Here is an example of a simple example in XML. In this example, we examine if the surface area of the feature is greater than 10 units of area. If it is, then a minimum of 25 points is requested. Otherwise (“else”), a minimum of 10 points is requested.

```
<MaxFeatureRules>
  <IfThenSurfaceRule>
    <GreaterThan>
      <FeatureArea/>
      <ArithmeticConstant val="10"/>
    </GreaterThan>
    <ThenPoints>
      <MinPoints>25</MinPoints>
    </ThenPoints>
  </IfThenSurfaceRule>
  <Else>
    <ThenPoints>
      <MinPoints>10</MinPoints>
    </ThenPoints>
  </Else>
</MaxFeatureRules>
```

Below is a list of Boolean Conditions supported by QIF Rules:

- Characteristic Type (is equal to a given characteristic type)
- Tolerance Value (compared to a given value)
- Feature Type (is equal to a given feature type)
- Shape class (part shape class is equal to one of prismatic, thin walled, etc.)
- Is feature a datum?
- Feature size (compared to a given value)
- Feature surface area/length (compared to a given value)
- User defined Sampling Rigor (compared to a given value)

And below, a list of possible Rules that can be triggered by the above Boolean Conditions:

- Number of measurement points
- Measurement point density
- Point sampling strategy (taken from ISO-14406:2010)
- Feature fitting algorithm to use

A Condition/Rule pair is applied to a Feature. If more than one characteristic applies to a feature, and a condition uses a characteristic type test, the condition should be tested for each characteristic applied to the feature.

Since measurement points are taken on Features (not Characteristics), there may often arise a situation where two or more conflicting Rules may apply to a given Feature. When this is the case, the Rule with the higher number of points should be used. If this happens when two sampling strategies are specified, then it is up to the evaluating system to make the appropriate decision. (Exception: the ***IfThenElseFeatureRulesType***; see section 6.3.3.)

6.2 QIF Rules schema files

The QIF Rules schema model includes the information items from the QIFRules.xsd schema file and several of the schema files from the QIF Library. As with other QIF Parts, the QIF Library are incorporated into the schema by a chain of “*include*” directives starting in the QIFRules.xsd schema file.

6.3 QIF Rules data types

This section describes the major elements and types in the QIF Rules model.

6.3.1 QIFRulesType

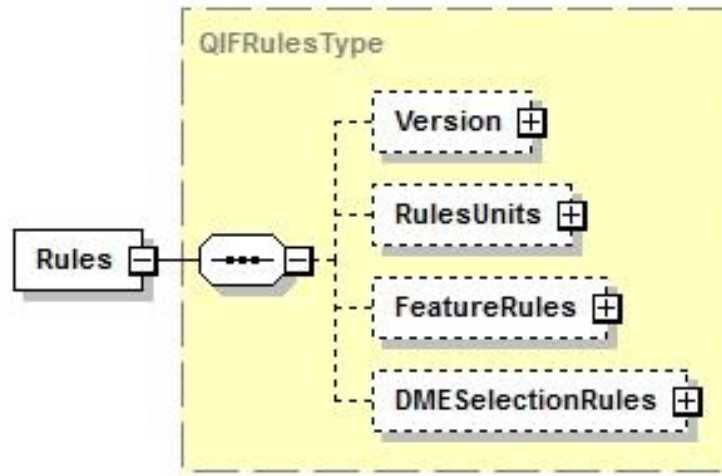


Figure 2 – *QIFRulesType*

The top-level data object under **QIFDocument** in a QIF Rules instance file is a **Rules** *element* of type **QIFRulesType**, as shown in Figure 2 – *QIFRulesType*. The **Rules** *element* contains a set of rules for features in its **FeatureRules** *element*, which is of type **FeatureRulesType**. It also appears to contain a set of rules for selecting dimensional measurement equipment in the **DMESelectionRules** *element*, but that is only a stub; it may be given meaning in future versions of QIF.

6.3.2 FeatureRulesType

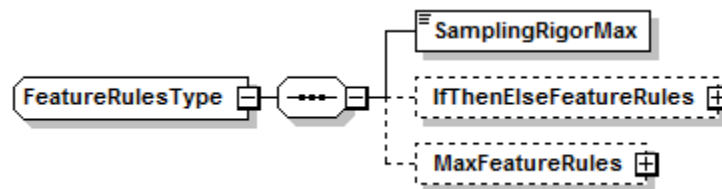


Figure 3 – *FeatureRulesType*

The **FeatureRulesType** models a collection of sets of rules for selecting quality measurement items. Currently it contains a complete model of rules for selecting any or all of the following for a given feature:

- the quantity (number or density) of hit points
- the strategy for selecting hit points
- a feature fitting algorithm.

As shown in Figure 3 – *FeatureRulesType*, the **FeatureRulesType** provides two methods of making selections, **IfThenElseFeatureRules** and **MaxFeatureRules**. Either method or both methods may be used. If the **IfThenElseFeatureRules** and **MaxFeatureRules** are both used

for quantity of points, both sets of rules should be applied and the greatest value should be used. Also if both are used and they give different point sampling strategies or feature fitting algorithms, the evaluating system may use any of those strategies or algorithms.

Since a given feature *F* may be associated with zero to many characteristics *C*₁ ... *C*_{*n*}, if there is any characteristic associated with *F*, the rules should be evaluated for each *FC*_{*i*} pair, and the number or density of hit points to use for *F* should be set to the maximum value.

If a number of hit points must be compared with a density of hit points, the evaluating system should find and use the area of the feature to convert density values to numbers of points.

The rules are to be evaluated in an environment in which the **SamplingRigor** has been set to some positive integer less than or equal to the **SamplingRigorMax** element so that the **SamplingRigorIsType** Boolean expression can be evaluated. **SamplingRigor** values start at 1 and end at the value of the **SamplingRigorMax** element. For example, if **SamplingRigorMax** is 2, the values 1 and 2 might represent **LOW** and **HIGH**, respectively. If the **SamplingRigorMax** is 4, the values 1 through 4 might represent **LOW**, **MEDIUM**, **HIGH**, and **SUPER**, respectively.

Numerical quantities that are constants, feature parameters, or characteristic parameters may be used in conditions.

The numerical quantities, **FeatureLength** and **FeatureArea**, may also be used in conditions. The evaluating system must set the value of **FeatureLength** for the feature being processed if **FeatureLength** is used in any condition. The evaluating system must set the value of **FeatureArea** for the feature being processed if **FeatureArea** is used in any condition.

The rules are not intended for picking the quantity of points or the point sampling strategy to use for measuring a characteristic that is not associated with any feature.

6.3.3 IfThenElseFeatureRulesType

The **IfThenElseFeatureRulesType** is a set of rules for selecting (1) the number or density of hit points for a single feature, possibly with an associated characteristic and/or (2) a strategy for selecting the points. To evaluate an **IfThenElseFeatureRulesType**, each **IfThenFeatureRule** should be considered in order until the 'if' part of one of them evaluates to true, at which point the 'then' part of the rule should be evaluated and the value of the point quantity and/or point sampling strategy returned. In this case, the remainder of the **IfThenFeatureRules** should be ignored. If the 'if' part of no **IfThenFeatureRule** evaluates to true, then the 'then' part of the Else should be evaluated and returned.

Below is an example of an **IfThenElseFeatureRulesType** rule.

```
<IfThenElseFeatureRules>
  <IfThenCylinderRule>
    <ThenPoints>
      <NumberOfPoints>23</NumberOfPoints>
    </ThenPoints>
  </IfThenCylinderRule>
```

```

<IfThenArcRule>
  <ThenPoints>
    <PointDensity>0.8</PointDensity>
  </ThenPoints>
</IfThenArcRule>
<Else>
  <ThenPoints>
    <MinPoints>13</MinPoints>
  </ThenPoints>
</Else>
</IfThenElseFeatureRules>

```

In this example, we first test if the Feature is a cylinder. If so, then we request 23 points and have completed the **IfThenElseFeatureRulesType** rule. Otherwise, we test if the Feature is an arc. If it is, then we request a point density of 0.8 (with the exact placement of points to be calculated by the evaluating system). If the Feature was an arc, then we are finished evaluating. Otherwise, we fall into the “Else” statement: a minimum of 13 points.

6.3.4 MaxFeatureRulesType

The **MaxFeatureRulesType** is a set of rules for selecting (1) the number or density of hit points for a single feature, possibly with an associated characteristic and/or (2) a point sampling strategy for the points. To evaluate a **MaxFeatureRulesType**, keep track of a current answer for point quantity, which is initially set to zero. Each **IfThenFeatureRule** should be considered in order. If the 'if' part of a rule evaluates to true, the 'then' part of the rule should be evaluated and if that value is greater than the current answer, the current answer should be set to that value. If the 'if' part of no **IfThenFeatureRule** evaluates to true, so that the current answer is still zero after all **IfThenFeatureRules** have been processed, then the 'then' part of the Else should be evaluated and the current answer set to that value. The returned value for point quantity is the final value of the current answer. In addition to keeping track of the current answer for point quantity, the evaluating system should collect all the strategies and feature fitting algorithms from the 'then' parts that were evaluated.

Below is an example of the **MaxFeatureRulesType** rule.

```

<MaxFeatureRules>
  <IfThenSurfaceRule name="FirstRule">
    <GreaterThan>
      <FeatureArea/>
      <ArithmeticConstant val="2"/>
    </GreaterThan>
    <ThenPoints>
      <MinPoints>4</MinPoints>
    </ThenPoints>
  </IfThenSurfaceRule>

  <IfThenSurfaceRule name="SecondRule">
    <And>

```

```

    <CharacteristicIs val="SURFACEPROFILE"/>
    <LessThan>
      <ArithmeticCharacteristicParameter>
        <CharacteristicTypeEnum>SURFACEPROFILE</CharacteristicTypeEnum>
        <Parameter>ToleranceValue</Parameter>
      </ArithmeticCharacteristicParameter>
      <ArithmeticConstant val="0.010"/>
    </LessThan>
  </And>
  <ThenPoints>
    <MinPointDensity>0.2</MinPointDensity>
  </ThenPoints>
</IfThenSurfaceRule>
<Else>
  <ThenPoints>
    <MinPoints>3</MinPoints>
  </ThenPoints>
</Else>
</MaxFeatureRules>

```

In this example, we must evaluate all rules to see which evaluate to true. There are 2 rules in this example: the **IfThenSurfaceRule** named “FirstRule” and the **IfThenSurfaceRule** named “SecondRule”. There is also an “Else” statement, to be executed only in the event that no rules apply to the Feature/Characteristic being evaluated.

In many cases, only the “FirstRule” or “SecondRule” will evaluate to true. In this case, the corresponding **ThenPoints** statement should be executed.

However if the Feature/Characteristic happens to be on a surface of greater than 2 units of area AND is a surface profile tolerance with a tolerance value of less than 0.010 units, then both of the rules evaluate true. If this is the case, then the evaluating system must determine which of the rules would apply more points to the surface, and choose that rule.

7 Data dictionary for QIF Rules information model

QIFRules.xsd is a single XML schema file that describes the QIF Rules information model. It defines several data types that are unique to the Rules application area, and, in keeping with QIF design guidelines; it reuses definitions from the QIF library whenever possible. In particular, QIFRules.xsd uses GenericExpression.xsd and Expressions.xsd. For the description of the QIF Rules data dictionary as instantiated using XML Schema, refer to Annex C – Data dictionary for schema file QIFRules.xsd

Annex A – Location of QIFRules.xsd

The QIF Rules information model is expressed in XML schema definition language in the file “QIFRules.xsd”. All QIF XML schema files are normative and are bundled into a single compressed folder file called “QIF_2.0_XMLSchemaFiles.zip”, which is available for download at www.qifstandards.org.

Annex B - Graphical conventions of the data dictionary

(informative)

This section describes the graphical conventions used in the QIF data dictionaries. The data dictionaries describe the structure of the information models and the manufacturing quality semantics of the data types.

The rules of encoding QIF instance files are primarily defined in the XML schema files, but the data dictionaries express many of the same requirements via the pictures and table entries.

Data type definitions are indicated by a box with beveled corners on the left side, as in Figure B.1.

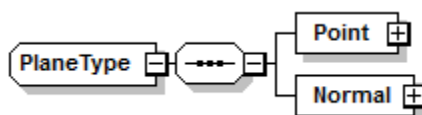


Figure B.1 – Notation for a type definition, *PlaneType*.

Rectangular boxes indicate data *elements*. A solid rectangle indicates a required *element*, whereas a dotted rectangle indicates an optional *element*. If an object is not designated optional, then it is required by default. Small boxes on the right hand end of *element* boxes, containing either "-" or "+" are used to indicate one of the following conditions exist:

- a "+" indicates that the additional structures or *elements* below this node have been hidden in this diagram.
- a "-" indicates that additional structures or *elements* below this node exist and are visible on the diagram.

The absence of any box at the right hand end of an *element* box indicates that the type of the *element* is a *primitive* type without any substructure, e.g., xs:decimal. In this case, there will also be three bars in the upper left corner of the *element* box. The beveled box with 3 dots on a line represents the XSDL *sequence* operator. It indicates that the object to the left is composed of all of the *elements* to the right, in that specified order.

Type definitions can be reused to generate data *elements*, as shown by a yellow box in dotted lines, with the name of the type definition at the top. Figure B.2 shows that **ZonePlane** is an *element* of type ***PlaneType***.

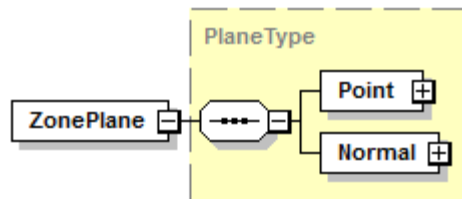


Figure B.2 – Reuse of the type definition *PlaneType* to generate element *ZonePlane*.

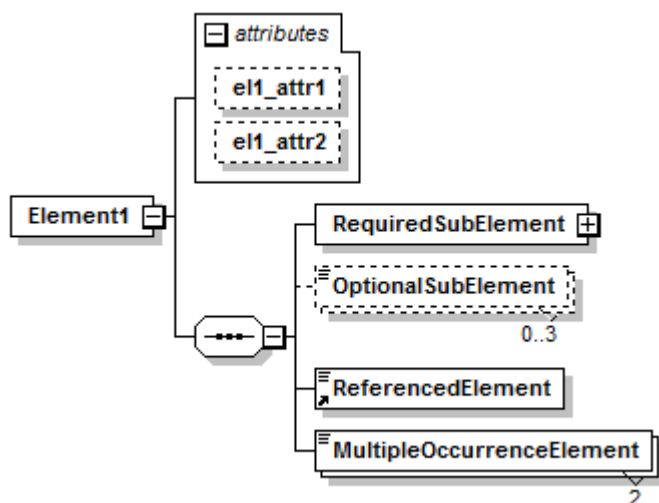


Figure B.3 – Notation for *elements*, *sub-elements*, and *attributes*.

Figure B.3 contains examples of numerous information modeling notations. *Element* definitions in XML schema files can be reused by "reference", indicated by an arrow in the lower left corner of the **ReferencedElement** box. *Elements* may appear in an XML instance document more than once. Figure B.3 shows the **OptionalSubElement** notated with two numerals separated by an ellipsis, "0..3", that indicates the number of occurrences as an inclusive range. The **OptionalSubElement** may occur 0, 1, 2, or 3 times as sub-*elements* of **Element1**. Where there is a single cardinality numeral, the *element* must occur exactly that number of times in the instance file. For example, the *element* **MultipleOccurrenceElement** must occur exactly two times as sub-*elements* of **Element1**. Information items can be instantiated in XSDL as *elements* or *attributes*. An *element's attributes* are shown in the data dictionaries as solid-lined boxes that are explicitly labeled *attributes*, as shown at the top of the diagram.

Figure B.4 shows an example *element* definition where exactly one of the three sub-*element* choices must be given. The beveled box with three square dots and a "switch" line (⎓) indicate the XSDL *choice* structure. When **Element2** is instantiated in an XML instance file, it must have exactly one sub-*element* chosen among the three sub-*elements* shown.

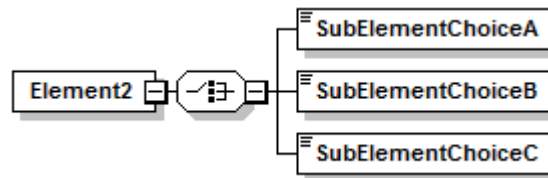


Figure B.4 – The *choice* of notation.

The data dictionaries are grouped by XML schema file. It is characteristic of QIF definitions to use types declared in other XML schema files. The sharing of definitions specified in other files is indicated by the XML schema file directive *include*.

Annex C – Data dictionary for schema file QIFRules.xsd

(normative)

Schema QIFRules.xsd

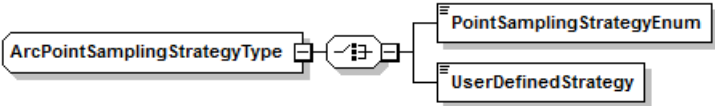
schema location: ..\QIFApplications\QIFRules.xsd
attributeFormDefault: unqualified
elementFormDefault: qualified
targetNamespace: http://qifstandards.org/xsd/qif2

Complex types


[ArcPointSamplingStrategyType](#)
[CirclePointSamplingStrategyType](#)
[ConePointSamplingStrategyType](#)
[ConicalSegmentPointSamplingStrategyType](#)
[CuboidPointSamplingStrategyType](#)
[CylinderPointSamplingStrategyType](#)
[CylindricalSegmentPointSamplingStrategyType](#)
[DMESelectionRulesType](#)
[DMESelectionRuleType](#)
[EllipsePointSamplingStrategyType](#)
[ElongatedCylinderPointSamplingStrategyType](#)
[ElseRuleType](#)
[ExtrudedCrossSectionPointSamplingStrategyType](#)
[FeatureRulesType](#)
[IfThenArcRuleType](#)
[IfThenCircleRuleType](#)
[IfThenConeRuleType](#)
[IfThenConicalSegmentRuleType](#)
[IfThenCuboidRuleType](#)
[IfThenCurveRuleType](#)
[IfThenCylinderRuleType](#)
[IfThenCylindricalSegmentRuleType](#)
[IfThenEllipseRuleType](#)
[IfThenElongatedCylinderRuleType](#)
[IfThenElseFeatureRulesType](#)
[IfThenExtrudedCrossSectionRuleType](#)
[IfThenFeatureRuleType](#)
[IfThenLineRuleType](#)
[IfThenOppositeLinesRuleType](#)
[IfThenOppositePlanesRuleType](#)
[IfThenPlaneRuleType](#)
[IfThenPointDefinedCurveRuleType](#)
[IfThenPointDefinedSurfaceRuleType](#)
[IfThenPointRuleType](#)
[IfThenSphereRuleType](#)
[IfThenSphericalSegmentRuleType](#)
[IfThenSurfaceOfRevolutionRuleType](#)
[IfThenSurfaceRuleType](#)
[IfThenToroidalSegmentRuleType](#)
[IfThenTorusRuleType](#)
[LinePointSamplingStrategyType](#)

[MaxFeatureRulesType](#)
[OppositeLinesPointSamplingStrategyType](#)
[OppositePlanesPointSamplingStrategyType](#)
[PlanePointSamplingStrategyType](#)
[PointDefinedCurvePointSamplingStrategyType](#)
[PointDefinedSurfacePointSamplingStrategyType](#)
[PointPointSamplingStrategyType](#)
[PointRuleBaseType](#)
[QIFRuleBaseType](#)
[QIFRulesType](#)
[SpherePointSamplingStrategyType](#)
[SphericalSegmentPointSamplingStrategyType](#)
[SurfaceOfRevolutionPointSamplingStrategyType](#)
[ThenPointsType](#)
[ToroidalSegmentPointSamplingStrategyType](#)
[TorusPointSamplingStrategyType](#)


complexType **ArcPointSamplingStrategyType**

diagram	
children	PointSamplingStrategyEnum UserDefinedStrategy
used by	element IfThenArcRuleType/ThenPointStrategy
annotation	documentation The ArcPointSamplingStrategyType defines a point sampling strategy for an arc. The strategy is either an enumerated arc strategy or a user defined strategy.

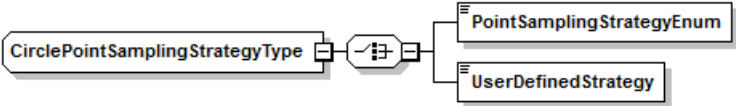
element **ArcPointSamplingStrategyType/PointSamplingStrategyEnum**

diagram							
type	OpenCurvePointSamplingStrategyEnumType						
properties	content simple						
facets	<table><tr><th>Kind</th><th>Value</th><th>Annotation</th></tr><tr><td>enumeration</td><td>POINTS</td><td></td></tr></table>	Kind	Value	Annotation	enumeration	POINTS	
Kind	Value	Annotation					
enumeration	POINTS						
annotation	<p>documentation</p> <p>The PointSamplingStrategyEnum element is a commonly used point sampling strategy for an arc.</p>						

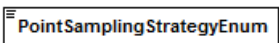
element **ArcPointSamplingStrategyType/UserDefinedStrategy**

diagram	
type	xs:token
properties	content simple
annotation	documentation The UserDefinedStrategy element is the name of a point sampling strategy defined by the user.


complexType CirclePointSamplingStrategyType

diagram	
children	PointSamplingStrategyEnum UserDefinedStrategy
used by	element IfThenCircleRuleType/ThenPointStrategy
annotation	documentation The CirclePointSamplingStrategyType defines a point sampling strategy for a circle. The strategy is either an enumerated circle strategy or a user defined strategy.

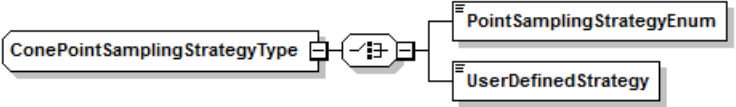
element CirclePointSamplingStrategyType/PointSamplingStrategyEnum

diagram										
type	ClosedCurvePointSamplingStrategyEnumType									
properties	content simple									
facets	<table><tr><td>Kind</td><td>Value</td><td>Annotation</td></tr><tr><td>enumeration</td><td>POINTS</td><td></td></tr><tr><td>enumeration</td><td>EQUIDISTANT</td><td></td></tr></table>	Kind	Value	Annotation	enumeration	POINTS		enumeration	EQUIDISTANT	
Kind	Value	Annotation								
enumeration	POINTS									
enumeration	EQUIDISTANT									
annotation	<p>documentation</p> <p>The PointSamplingStrategyEnum element is a commonly used point sampling strategy for a circle.</p>									


element CirclePointSamplingStrategyType/UserDefinedStrategy

diagram	
type	xs:token
properties	content simple
annotation	documentation The UserDefinedStrategy element is the name of a point sampling strategy defined by the user.


complexType ConePointSamplingStrategyType

diagram	
children	PointSamplingStrategyEnum UserDefinedStrategy
used by	element IfThenConeRuleType/ThenPointStrategy
annotation	documentation The ConePointSamplingStrategyType defines a point sampling strategy for a cone. The strategy is either an enumerated cone strategy or a user defined strategy.

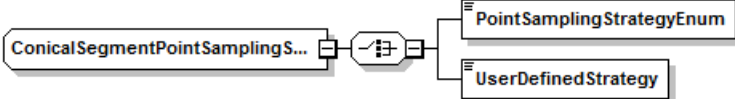
element **ConePointSamplingStrategyType/PointSamplingStrategyEnum**

diagram			
type	ConePointSamplingStrategyEnumType		
properties	content	simple	
facets	Kind	Value	Annotation
	enumeration	POLARGRID	
	enumeration	STRATIFIED	
	enumeration	SPIRAL	
	enumeration	SPIDERWEB	
	enumeration	POINTS	
annotation	documentation The PointSamplingStrategyEnum element is a commonly used point sampling strategy for a cone.		

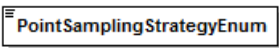
element **ConePointSamplingStrategyType/UserDefinedStrategy**

diagram	
type	xs:token
properties	content simple
annotation	<div>documentation</div> <div>The UserDefinedStrategy element is the name of a point sampling strategy defined by the user.</div>

complexType **ConicalSegmentPointSamplingStrategyType**


diagram	
children	PointSamplingStrategyEnum UserDefinedStrategy
used by	element IfThenConicalSegmentRuleType/ThenPointStrategy
annotation	documentation The ConicalSegmentPointSamplingStrategyType defines a point sampling strategy for a conical segment. The strategy is either an enumerated conical segment strategy or a user defined strategy.

element **ConicalSegmentPointSamplingStrategyType/PointSamplingStrategyEnum**

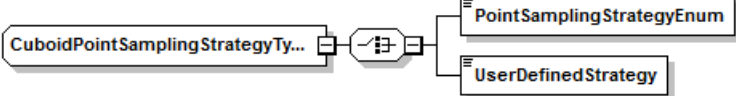
diagram			
type	ConePointSamplingStrategyEnumType		
properties	content	simple	
facets	Kind	Value	Annotation
	enumeration	POLARGRID	
	enumeration	STRATIFIED	
	enumeration	SPIRAL	
	enumeration	SPIDERWEB	
	enumeration	POINTS	

annotation	documentation The PointSamplingStrategyEnum element is a commonly used point sampling strategy for a conical segment.
------------	--


element **ConicalSegmentPointSamplingStrategyType/UserDefinedStrategy**

diagram	
type	xs:token
properties	content simple
annotation	documentation The UserDefinedStrategy element is the name of a point sampling strategy defined by the user.


complexType **CuboidPointSamplingStrategyType**

diagram	
children	PointSamplingStrategyEnum UserDefinedStrategy
used by	element IfThenCuboidRuleType/ThenPointStrategy
annotation	documentation The CuboidPointSamplingStrategyType defines a point sampling strategy for a cuboid. The strategy is either an enumerated cuboid strategy or a user defined strategy.

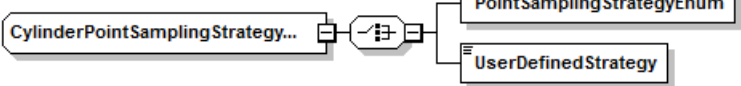
element **CuboidPointSamplingStrategyType/PointSamplingStrategyEnum**

diagram																						
type	PrismPointSamplingStrategyEnumType																					
properties	content simple																					
facets	<table><tr><th>Kind</th><th>Value</th><th>Annotation</th></tr><tr><td>enumeration</td><td>ORTHOGONALGRID</td><td></td></tr><tr><td>enumeration</td><td>BIRDCAGE</td><td></td></tr><tr><td>enumeration</td><td>SPECIFIEDGRID</td><td></td></tr><tr><td>enumeration</td><td>STRATIFIED</td><td></td></tr><tr><td>enumeration</td><td>HELIX</td><td></td></tr><tr><td>enumeration</td><td>POINTS</td><td></td></tr></table>	Kind	Value	Annotation	enumeration	ORTHOGONALGRID		enumeration	BIRDCAGE		enumeration	SPECIFIEDGRID		enumeration	STRATIFIED		enumeration	HELIX		enumeration	POINTS	
Kind	Value	Annotation																				
enumeration	ORTHOGONALGRID																					
enumeration	BIRDCAGE																					
enumeration	SPECIFIEDGRID																					
enumeration	STRATIFIED																					
enumeration	HELIX																					
enumeration	POINTS																					
annotation	<div>documentation</div> <div>The PointSamplingStrategyEnum element is a commonly used point sampling strategy for a cuboid.</div>																					

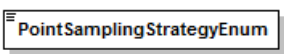
element **CuboidPointSamplingStrategyType/UserDefinedStrategy**

diagram	
type	xs:token
properties	content simple
annotation	documentation The UserDefinedStrategy element is the name of a point sampling strategy defined by the user.


complexType **CylinderPointSamplingStrategyType**

diagram	
children	PointSamplingStrategyEnum UserDefinedStrategy
used by	element IfThenCylinderRuleType/ThenPointStrategy
annotation	documentation The CylinderPointSamplingStrategyType defines a point sampling strategy for a cylinder. The strategy is either an enumerated cylinder strategy or a user defined strategy.

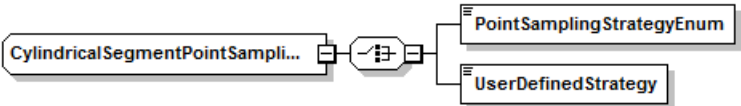
element **CylinderPointSamplingStrategyType/PointSamplingStrategyEnum**

diagram																						
type	SurfaceOfRevolutionPointSamplingStrategyEnumType																					
properties	content simple																					
facets	<table><tr><th>Kind</th><th>Value</th><th>Annotation</th></tr><tr><td>enumeration</td><td>ORTHOGONALGRID</td><td></td></tr><tr><td>enumeration</td><td>BIRDCAGE</td><td></td></tr><tr><td>enumeration</td><td>SPECIFIEDGRID</td><td></td></tr><tr><td>enumeration</td><td>STRATIFIED</td><td></td></tr><tr><td>enumeration</td><td>HELIX</td><td></td></tr><tr><td>enumeration</td><td>POINTS</td><td></td></tr></table>	Kind	Value	Annotation	enumeration	ORTHOGONALGRID		enumeration	BIRDCAGE		enumeration	SPECIFIEDGRID		enumeration	STRATIFIED		enumeration	HELIX		enumeration	POINTS	
Kind	Value	Annotation																				
enumeration	ORTHOGONALGRID																					
enumeration	BIRDCAGE																					
enumeration	SPECIFIEDGRID																					
enumeration	STRATIFIED																					
enumeration	HELIX																					
enumeration	POINTS																					
annotation	<div>documentation</div> <div>The PointSamplingStrategyEnum element is a commonly used point sampling strategy for a cylinder.</div>																					

element **CylinderPointSamplingStrategyType/UserDefinedStrategy**


diagram	
type	xs:token
properties	content simple
annotation	documentation The UserDefinedStrategy element is the name of a point sampling strategy defined by the user.

complexType **CylindricalSegmentPointSamplingStrategyType**


diagram	
children	PointSamplingStrategyEnum UserDefinedStrategy
used by	element IfThenCylindricalSegmentRuleType/ThenPointStrategy
annotation	documentation The CylindricalSegmentPointSamplingStrategyType defines a point sampling strategy for a cylindrical segment.

	The strategy is either an enumerated cylindrical segment strategy or a user defined strategy.
--	---

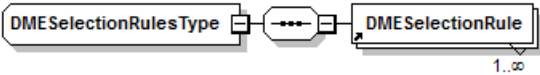
element CylindricalSegmentPointSamplingStrategyType/PointSamplingStrategyEnum

diagram			
type	SurfaceOfRevolutionPointSamplingStrategyEnumType		
properties	content	simple	
facets	Kind	Value	Annotation
	enumeration	ORTHOGONALGRID	
	enumeration	BIRDCAGE	
	enumeration	SPECIFIEDGRID	
	enumeration	STRATIFIED	
	enumeration	HELIX	
	enumeration	POINTS	
annotation	documentation	The PointSamplingStrategyEnum element is a commonly used point sampling strategy for a cylindrical segment.	


element CylindricalSegmentPointSamplingStrategyType/UserDefinedStrategy

diagram			
type	xs:token		
properties	content	simple	
annotation	documentation	The UserDefinedStrategy element is the name of a point sampling strategy defined by the user.	

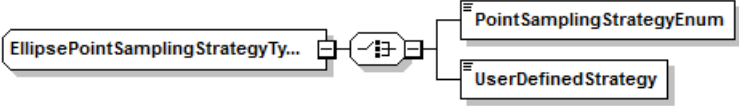
complexType DMESelectionRulesType

diagram			
children	DMESelectionRule		
used by	element	QIFRulesType/DMESelectionRules	
annotation	documentation	The DMESelectionRulesType defines a stub for a set of rules for selecting DMEs.	


complexType DMESelectionRuleType

diagram			
used by	element	DMESelectionRule	
annotation	documentation	The DMESelectionRuleType defines a stub for a rule for selecting a DME.	


complexType **EllipsePointSamplingStrategyType**

diagram	
children	PointSamplingStrategyEnum UserDefinedStrategy
used by	element IfThenEllipseRuleType/ThenPointStrategy
annotation	documentation The EllipsePointSamplingStrategyType defines a point sampling strategy for an ellipse. The strategy is either an enumerated ellipse strategy or a user defined strategy.

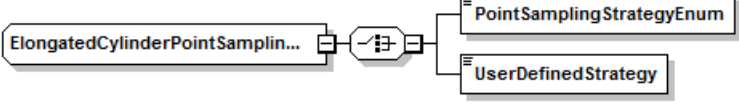
element **EllipsePointSamplingStrategyType/PointSamplingStrategyEnum**

diagram			
type	ClosedCurvePointSamplingStrategyEnumType		
properties	content	simple	
facets	Kind enumeration	Value POINTS enumeration	Annotation EQUIDISTANT
annotation	documentation The PointSamplingStrategyEnum element is a commonly used point sampling strategy for an ellipse.		


element **EllipsePointSamplingStrategyType/UserDefinedStrategy**

diagram	
type	xs:token
properties	content simple
annotation	documentation The UserDefinedStrategy element is the name of a point sampling strategy defined by the user.

complexType **ElongatedCylinderPointSamplingStrategyType**


diagram	
children	PointSamplingStrategyEnum UserDefinedStrategy
used by	element IfThenElongatedCylinderRuleType/ThenPointStrategy
annotation	documentation The ElongatedCylinderPointSamplingStrategyType defines a point sampling strategy for an elongated cylinder. The strategy is either an enumerated elongated cylinder strategy or a user defined strategy.

element **ElongatedCylinderPointSamplingStrategyType/PointSamplingStrategyEnum**

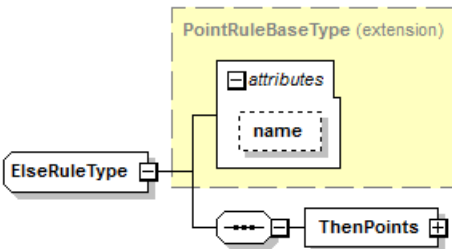
diagram	
---------	---

type	ElongatedCylinderPointSamplingStrategyEnumType		
properties	content	simple	
facets	Kind	Value	Annotation
	enumeration	ORTHOGONALGRID	
	enumeration	BIRDCAGE	
	enumeration	SPECIFIEDGRID	
	enumeration	STRATIFIED	
	enumeration	HELIX	
annotation	enumeration	POINTS	
	documentation	The PointSamplingStrategyEnum element is a commonly used point sampling strategy for an elongated cylinder.	

element **ElongatedCylinderPointSamplingStrategyType/UserDefinedStrategy**

diagram					
type	xs:token				
properties	content	simple			
annotation	documentation	The UserDefinedStrategy element is the name of a point sampling strategy defined by the user.			

complexType **ElseRuleType**

diagram													
type	extension of PointRuleBaseType												
properties	base PointRuleBaseType												
children	ThenPoints												
used by	elements IfThenElseFeatureRulesType/Else MaxFeatureRulesType/Else												
attributes	<table><tr><td>Name</td><td>Type</td><td>Use</td><td>Default</td><td>Fixed</td><td>Annotation</td></tr><tr><td>name</td><td>xs:token</td><td></td><td></td><td></td><td>documentation The optional name attribute is the name of the rule.</td></tr></table>	Name	Type	Use	Default	Fixed	Annotation	name	xs:token				documentation The optional name attribute is the name of the rule.
Name	Type	Use	Default	Fixed	Annotation								
name	xs:token				documentation The optional name attribute is the name of the rule.								
annotation	<p>documentation</p> <p>The ElseRuleType defines a rule containing an optional 'then points' part which always applies.</p>												


element **ElseRuleType/ThenPoints**

diagram	
type	ThenPointsType
properties	content complex
children	NumberOfPoints MinPoints PointDensity MinPointDensity
annotation	documentation The ThenPoints element gives (only) the quantity of points to use.

complexType **ExtrudedCrossSectionPointSamplingStrategyType**

diagram	
children	PointSamplingStrategyEnum UserDefinedStrategy
used by	element IfThenExtrudedCrossSectionRuleType/ThenPointStrategy
annotation	documentation The ExtrudedCrossSectionPointSamplingStrategyType defines a point sampling strategy for an extruded cross-section. The strategy is either an enumerated extruded cross-section strategy or a user defined strategy.

element **ExtrudedCrossSectionPointSamplingStrategyType/PointSamplingStrategyEnum**

diagram													
type	ExtrudedCrossSectionPointSamplingStrategyEnumType												
properties	content simple												
facets	<table><tr><th>Kind</th><th>Value</th><th>Annotation</th></tr><tr><td>enumeration</td><td>BIRDCAGE</td><td></td></tr><tr><td>enumeration</td><td>STRATIFIED</td><td></td></tr><tr><td>enumeration</td><td>POINTS</td><td></td></tr></table>	Kind	Value	Annotation	enumeration	BIRDCAGE		enumeration	STRATIFIED		enumeration	POINTS	
Kind	Value	Annotation											
enumeration	BIRDCAGE												
enumeration	STRATIFIED												
enumeration	POINTS												
annotation	<p>documentation</p> <p>The PointSamplingStrategyEnum element is a commonly used point sampling strategy for an extruded cross-section</p>												

element **ExtrudedCrossSectionPointSamplingStrategyType/UserDefinedStrategy**

diagram	
type	xs:token

properties	content simple
annotation	documentation The UserDefinedStrategy element is the name of a point sampling strategy defined by the user.

complexType FeatureRulesType

diagram	
children	SamplingRigorMax IfThenElseFeatureRules MaxFeatureRules
used by	element QIFRulesType/FeatureRules
annotation	<p>documentation</p> <p>The FeatureRulesType defines two methods of setting the quantity of hit points to use for a feature and/or a strategy to use for selecting the hit points and/or the substitute feature algorithm to use to fit the feature to the hit points. If the IfThenElseFeatureRules and MaxFeatureRules are both used to find the quantity of points for a feature, both sets of rules should be applied and the maximum value should be used. Also if both are used and they give different point sampling strategies or different substitute feature algorithms, the evaluating system may use any of those strategies or algorithms. Since a given feature F may be associated with zero to many characteristics C1 ... Cn, if there is any characteristic associated with F, the rules should be evaluated for each FCi pair, and the number or density of hit points to use for F should be set to the maximum value. If a number of hit points must be compared with a density of hit points, the evaluating system should find and use the area of the feature to convert density values to numbers of points. The rules are to be evaluated in an environment in which the SamplingRigor has been set to some positive integer less than or equal to the SamplingRigorMax element so that the SamplingRigorIsType Boolean expression can be evaluated. SamplingRigor values start at 1 and end at the value of the SamplingRigorMax element. For example, if SamplingRigorMax is 2, the values 1 and 2 might represent LOW and HIGH, respectively. If the SamplingRigorMax is 4, the values 1 through 4 might represent LOW, MEDIUM, HIGH, and SUPER, respectively. Numerical quantities that are constants, feature parameters, or characteristic parameters may be used in conditions. The numerical quantities FeatureLength and FeatureArea may also be used in conditions. The evaluating system must set the value of FeatureLength for the feature being processed if FeatureLength is used in any condition. The evaluating system must set the value of FeatureArea for the feature being processed if FeatureArea is used in any condition. The rules are not intended for picking the quantity of points or point sampling strategy to use for measuring a characteristic that is not associated with any feature.</p>

element FeatureRulesType/SamplingRigorMax

diagram	
type	xs:unsignedInt
properties	content simple
annotation	<p>documentation</p> <p>The SamplingRigorMax element is the inclusive upper limit on point levels.</p>

element **FeatureRulesType/IfThenElseFeatureRules**

diagram	
type	IfThenElseFeatureRulesType
properties	minOcc 0 maxOcc 1 content complex
children	IfThenFeatureRule Else
annotation	documentation The optional IfThenElseFeatureRules element is a set of rules for selecting the number or density of hit points for a feature and/or a strategy to use for selecting the hit points and/or a substitute feature algorithm. The first rule that applies is used to set the quantity of hit points and/or the strategy and/or the algorithm.

element **FeatureRulesType/MaxFeatureRules**

diagram	
type	MaxFeatureRulesType
properties	minOcc 0 maxOcc 1 content complex
children	IfThenFeatureRule Else
annotation	documentation The optional MaxFeatureRules element is a set of rules for selecting the number or density of hit points for a feature and/or a strategy to use for selecting the hit points and/or a substitute feature algorithm. The largest value returned by all rules that apply is used to set the quantity of hit points. If the rules that apply give different point sampling strategies or substitute feature algorithms, the evaluating system may use any of those strategies or algorithms.

complexType IfThenArcRuleType

diagram	<p>The diagram shows the structure of the <code>IfThenArcRuleType</code> complex type. It is an extension of <code>IfThenFeatureRuleType</code>. It includes an optional attribute <code>name</code>. The main body of the type is a choice of four elements: <code>BooleanExpression</code>, <code>ThenPoints</code>, <code>ThenPointStrategy</code>, and <code>ThenFittingAlgorithm</code>. The <code>BooleanExpression</code> element is further detailed as a choice of various logical and feature-based expressions: <code>And</code>, <code>ArithmeticEqual</code>, <code>BooleanEqual</code>, <code>Characteristics</code>, <code>Constants</code>, <code>FeatureIsDatum</code>, <code>FeatureIsInternal</code>, <code>GreaterOrEqual</code>, <code>GreaterThan</code>, <code>LessOrEqual</code>, <code>LessThan</code>, <code>Not</code>, <code>Or</code>, <code>SamplingRigorIs</code>, <code>ShapeClasses</code>, and <code>ThenPoints</code>.</p>					
type	extension of IfThenFeatureRuleType					
properties	base <code>IfThenFeatureRuleType</code>					
children	BooleanExpression ThenPoints ThenPointStrategy ThenFittingAlgorithm					
used by	element IfThenArcRule					
attributes	Name name	Type xs:token	Use	Default	Fixed	Annotation documentation The optional name attribute is the name of

	the rule.
annotation	<p>documentation</p> <p>The IfThenArcRuleType defines a conditional rule that applies if the feature under consideration is an arc. The rule contains an inherited 'if' part, an inherited ThenPoints part, and two more 'then' parts. The 'then' parts of the rule apply if the 'if' part does not exist or evaluates to true.</p>

element IfThenArcRuleType/ThenPointStrategy

diagram	
type	ArcPointSamplingStrategyType
properties	<p>minOcc 0</p> <p>maxOcc 1</p> <p>content complex</p>
children	PointSamplingStrategyEnum UserDefinedStrategy
annotation	<p>documentation</p> <p>The optional ThenPointStrategy element is the point sampling strategy to be used if the 'if' part of the rule does not exist or evaluates to true.</p>

element IfThenArcRuleType/ThenFittingAlgorithm

diagram	
type	NonFeatureOfSizeSubstituteFeatureAlgorithmType
properties	<p>minOcc 0</p> <p>maxOcc 1</p> <p>content complex</p>
children	NonFeatureOfSizeSubstituteFeatureAlgorithmEnum OtherNonFeatureOfSizeSubstituteFeatureAlgorithm
annotation	<p>documentation</p> <p>The optional ThenFittingAlgorithm element is the substitute feature algorithm to be used if the 'if' part of the rule does not exist or evaluates to true.</p>

complexType **IfThenCircleRuleType**

diagram	<p>The diagram shows the structure of IfThenCircleRuleType. It is a complex type that extends IfThenFeatureRuleType. It has an attribute name of type xs:string. The main body of the type is a sequence of elements: BooleanExpression, And, ArithmeticEqual, BooleanEqual, Characteristics, ConstantIs, FeatureIsDatum, FeatureIsInternal, GreaterOrEqual, GreaterThan, LessOrEqual, LessThan, Not, Or, SamplingRigors, ShapeClassIs, and ThenPoints. Additionally, it has two optional elements: ThenPointStrategy and ThenFittingAlgorithm.</p>					
type	extension of IfThenFeatureRuleType					
properties	base IfThenFeatureRuleType					
children	BooleanExpression ThenPoints ThenPointStrategy ThenFittingAlgorithm					
used by	element IfThenCircleRule					
attributes	Name name	Type xs:string	Use	Default	Fixed	Annotation documentation The optional name attribute is the name of the rule.
annotation	documentation The IfThenCircleRuleType defines a conditional rule that applies if the feature under consideration is a circle. The					

	rule contains an inherited 'if' part, an inherited ThenPoints part, and two more 'then' parts. The 'then' parts of the rule apply if the 'if' part does not exist or evaluates to true.
--	---

element **IfThenCircleRuleType/ThenPointStrategy**

diagram	
type	CirclePointSamplingStrategyType
properties	minOcc 0 maxOcc 1 content complex
children	PointSamplingStrategyEnum UserDefinedStrategy
annotation	documentation The optional ThenPointStrategy element is the point sampling strategy to be used if the 'if' part of the rule does not exist or evaluates to true.

element **IfThenCircleRuleType/ThenFittingAlgorithm**

diagram	
type	FeatureOfSizeSubstituteFeatureAlgorithmType
properties	minOcc 0 maxOcc 1 content complex
children	FeatureOfSizeSubstituteFeatureAlgorithmEnum OtherFeatureOfSizeSubstituteFeatureAlgorithm
annotation	documentation The optional ThenFittingAlgorithm element is the substitute feature algorithm to be used if the 'if' part of the rule does not exist or does not exist or evaluates to true.

complexType **IfThenConeRuleType**

diagram	<p>The diagram shows the structure of IfThenConeRuleType. It is a complex type that extends IfThenFeatureRuleType. It has an optional attribute name. The body contains a choice between a BooleanExpression (which is further broken down into And, ArithmeticEqual, BooleanEqual, Characteristics, ConstantsIs, FeatureIsDatum, FeatureIsInternal, GreaterOrEqual, GreaterThan, LessOrEqual, LessThan, Not, Or, SamplingRigoris, ShapeClassIs, and ThenPoints) and a choice between ThenPointStrategy and ThenFittingAlgorithm.</p>					
type	extension of IfThenFeatureRuleType					
properties	base IfThenFeatureRuleType					
children	BooleanExpression ThenPoints ThenPointStrategy ThenFittingAlgorithm					
used by	element IfThenConeRule					
attributes	Name name	Type xs:token	Use	Default	Fixed	Annotation documentation The optional name attribute is the name of the rule.
annotation	documentation The IfThenConeRuleType defines a conditional rule that applies if the feature under consideration is a cone. The					

	rule contains an inherited 'if' part, an inherited ThenPoints part, and two more 'then' parts. The 'then' parts of the rule apply if the 'if' part does not exist or evaluates to true.
--	---

element IfThenConeRuleType/ThenPointStrategy

diagram	
type	ConePointSamplingStrategyType
properties	minOcc 0 maxOcc 1 content complex
children	PointSamplingStrategyEnum UserDefinedStrategy
annotation	documentation The optional ThenPointStrategy element is the point sampling strategy to be used if the 'if' part of the rule does not exist or evaluates to true.

element IfThenConeRuleType/ThenFittingAlgorithm

diagram	
type	FeatureOfSizeSubstituteFeatureAlgorithmType
properties	minOcc 0 maxOcc 1 content complex
children	FeatureOfSizeSubstituteFeatureAlgorithmEnum OtherFeatureOfSizeSubstituteFeatureAlgorithm
annotation	documentation The optional ThenFittingAlgorithm element is the substitute feature algorithm to be used if the 'if' part of the rule does not exist or evaluates to true.

complexType **IfThenConicalSegmentRuleType**

diagram						
type	extension of IfThenFeatureRuleType					
properties	base IfThenFeatureRuleType					
children	BooleanExpression ThenPoints ThenPointStrategy ThenFittingAlgorithm					
used by	element IfThenConicalSegmentRule					
attributes	Name name	Type xs:token	Use	Default	Fixed	Annotation documentation The optional name attribute is the name of the rule.
annotation	documentation The IfThenConicalSegmentRuleType defines a conditional rule that applies if the feature under consideration is a					

	conical segment. The rule contains an inherited 'if' part, an inherited ThenPoints part, and two more 'then' parts. The 'then' parts of the rule apply if the 'if' part does not exist or evaluates to true.
--	--

element **IfThenConicalSegmentRuleType/ThenPointStrategy**

diagram	
type	ConicalSegmentPointSamplingStrategyType
properties	minOcc 0 maxOcc 1 content complex
children	PointSamplingStrategyEnum UserDefinedStrategy
annotation	documentation The optional ThenPointStrategy element is the point sampling strategy to be used if the 'if' part of the rule does not exist or evaluates to true.

element **IfThenConicalSegmentRuleType/ThenFittingAlgorithm**

diagram	
type	NonFeatureOfSizeSubstituteFeatureAlgorithmType
properties	minOcc 0 maxOcc 1 content complex
children	NonFeatureOfSizeSubstituteFeatureAlgorithmEnum OtherNonFeatureOfSizeSubstituteFeatureAlgorithm
annotation	documentation The optional ThenFittingAlgorithm element is the substitute feature algorithm to be used if the 'if' part of the rule does not exist or evaluates to true.

complexType **IfThenCuboidRuleType**

diagram	<p>The diagram shows the structure of IfThenCuboidRuleType. It is a complex type that extends IfThenFeatureRuleType. It has an optional attribute name. The content model is a sequence of elements: BooleanExpression (required), followed by an optional choice of And, ArithmeticEqual, BooleanEqual, Characteristics, ConstantIs, FeatureIsDatum, FeatureIsInternal, GreaterOrEqual, GreaterThan, LessOrEqual, LessThan, Not, Or, SamplingRigoris, ShapeClassIs, and ThenPoints. This is followed by another optional choice of ThenPointStrategy and ThenFittingAlgorithm.</p>					
type	extension of IfThenFeatureRuleType					
properties	base IfThenFeatureRuleType					
children	BooleanExpression ThenPoints ThenPointStrategy ThenFittingAlgorithm					
used by	element IfThenCuboidRule					
attributes	Name name	Type xs:token	Use	Default	Fixed	Annotation documentation The optional name attribute is the name of the rule.
annotation	documentation The IfThenCuboidRuleType defines a conditional rule that applies if the feature under consideration is a cuboid.					

	The rule contains an inherited 'if' part, an inherited ThenPoints part, and two more 'then' parts. The 'then' parts of the rule apply if the 'if' part does not exist or evaluates to true.
--	---

element **IfThenCuboidRuleType/ThenPointStrategy**

diagram	
type	CuboidPointSamplingStrategyType
properties	minOcc 0 maxOcc 1 content complex
children	PointSamplingStrategyEnum UserDefinedStrategy
annotation	documentation The optional ThenPointStrategy element is the point sampling strategy to be used if the 'if' part of the rule does not exist or evaluates to true.

element **IfThenCuboidRuleType/ThenFittingAlgorithm**

diagram	
type	FeatureOfSizeSubstituteFeatureAlgorithmType
properties	minOcc 0 maxOcc 1 content complex
children	FeatureOfSizeSubstituteFeatureAlgorithmEnum OtherFeatureOfSizeSubstituteFeatureAlgorithm
annotation	documentation The optional ThenFittingAlgorithm element is the substitute feature algorithm to be used if the 'if' part of the rule does not exist or evaluates to true.

complexType **IfThenCurveRuleType**

diagram						
type	extension of IfThenFeatureRuleType					
properties	base IfThenFeatureRuleType					
children	BooleanExpression ThenPoints					
used by	element IfThenCurveRule					
attributes	Name name	Type xs:token	Use	Default	Fixed	Annotation documentation The optional name attribute is the name of the rule.
annotation	documentation The IfThenCurveRuleType defines a conditional rule that applies if the feature under consideration is a curve. The rule contains an inherited 'if' part and an inherited ThenPoints part. The 'then' part of the rule applies if the 'if' part does not exist or evaluates to true.					

complexType **IfThenCylinderRuleType**

diagram						
type	extension of IfThenFeatureRuleType					
properties	base IfThenFeatureRuleType					
children	BooleanExpression ThenPoints ThenPointStrategy ThenFittingAlgorithm					
used by	element IfThenCylinderRule					
attributes	Name name	Type xs:token	Use	Default	Fixed	Annotation documentation The optional name attribute is the name of the rule.
annotation	documentation The IfThenCylinderRuleType defines a conditional rule that applies if the feature under consideration is a cylinder.					

	The rule contains an inherited 'if' part, an inherited ThenPoints part, and two more 'then' parts. The 'then' parts of the rule apply if the 'if' part does not exist or evaluates to true.
--	---

element IfThenCylinderRuleType/ThenPointStrategy

diagram	
type	CylinderPointSamplingStrategyType
properties	minOcc 0 maxOcc 1 content complex
children	PointSamplingStrategyEnum UserDefinedStrategy
annotation	documentation The optional ThenPointStrategy element is the point sampling strategy to be used if the 'if' part of the rule does not exist or evaluates to true.

element IfThenCylinderRuleType/ThenFittingAlgorithm

diagram	
type	FeatureOfSizeSubstituteFeatureAlgorithmType
properties	minOcc 0 maxOcc 1 content complex
children	FeatureOfSizeSubstituteFeatureAlgorithmEnum OtherFeatureOfSizeSubstituteFeatureAlgorithm
annotation	documentation The optional ThenFittingAlgorithm element is the substitute feature algorithm to be used if the 'if' part of the rule does not exist or evaluates to true.

complexType **IfThenCylindricalSegmentRuleType**

diagram						
type	extension of IfThenFeatureRuleType					
properties	base IfThenFeatureRuleType					
children	BooleanExpression ThenPoints ThenPointStrategy ThenFittingAlgorithm					
used by	element IfThenCylindricalSegmentRule					
attributes	Name name	Type xs:token	Use	Default	Fixed	Annotation documentation The optional name attribute is the name of the rule.
annotation	documentation The IfThenCylindricalSegmentRuleType defines a conditional rule that applies if the feature under consideration is					

	a cylindrical segment. The rule contains an inherited 'if' part, an inherited ThenPoints part, and two more 'then' parts. The 'then' parts of the rule apply if the 'if' part evaluates to true.
--	--

element **IfThenCylindricalSegmentRuleType/ThenPointStrategy**

diagram	
type	CylindricalSegmentPointSamplingStrategyType
properties	minOcc 0 maxOcc 1 content complex
children	PointSamplingStrategyEnum UserDefinedStrategy
annotation	documentation The optional ThenPointStrategy element is the point sampling strategy to be used if the 'if' part of the rule does not exist or evaluates to true.

element **IfThenCylindricalSegmentRuleType/ThenFittingAlgorithm**

diagram	
type	NonFeatureOfSizeSubstituteFeatureAlgorithmType
properties	minOcc 0 maxOcc 1 content complex
children	NonFeatureOfSizeSubstituteFeatureAlgorithmEnum OtherNonFeatureOfSizeSubstituteFeatureAlgorithm
annotation	documentation The optional ThenFittingAlgorithm element is the substitute feature algorithm to be used if the 'if' part of the rule does not exist or evaluates to true.

complexType **IfThenEllipseRuleType**

diagram						
type	extension of IfThenFeatureRuleType					
properties	base IfThenFeatureRuleType					
children	BooleanExpression ThenPoints ThenPointStrategy ThenFittingAlgorithm					
used by	element IfThenEllipseRule					
attributes	Name name	Type xs:token	Use	Default	Fixed	Annotation documentation The optional name attribute is the name of the rule.
annotation	documentation The IfThenEllipseRuleType defines a conditional rule that applies if the feature under consideration is an ellipse.					

	The rule contains an inherited 'if' part, an inherited ThenPoints part, and two more 'then' parts. The 'then' parts of the rule apply if the 'if' part does not exist or evaluates to true.
--	---

element IfThenEllipseRuleType/ThenPointStrategy

diagram	
type	EllipsePointSamplingStrategyType
properties	minOcc 0 maxOcc 1 content complex
children	PointSamplingStrategyEnum UserDefinedStrategy
annotation	documentation The optional ThenPointStrategy element is the point sampling strategy to be used if the 'if' part of the rule does not exist or evaluates to true.

element IfThenEllipseRuleType/ThenFittingAlgorithm

diagram	
type	FeatureOfSizeSubstituteFeatureAlgorithmType
properties	minOcc 0 maxOcc 1 content complex
children	FeatureOfSizeSubstituteFeatureAlgorithmEnum OtherFeatureOfSizeSubstituteFeatureAlgorithm
annotation	documentation The optional ThenFittingAlgorithm element is the substitute feature algorithm to be used if the 'if' part of the rule does not exist or evaluates to true.

complexType **IfThenElongatedCylinderRuleType**

diagram	<p>The diagram shows the structure of IfThenElongatedCylinderRuleType. It is a complex type that extends IfThenFeatureRuleType. It includes an optional attribute name. The main body of the type is a sequence of elements: BooleanExpression, And, ArithmeticEqual, BooleanEqual, Characteristics, ConstantIs, FeatureIsDatum, FeatureIsInternal, GreaterOrEqual, GreaterThan, LessOrEqual, LessThan, Not, Or, SamplingRigorIs, ShapeClassIs, and ThenPoints. There are two optional groups: one containing GreaterOrEqual, GreaterThan, LessOrEqual, and LessThan; another containing ThenPointStrategy and ThenFittingAlgorithm.</p>					
type	extension of IfThenFeatureRuleType					
properties	base IfThenFeatureRuleType					
children	BooleanExpression ThenPoints ThenPointStrategy ThenFittingAlgorithm					
used by	element IfThenElongatedCylinderRule					
attributes	Name name	Type xs:token	Use	Default	Fixed	Annotation documentation The optional name attribute is the name of the rule.
annotation	documentation The IfThenElongatedCylinderRuleType defines a conditional rule that applies if the feature under consideration is					

	an elongated cylinder. The rule contains an inherited 'if' part, an inherited ThenPoints part, and two more 'then' parts. The 'then' parts of the rule apply if the 'if' part evaluates to true.
--	--

element **IfThenElongatedCylinderRuleType/ThenPointStrategy**

diagram	
type	ElongatedCylinderPointSamplingStrategyType
properties	minOcc 0 maxOcc 1 content complex
children	PointSamplingStrategyEnum UserDefinedStrategy
annotation	documentation The optional ThenPointStrategy element is the point sampling strategy to be used if the 'if' part of the rule does not exist or evaluates to true.

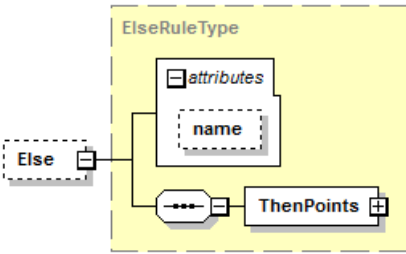
element **IfThenElongatedCylinderRuleType/ThenFittingAlgorithm**

diagram	
type	FeatureOfSizeSubstituteFeatureAlgorithmType
properties	minOcc 0 maxOcc 1 content complex
children	FeatureOfSizeSubstituteFeatureAlgorithmEnum OtherFeatureOfSizeSubstituteFeatureAlgorithm
annotation	documentation The optional ThenFittingAlgorithm element is the substitute feature algorithm to be used if the 'if' part of the rule does not exist or evaluates to true.

complexType IfThenElseFeatureRulesType

diagram	
children	IfThenFeatureRule Else
used by	element FeatureRulesType/IfThenElseFeatureRules
annotation	<p>documentation</p> <p>The IfThenElseFeatureRulesType defines a set of rules for selecting (1) the number or density of hit points for a single feature, possibly with an associated characteristic, and/or (2) a strategy for selecting the points, and/or (3) the substitute feature algorithm to use to fit the feature to the hit points. To evaluate an IfThenElseFeatureRulesType, each IfThenFeatureRule should be considered in order until the 'if' part of one of them evaluates to true, at which point the 'then' part of the rule should be evaluated and the value of the point quantity and or point sampling strategy returned. In this case, the remainder of the IfThenFeatureRules should be ignored. If the 'if' part of no IfThenFeatureRule evaluates to true, then the 'then' part of the Else should be evaluated and returned.</p>

element **IfThenElseFeatureRulesType/Else**

diagram						
type	ElseRuleType					
properties	minOcc	0				
	maxOcc	1				
	content	complex				
children	ThenPoints					
attributes	Name	Type	Use	Default	Fixed	Annotation
	name	xs:token				documentation The optional name attribute is the name of the rule.
annotation	documentation The Else element is an unconditional rule for selecting the number or density of hit points.					

complexType **IfThenExtrudedCrossSectionRuleType**

diagram						
type	extension of IfThenFeatureRuleType					
properties	base IfThenFeatureRuleType					
children	BooleanExpression ThenPoints ThenPointStrategy ThenFittingAlgorithm					
used by	element IfThenExtrudedCrossSectionRule					
attributes	Name name	Type xs:token	Use	Default	Fixed	Annotation documentation The optional name attribute is the name of the rule.
annotation	documentation The IfThenExtrudedCrossSectionRuleType defines a conditional rule that applies if the feature under consideration					

	is an extruded cross-section. The rule contains an inherited 'if' part, an inherited ThenPoints part, and two more 'then' parts. The 'then' parts of the rule apply if the 'if' part evaluates to true.
--	---

element **IfThenExtrudedCrossSectionRuleType/ThenPointStrategy**

diagram	
type	ExtrudedCrossSectionPointSamplingStrategyType
properties	minOcc 0 maxOcc 1 content complex
children	PointSamplingStrategyEnum UserDefinedStrategy
annotation	documentation The optional ThenPointStrategy element is the point sampling strategy to be used if the 'if' part of the rule does not exist or evaluates to true.

element **IfThenExtrudedCrossSectionRuleType/ThenFittingAlgorithm**

diagram	
type	CurveSubstituteFeatureAlgorithmType
properties	minOcc 0 maxOcc 1 content complex
children	CurveSubstituteFeatureAlgorithmEnum OtherCurveSubstituteFeatureAlgorithm
annotation	documentation The optional ThenFittingAlgorithm element is the substitute feature algorithm to be used if the 'if' part of the rule does not exist or evaluates to true.

complexType IfThenFeatureRuleType

diagram						
type	extension of PointRuleBaseType					
properties	base PointRuleBaseType					
children	BooleanExpression ThenPoints					
used by	element complexTypes	IfThenFeatureRule IfThenArcRuleType IfThenCircleRuleType IfThenConeRuleType IfThenConicalSegmentRuleType IfThenCuboidRuleType IfThenCurveRuleType IfThenCylinderRuleType IfThenCylindricalSegmentRuleType IfThenEllipseRuleType IfThenElongatedCylinderRuleType IfThenExtrudedCrossSectionRuleType IfThenLineRuleType IfThenOppositeLinesRuleType IfThenOppositePlanesRuleType IfThenPlaneRuleType IfThenPointDefinedCurveRuleType IfThenPointDefinedSurfaceRuleType IfThenPointRuleType IfThenSphereRuleType IfThenSphericalSegmentRuleType IfThenSurfaceOfRevolutionRuleType IfThenSurfaceRuleType IfThenToroidalSegmentRuleType IfThenTorusRuleType				
attributes	Name name	Type	Use	Default	Fixed	Annotation documentation The optional name attribute is the name of the

	rule.
annotation	documentation The IfThenFeatureRuleType defines a rule containing an optional 'if' part and an optional 'then points' part. The 'then points' part of the rule applies if the 'if' part does not exist or evaluates to true.

element **IfThenFeatureRuleType/ThenPoints**

diagram	
type	ThenPointsType
properties	minOcc 0 maxOcc 1 content complex
children	NumberOfPoints MinPoints PointDensity MinPointDensity
annotation	documentation The optional ThenPoints element gives the quantity of points to use if the 'if' part evaluates to true.

complexType **IfThenLineRuleType**

diagram	<p>The diagram shows the structure of IfThenLineRuleType. It is an extension of IfThenFeatureRuleType. It has an optional attribute name. The main body is a choice between a BooleanExpression and a choice between ThenPointStrategy and ThenFittingAlgorithm. The BooleanExpression includes various logical and feature-based conditions like And, ArithmeticEqual, BooleanEqual, Characteristics, ConstantIs, FeatureIsDatum, FeatureIsInternal, GreaterOrEqual, GreaterThan, LessOrEqual, LessThan, Not, Or, SamplingRigors, ShapeClassIs, and ThenPoints.</p>					
type	extension of IfThenFeatureRuleType					
properties	base IfThenFeatureRuleType					
children	BooleanExpression ThenPoints ThenPointStrategy ThenFittingAlgorithm					
used by	element IfThenLineRule					
attributes	Name name	Type xs:token	Use	Default	Fixed	Annotation documentation The optional name attribute is the name of the rule.
annotation	documentation The IfThenLineRuleType defines a conditional rule that applies if the feature under consideration is a line. The rule					

	contains an inherited 'if' part, an inherited ThenPoints part, and two more 'then' parts. The 'then' parts of the rule apply if the 'if' part does not exist or evaluates to true.
--	--

element **IfThenLineRuleType/ThenPointStrategy**

diagram	
type	LinePointSamplingStrategyType
properties	minOcc 0 maxOcc 1 content complex
children	PointSamplingStrategyEnum UserDefinedStrategy
annotation	documentation The optional ThenPointStrategy element is the point sampling strategy to be used if the 'if' part of the rule does not exist or evaluates to true.

element **IfThenLineRuleType/ThenFittingAlgorithm**

diagram	
type	NonFeatureOfSizeSubstituteFeatureAlgorithmType
properties	minOcc 0 maxOcc 1 content complex
children	NonFeatureOfSizeSubstituteFeatureAlgorithmEnum OtherNonFeatureOfSizeSubstituteFeatureAlgorithm
annotation	documentation The optional ThenFittingAlgorithm element is the substitute feature algorithm to be used if the 'if' part of the rule does not exist or evaluates to true.

complexType **IfThenOppositeLinesRuleType**

diagram						
type	extension of IfThenFeatureRuleType					
properties	base IfThenFeatureRuleType					
children	BooleanExpression ThenPoints ThenPointStrategy ThenFittingAlgorithm					
used by	element IfThenOppositeLinesRule					
attributes	Name name	Type xs:token	Use	Default	Fixed	Annotation documentation The optional name attribute is the name of the rule.
annotation	documentation The IfThenOppositeLinesRuleType defines a conditional rule that applies if the feature under consideration is an					

	opposite lines. The rule contains an inherited 'if' part, an inherited ThenPoints part, and two more 'then' parts. The 'then' parts of the rule apply if the 'if' part does not exist or evaluates to true.
--	---

element **IfThenOppositeLinesRuleType/ThenPointStrategy**

diagram	
type	OppositeLinesPointSamplingStrategyType
properties	minOcc 0 maxOcc 1 content complex
children	PointSamplingStrategyEnum UserDefinedStrategy
annotation	documentation The optional ThenPointStrategy element is the point sampling strategy to be used if the 'if' part of the rule does not exist or evaluates to true.

element **IfThenOppositeLinesRuleType/ThenFittingAlgorithm**

diagram	
type	FeatureOfSizeSubstituteFeatureAlgorithmType
properties	minOcc 0 maxOcc 1 content complex
children	FeatureOfSizeSubstituteFeatureAlgorithmEnum OtherFeatureOfSizeSubstituteFeatureAlgorithm
annotation	documentation The optional ThenFittingAlgorithm element is the substitute feature algorithm to be used if the 'if' part of the rule does not exist or evaluates to true.

complexType **IfThenOppositePlanesRuleType**

diagram						
type	extension of IfThenFeatureRuleType					
properties	base IfThenFeatureRuleType					
children	BooleanExpression ThenPoints ThenPointStrategy ThenFittingAlgorithm					
used by	element IfThenOppositePlanesRule					
attributes	Name name	Type xs:token	Use	Default	Fixed	Annotation documentation The optional name attribute is the name of the rule.
annotation	documentation The IfThenOppositePlanesRuleType defines a conditional rule that applies if the feature under consideration is an					

	opposite planes. The rule contains an inherited 'if' part, an inherited ThenPoints part, and two more 'then' parts. The 'then' parts of the rule apply if the 'if' part does not exist or evaluates to true.
--	--

element IfThenOppositePlanesRuleType/ThenPointStrategy

diagram	
type	OppositePlanesPointSamplingStrategyType
properties	minOcc 0 maxOcc 1 content complex
children	PointSamplingStrategyEnum UserDefinedStrategy
annotation	documentation The optional ThenPointStrategy element is the point sampling strategy to be used if the 'if' part of the rule does not exist or evaluates to true.

element IfThenOppositePlanesRuleType/ThenFittingAlgorithm

diagram	
type	FeatureOfSizeSubstituteFeatureAlgorithmType
properties	minOcc 0 maxOcc 1 content complex
children	FeatureOfSizeSubstituteFeatureAlgorithmEnum OtherFeatureOfSizeSubstituteFeatureAlgorithm
annotation	documentation The optional ThenFittingAlgorithm element is the substitute feature algorithm to be used if the 'if' part of the rule does not exist or evaluates to true.

complexType **IfThenPlaneRuleType**

diagram	<p>The diagram shows the structure of IfThenPlaneRuleType. It is a complex type that extends IfThenFeatureRuleType. It has an optional attribute name. The content model is a sequence of elements: BooleanExpression, ThenPoints, ThenPointStrategy, and ThenFittingAlgorithm. BooleanExpression is a complex type with a choice of elements: And, ArithmeticEqual, BooleanEqual, Characteristics, ConstantIs, FeatureIsDatum, FeatureIsInternal, GreaterOrEqual, GreaterThan, LessOrEqual, LessThan, Not, Or, SamplingRigorIs, ShapeClassIs, and ThenPoints. All elements in the choice are optional.</p>					
type	extension of IfThenFeatureRuleType					
properties	base IfThenFeatureRuleType					
children	BooleanExpression ThenPoints ThenPointStrategy ThenFittingAlgorithm					
used by	element IfThenPlaneRule					
attributes	Name name	Type xs:token	Use	Default	Fixed	Annotation documentation The optional name attribute is the name of the rule.
annotation	documentation The IfThenPlaneRuleType defines a conditional rule that applies if the feature under consideration is a plane. The					

	rule contains an inherited 'if' part, an inherited ThenPoints part, and two more 'then' parts. The 'then' parts of the rule apply if the 'if' part does not exist or evaluates to true.
--	---

element IfThenPlaneRuleType/ThenPointStrategy

diagram	
type	PlanePointSamplingStrategyType
properties	minOcc 0 maxOcc 1 content complex
children	PointSamplingStrategyEnum UserDefinedStrategy
annotation	documentation The optional ThenPointStrategy element is the point sampling strategy to be used if the 'if' part of the rule does not exist or evaluates to true.

element IfThenPlaneRuleType/ThenFittingAlgorithm

diagram	
type	NonFeatureOfSizeSubstituteFeatureAlgorithmType
properties	minOcc 0 maxOcc 1 content complex
children	NonFeatureOfSizeSubstituteFeatureAlgorithmEnum OtherNonFeatureOfSizeSubstituteFeatureAlgorithm
annotation	documentation The optional ThenFittingAlgorithm element is the substitute feature algorithm to be used if the 'if' part of the rule does not exist or evaluates to true.

complexType **IfThenPointDefinedCurveRuleType**

diagram						
type	extension of IfThenFeatureRuleType					
properties	base IfThenFeatureRuleType					
children	BooleanExpression ThenPoints ThenPointStrategy ThenFittingAlgorithm					
used by	element IfThenPointDefinedCurveRule					
attributes	Name name	Type xs:token	Use	Default	Fixed	Annotation documentation The optional name attribute is the name of the rule.
annotation	documentation The IfThenPointDefinedCurveRuleType defines a conditional rule that applies if the feature under consideration is a					

	point-defined curve. The rule contains an inherited 'if' part, an inherited ThenPoints part, and two more 'then' parts. The 'then' parts of the rule apply if the 'if' part does not exist or evaluates to true.
--	--

element IfThenPointDefinedCurveRuleType/ThenPointStrategy

diagram	
type	PointDefinedCurvePointSamplingStrategyType
properties	minOcc 0 maxOcc 1 content complex
children	PointSamplingStrategyEnum UserDefinedStrategy
annotation	documentation The optional ThenPointStrategy element is the point sampling strategy to be used if the 'if' part of the rule does not exist or evaluates to true.

element IfThenPointDefinedCurveRuleType/ThenFittingAlgorithm

diagram	
type	CurveSubstituteFeatureAlgorithmType
properties	minOcc 0 maxOcc 1 content complex
children	CurveSubstituteFeatureAlgorithmEnum OtherCurveSubstituteFeatureAlgorithm
annotation	documentation The optional ThenFittingAlgorithm element is the substitute feature algorithm to be used if the 'if' part of the rule does not exist or evaluates to true.

complexType **IfThenPointDefinedSurfaceRuleType**

diagram						
type	extension of IfThenFeatureRuleType					
properties	base IfThenFeatureRuleType					
children	BooleanExpression ThenPoints ThenPointStrategy ThenFittingAlgorithm					
used by	element IfThenPointDefinedSurfaceRule					
attributes	Name name	Type xs:token	Use	Default	Fixed	Annotation documentation The optional name attribute is the name of the rule.
annotation	documentation The IfThenPointDefinedSurfaceRuleType defines a conditional rule that applies if the feature under consideration is					

	a point-defined surface. The rule contains an inherited 'if' part, an inherited ThenPoints part, and two more 'then' parts. The 'then' parts of the rule apply if the 'if' part does not exist or evaluates to true.
--	--

element **IfThenPointDefinedSurfaceRuleType/ThenPointStrategy**

diagram	
type	PointDefinedSurfacePointSamplingStrategyType
properties	minOcc 0 maxOcc 1 content complex
children	PointSamplingStrategyEnum UserDefinedStrategy
annotation	documentation The optional ThenPointStrategy element is the point sampling strategy to be used if the 'if' part of the rule does not exist or evaluates to true.

element **IfThenPointDefinedSurfaceRuleType/ThenFittingAlgorithm**

diagram	
type	SurfaceSubstituteFeatureAlgorithmType
properties	minOcc 0 maxOcc 1 content complex
children	SurfaceSubstituteFeatureAlgorithmEnum OtherSurfaceSubstituteFeatureAlgorithm
annotation	documentation The optional ThenPointAlgorithm element is the substitute feature algorithm to be used if the 'if' part of the rule does not exist or evaluates to true.

complexType **IfThenPointRuleType**

diagram	<p>The diagram shows the structure of IfThenPointRuleType. It is a complex type that extends IfThenFeatureRuleType. It contains an optional attribute name of type xs:token. The main body of the type is a choice between two elements: BooleanExpression and ThenPointStrategy. BooleanExpression is itself a choice of several other types: And, ArithmeticEqual, BooleanEqual, Characteristics, ConstantIs, FeatureIsDatum, FeatureIsInternal, GreaterOrEqual, GreaterThan, LessOrEqual, LessThan, Not, Or, SamplingRigors, ShapeClassIs, and ThenPoints. All these elements are optional, as indicated by the '+' symbol in their UML boxes.</p>					
type	extension of IfThenFeatureRuleType					
properties	base IfThenFeatureRuleType					
children	BooleanExpression ThenPoints ThenPointStrategy					
used by	element IfThenPointRule					
attributes	Name name	Type xs:token	Use	Default	Fixed	Annotation documentation The optional name attribute is the name of the rule.
annotation	documentation The IfThenPointRuleType defines a conditional rule that applies if the feature under consideration is a point. The rule contains an inherited 'if' part, an inherited ThenPoints part, and one more 'then' part. The 'then' parts of the rule apply if the 'if' part does not exist or evaluates to true.					

element **IfThenPointRuleType/ThenPointStrategy**

diagram	
type	PointPointSamplingStrategyType
properties	minOcc 0 maxOcc 1 content complex
children	PointSamplingStrategyEnum UserDefinedStrategy
annotation	documentation The optional ThenPointStrategy element is the point sampling strategy to be used if the 'if' part of the rule does not exist or evaluates to true.

complexType **IfThenSphereRuleType**

diagram	<p>The diagram shows the structure of IfThenSphereRuleType. It is a complex type that extends IfThenFeatureRuleType. It has an optional attribute name. The body contains a sequence of elements: BooleanExpression, And, ArithmeticEqual, BooleanEqual, Characteristics, ConstantIs, FeatureIsDatum, FeatureIsInternal, GreaterOrEqual, GreaterThan, LessOrEqual, LessThan, Not, Or, SamplingRigorsIs, ShapeClassIs, and ThenPoints. This sequence is followed by an optional choice of ThenPointStrategy and ThenFittingAlgorithm.</p>					
type	extension of IfThenFeatureRuleType					
properties	base IfThenFeatureRuleType					
children	BooleanExpression ThenPoints ThenPointStrategy ThenFittingAlgorithm					
used by	element IfThenSphereRule					
attributes	Name name	Type xs:token	Use	Default	Fixed	Annotation documentation The optional name attribute is the name of the rule.
annotation	documentation The IfThenSphereRuleType defines a conditional rule that applies if the feature under consideration is a sphere.					

	The rule contains an inherited 'if' part, an inherited ThenPoints part, and two more 'then' parts. The 'then' parts of the rule apply if the 'if' part does not exist or evaluates to true.
--	---

element IfThenSphereRuleType/ThenPointStrategy

diagram	
type	SpherePointSamplingStrategyType
properties	minOcc 0 maxOcc 1 content complex
children	PointSamplingStrategyEnum UserDefinedStrategy
annotation	documentation The optional ThenPointStrategy element is the point sampling strategy to be used if the 'if' part of the rule does not exist or evaluates to true.

element IfThenSphereRuleType/ThenFittingAlgorithm

diagram	
type	FeatureOfSizeSubstituteFeatureAlgorithmType
properties	minOcc 0 maxOcc 1 content complex
children	FeatureOfSizeSubstituteFeatureAlgorithmEnum OtherFeatureOfSizeSubstituteFeatureAlgorithm
annotation	documentation The optional ThenFittingAlgorithm element is the substitute feature algorithm to be used if the 'if' part of the rule does not exist or evaluates to true.

complexType **IfThenSphericalSegmentRuleType**

diagram						
type	extension of IfThenFeatureRuleType					
properties	base IfThenFeatureRuleType					
children	BooleanExpression ThenPoints ThenPointStrategy ThenFittingAlgorithm					
used by	element IfThenSphericalSegmentRule					
attributes	Name name	Type xs:token	Use	Default	Fixed	Annotation documentation The optional name attribute is the name of the rule.
annotation	documentation The IfThenSphericalSegmentRuleType defines a conditional rule that applies if the feature under consideration is a					

	spherical segment. The rule contains an inherited 'if' part, an inherited ThenPoints part, and two more 'then' parts. The 'then' parts of the rule apply if the 'if' part does not exist or evaluates to true.
--	--

element IfThenSphericalSegmentRuleType/ThenPointStrategy

diagram	
type	SphericalSegmentPointSamplingStrategyType
properties	minOcc 0 maxOcc 1 content complex
children	PointSamplingStrategyEnum UserDefinedStrategy
annotation	documentation The optional ThenPointStrategy element is the point sampling strategy to be used if the 'if' part of the rule does not exist or evaluates to true.

element IfThenSphericalSegmentRuleType/ThenFittingAlgorithm

diagram	
type	NonFeatureOfSizeSubstituteFeatureAlgorithmType
properties	minOcc 0 maxOcc 1 content complex
children	NonFeatureOfSizeSubstituteFeatureAlgorithmEnum OtherNonFeatureOfSizeSubstituteFeatureAlgorithm
annotation	documentation The optional ThenFittingAlgorithm element is the substitute feature algorithm to be used if the 'if' part of the rule does not exist or evaluates to true.

complexType **IfThenSurfaceOfRevolutionRuleType**

diagram						
type	extension of IfThenFeatureRuleType					
properties	base IfThenFeatureRuleType					
children	BooleanExpression ThenPoints ThenPointStrategy ThenFittingAlgorithm					
used by	element IfThenSurfaceOfRevolutionRule					
attributes	Name name	Type xs:token	Use	Default	Fixed	Annotation documentation The optional name attribute is the name of the rule.
annotation	documentation The IfThenSurfaceOfRevolutionRuleType defines a conditional rule that applies if the feature under consideration is					

	a surface of revolution. The rule contains an inherited 'if' part, an inherited ThenPoints part, and two more 'then' parts. The 'then' parts of the rule apply if the 'if' part does not exist or evaluates to true.
--	--

element IfThenSurfaceOfRevolutionRuleType/ThenPointStrategy

diagram	
type	SurfaceOfRevolutionPointSamplingStrategyType
properties	minOcc 0 maxOcc 1 content complex
children	PointSamplingStrategyEnum UserDefinedStrategy
annotation	documentation The optional ThenPointStrategy element is the point sampling strategy to be used if the 'if' part of the rule does not exist or evaluates to true.

element IfThenSurfaceOfRevolutionRuleType/ThenFittingAlgorithm

diagram	
type	CurveSubstituteFeatureAlgorithmType
properties	minOcc 0 maxOcc 1 content complex
children	CurveSubstituteFeatureAlgorithmEnum OtherCurveSubstituteFeatureAlgorithm
annotation	documentation The optional ThenFittingAlgorithm element is the substitute feature algorithm to be used if the 'if' part of the rule does not exist or evaluates to true.

complexType **IfThenSurfaceRuleType**

diagram	<p>The diagram shows the structure of IfThenSurfaceRuleType. It is a complex type that extends IfThenFeatureRuleType. It has an optional attribute name. The if part is a BooleanExpression, which can be one of the following: And, ArithmeticEqual, BooleanEqual, Characteristics, ConstantIs, FeatureIsDatum, FeatureIsInternal, GreaterOrEqual, GreaterThan, LessOrEqual, LessThan, Not, Or, SamplingRigorIs, ShapeClassIs, or ThenPoints. The then part is ThenPoints.</p>					
type	extension of IfThenFeatureRuleType					
properties	base IfThenFeatureRuleType					
children	BooleanExpression ThenPoints					
used by	element IfThenSurfaceRule					
attributes	Name name	Type xs:token	Use	Default	Fixed	Annotation documentation The optional name attribute is the name of the rule.
annotation	documentation The IfThenSurfaceRuleType defines a conditional rule that applies if the feature under consideration is a surface. The rule contains an inherited 'if' part and an inherited ThenPoints part. The 'then' part of the rule applies if the 'if' part does not exist or evaluates to true.					

complexType **IfThenToroidalSegmentRuleType**

diagram	<p>The diagram shows the structure of IfThenToroidalSegmentRuleType. It is a complex type that extends IfThenFeatureRuleType. It has an optional attribute name of type xs:string. The body contains a sequence of elements: BooleanExpression, And, ArithmeticEqual, BooleanEqual, Characteristics, ConstantIs, FeatureIsDatum, FeatureIsInternal, GreaterOrEqual, GreaterThan, LessOrEqual, LessThan, Not, Or, SamplingRigoris, ShapeClassIs, and ThenPoints. Additionally, there is an optional choice between ThenPointStrategy and ThenFittingAlgorithm.</p>					
type	extension of IfThenFeatureRuleType					
properties	base IfThenFeatureRuleType					
children	BooleanExpression ThenPoints ThenPointStrategy ThenFittingAlgorithm					
used by	element IfThenToroidalSegmentRule					
attributes	Name name	Type xs:string	Use Optional	Default	Fixed	Annotation documentation The optional name attribute is the name of the rule.
annotation	documentation The IfThenToroidalSegmentRuleType defines a conditional rule that applies if the feature under consideration is a					

	toroidal segment. The rule contains an inherited 'if' part, an inherited ThenPoints part, and two more 'then' parts. The 'then' parts of the rule apply if the 'if' part does not exist or evaluates to true.
--	---

element IfThenToroidalSegmentRuleType/ThenPointStrategy

diagram	
type	ToroidalSegmentPointSamplingStrategyType
properties	minOcc 0 maxOcc 1 content complex
children	PointSamplingStrategyEnum UserDefinedStrategy
annotation	documentation The optional ThenPointStrategy element is the point sampling strategy to be used if the 'if' part of the rule does not exist or evaluates to true.

element IfThenToroidalSegmentRuleType/ThenFittingAlgorithm

diagram	
type	NonFeatureOfSizeSubstituteFeatureAlgorithmType
properties	minOcc 0 maxOcc 1 content complex
children	NonFeatureOfSizeSubstituteFeatureAlgorithmEnum OtherNonFeatureOfSizeSubstituteFeatureAlgorithm
annotation	documentation The optional ThenFittingAlgorithm element is the substitute feature algorithm to be used if the 'if' part of the rule does not exist or evaluates to true.

complexType **IfThenTorusRuleType**

diagram						
type	extension of IfThenFeatureRuleType					
properties	base IfThenFeatureRuleType					
children	BooleanExpression ThenPoints ThenPointStrategy ThenFittingAlgorithm					
used by	element IfThenTorusRule					
attributes	Name name	Type xs:token	Use	Default	Fixed	Annotation documentation The optional name attribute is the name of the rule.
annotation	documentation The IfThenTorusRuleType defines a conditional rule that applies if the feature under consideration is a torus. The					

	rule contains an inherited 'if' part, an inherited ThenPoints part, and two more 'then' parts. The 'then' parts of the rule apply if the 'if' part does not exist or evaluates to true.
--	---

element **IfThenTorusRuleType/ThenPointStrategy**

diagram	
type	TorusPointSamplingStrategyType
properties	minOcc 0 maxOcc 1 content complex
children	PointSamplingStrategyEnum UserDefinedStrategy
annotation	documentation The optional ThenPointStrategy element is the point sampling strategy to be used if the 'if' part of the rule does not exist or evaluates to true.

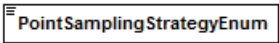
element **IfThenTorusRuleType/ThenFittingAlgorithm**

diagram	
type	FeatureOfSizeSubstituteFeatureAlgorithmType
properties	minOcc 0 maxOcc 1 content complex
children	FeatureOfSizeSubstituteFeatureAlgorithmEnum OtherFeatureOfSizeSubstituteFeatureAlgorithm
annotation	documentation The optional ThenFittingAlgorithm element is the substitute feature algorithm to be used if the 'if' part of the rule does not exist or evaluates to true.


complexType **LinePointSamplingStrategyType**

diagram	
children	PointSamplingStrategyEnum UserDefinedStrategy
used by	element IfThenLineRuleType/ThenPointStrategy
annotation	documentation The LinePointSamplingStrategyType defines a point sampling strategy for a line. The strategy is either an enumerated line strategy or a user defined strategy.

element **LinePointSamplingStrategyType/PointSamplingStrategyEnum**

diagram							
type	OpenCurvePointSamplingStrategyEnumType						
properties	content simple						
facets	<table><tr><td>Kind</td><td>Value</td><td>Annotation</td></tr><tr><td>enumeration</td><td>POINTS</td><td></td></tr></table>	Kind	Value	Annotation	enumeration	POINTS	
Kind	Value	Annotation					
enumeration	POINTS						
annotation	<div>documentation</div> <div>The PointSamplingStrategyEnum element is a commonly used point sampling strategy for a line.</div>						

element **LinePointSamplingStrategyType/UserDefinedStrategy**

diagram	
type	xs:token
properties	content simple
annotation	documentation The UserDefinedStrategy element is the name of a point sampling strategy defined by the user.

complexType **MaxFeatureRulesType**

diagram	
children	IfThenFeatureRule Else
used by	element FeatureRulesType/MaxFeatureRules
annotation	<p>documentation</p> <p>The MaxFeatureRulesType defines a set of rules for selecting (1) the number or density of hit points for a single feature, possibly with an associated characteristic and/or (2) a strategy for selecting the points, and/or (3) the substitute feature algorithm to use to fit the feature to the hit points. To evaluate a MaxFeatureRulesType, keep track of a current answer for point quantity, which is initially set to zero. Each IfThenFeatureRule should be considered in order. If the 'if' part of a rule evaluates to true, the 'then' part of the rule should be evaluated and if that value is greater than the current answer, the current answer should be set to that value. If the 'if' part of no IfThenFeatureRule evaluates to true, so that the current answer is still zero after all IfThenFeatureRules have been processed, then the 'then' part of the Else should be evaluated and the current answer set to that value. The returned value for point quantity is the final value of the current answer. In addition to keeping track of the current</p>

	answer for point quantity, the evaluating system should collect all the strategies and substitute feature algorithms from the 'then' parts that were evaluated.
--	---


element MaxFeatureRulesType/Else

diagram						
type	ElseRuleType					
properties	minOcc	0	maxOcc	1	content	complex
children	ThenPoints					
attributes	Name	Type	Use	Default	Fixed	Annotation
	name	xs:token				documentation The optional name attribute is the name of the rule.
annotation	documentation The optional Else element is an unconditional rule for selecting the number or density of hit points.					


complexType OppositeLinesPointSamplingStrategyType

diagram	<pre>classDiagram class OppositeLinesPointSamplingStrategyType class PointSamplingStrategyEnum class UserDefinedStrategy OppositeLinesPointSamplingStrategyType -- > PointSamplingStrategyEnum OppositeLinesPointSamplingStrategyType -- > UserDefinedStrategy</pre>
children	PointSamplingStrategyEnum UserDefinedStrategy
used by	element IfThenOppositeLinesRuleType/ThenPointStrategy
annotation	documentation The OppositeLinesPointSamplingStrategyType defines a point sampling strategy for an opposite lines. The strategy is either an enumerated opposite lines strategy or a user defined strategy.

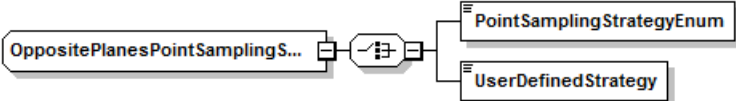
element OppositeLinesPointSamplingStrategyType/PointSamplingStrategyEnum

diagram							
type	OpenCurvePointSamplingStrategyEnumType						
properties	content simple						
facets	<table><tr><td>Kind</td><td>Value</td><td>Annotation</td></tr><tr><td>enumeration</td><td>POINTS</td><td></td></tr></table>	Kind	Value	Annotation	enumeration	POINTS	
Kind	Value	Annotation					
enumeration	POINTS						
annotation	<p>documentation</p> <p>The PointSamplingStrategyEnum element is a commonly used point sampling strategy for an opposite lines.</p>						

element **OppositeLinesPointSamplingStrategyType/UserDefinedStrategy**

diagram	
type	xs:token
properties	content simple
annotation	documentation The UserDefinedStrategy element is the name of a point sampling strategy defined by the user.


complexType **OppositePlanesPointSamplingStrategyType**

diagram	
children	PointSamplingStrategyEnum UserDefinedStrategy
used by	element IfThenOppositePlanesRuleType/ThenPointStrategy
annotation	documentation The OppositePlanesPointSamplingStrategyType defines a point sampling strategy for an opposite planes. The strategy is either an enumerated opposite planes strategy or a user defined strategy.

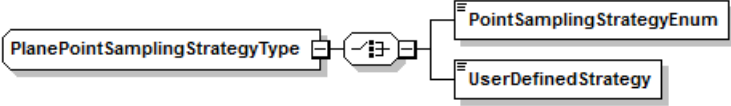
element **OppositePlanesPointSamplingStrategyType/PointSamplingStrategyEnum**

diagram	<div><div><div><div><div></div><div></div><div></div></div><div><div><div></div><div></div><div></div></div></div><div>PointSamplingStrategyEnum</div></div></div></div>																								
type	PlanePointSamplingStrategyEnumType																								
properties	content simple																								
facets	<table><tr><td>Kind</td><td>Value</td><td>Annotation</td></tr><tr><td>enumeration</td><td>ORTHOGONALGRID</td><td></td></tr><tr><td>enumeration</td><td>POLARGRID</td><td></td></tr><tr><td>enumeration</td><td>SPECIFIEDGRID</td><td></td></tr><tr><td>enumeration</td><td>STRATIFIED</td><td></td></tr><tr><td>enumeration</td><td>SPIRAL</td><td></td></tr><tr><td>enumeration</td><td>SPIDERWEB</td><td></td></tr><tr><td>enumeration</td><td>POINTS</td><td></td></tr></table>	Kind	Value	Annotation	enumeration	ORTHOGONALGRID		enumeration	POLARGRID		enumeration	SPECIFIEDGRID		enumeration	STRATIFIED		enumeration	SPIRAL		enumeration	SPIDERWEB		enumeration	POINTS	
Kind	Value	Annotation																							
enumeration	ORTHOGONALGRID																								
enumeration	POLARGRID																								
enumeration	SPECIFIEDGRID																								
enumeration	STRATIFIED																								
enumeration	SPIRAL																								
enumeration	SPIDERWEB																								
enumeration	POINTS																								
annotation	<div>documentation</div> <div>The PointSamplingStrategyEnum element is a commonly used point sampling strategy for an opposite planes.</div>																								


element **OppositePlanesPointSamplingStrategyType/UserDefinedStrategy**

diagram	
type	xs:token
properties	content simple
annotation	documentation The UserDefinedStrategy element is the name of a point sampling strategy defined by the user.


complexType **PlanePointSamplingStrategyType**

diagram	
children	PointSamplingStrategyEnum UserDefinedStrategy
used by	element IfThenPlaneRuleType/ThenPointStrategy
annotation	documentation The PlanePointSamplingStrategyType defines a point sampling strategy for a plane. The strategy is either an enumerated plane strategy or a user defined strategy.

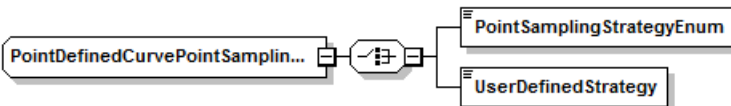
element **PlanePointSamplingStrategyType/PointSamplingStrategyEnum**

diagram																									
type	PlanePointSamplingStrategyEnumType																								
properties	content simple																								
facets	<table><tr><td>Kind</td><td>Value</td><td>Annotation</td></tr><tr><td>enumeration</td><td>ORTHOGONALGRID</td><td></td></tr><tr><td>enumeration</td><td>POLARGRID</td><td></td></tr><tr><td>enumeration</td><td>SPECIFIEDGRID</td><td></td></tr><tr><td>enumeration</td><td>STRATIFIED</td><td></td></tr><tr><td>enumeration</td><td>SPIRAL</td><td></td></tr><tr><td>enumeration</td><td>SPIDERWEB</td><td></td></tr><tr><td>enumeration</td><td>POINTS</td><td></td></tr></table>	Kind	Value	Annotation	enumeration	ORTHOGONALGRID		enumeration	POLARGRID		enumeration	SPECIFIEDGRID		enumeration	STRATIFIED		enumeration	SPIRAL		enumeration	SPIDERWEB		enumeration	POINTS	
Kind	Value	Annotation																							
enumeration	ORTHOGONALGRID																								
enumeration	POLARGRID																								
enumeration	SPECIFIEDGRID																								
enumeration	STRATIFIED																								
enumeration	SPIRAL																								
enumeration	SPIDERWEB																								
enumeration	POINTS																								
annotation	<div>documentation</div> <div>The PointSamplingStrategyEnum element is a commonly used point sampling strategy for a plane.</div>																								


element **PlanePointSamplingStrategyType/UserDefinedStrategy**

diagram	
type	xs:token
properties	content simple
annotation	documentation The UserDefinedStrategy element is the name of a point sampling strategy defined by the user.


complexType **PointDefinedCurvePointSamplingStrategyType**

diagram	
children	PointSamplingStrategyEnum UserDefinedStrategy
used by	element IfThenPointDefinedCurveRuleType/ThenPointStrategy
annotation	documentation The PointDefinedCurvePointSamplingStrategyType defines a point sampling strategy for a point-defined curve. The strategy is either an enumerated point-defined curve strategy or a user defined strategy.

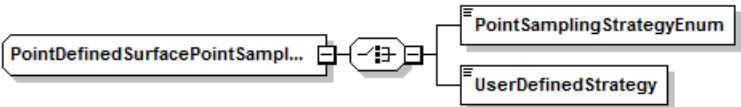
element PointDefinedCurvePointSamplingStrategyType/PointSamplingStrategyEnum

diagram										
type	PointDefinedPointSamplingStrategyEnumType									
properties	content simple									
facets	<table><tr><td>Kind</td><td>Value</td><td>Annotation</td></tr><tr><td>enumeration</td><td>GIVENPOINTS</td><td></td></tr><tr><td>enumeration</td><td>POINTS</td><td></td></tr></table>	Kind	Value	Annotation	enumeration	GIVENPOINTS		enumeration	POINTS	
Kind	Value	Annotation								
enumeration	GIVENPOINTS									
enumeration	POINTS									
annotation	<p>documentation</p> <p>The PointSamplingStrategyEnum element is a commonly used point sampling strategy for a point-defined curve.</p>									


element PointDefinedCurvePointSamplingStrategyType/UserDefinedStrategy

diagram	
type	xs:token
properties	content simple
annotation	documentation The UserDefinedStrategy element is the name of a point sampling strategy defined by the user.


complexType PointDefinedSurfacePointSamplingStrategyType

diagram	
children	PointSamplingStrategyEnum UserDefinedStrategy
used by	element IfThenPointDefinedSurfaceRuleType/ThenPointStrategy
annotation	documentation The PointDefinedSurfacePointSamplingStrategyType defines a point sampling strategy for a point-defined surface. The strategy is either an enumerated point-defined surface strategy or a user defined strategy.

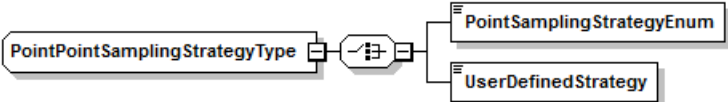
element PointDefinedSurfacePointSamplingStrategyType/PointSamplingStrategyEnum

diagram			
type	PointDefinedPointSamplingStrategyEnumType		
properties	content	simple	
facets	Kind enumeration	Value GIVENPOINTS	Annotation
	enumeration	POINTS	
annotation	documentation The PointSamplingStrategyEnum element is a commonly used point sampling strategy for a point-defined surface.		


element **PointDefinedSurfacePointSamplingStrategyType/UserDefinedStrategy**

diagram	
type	xs:token
properties	content simple
annotation	documentation The UserDefinedStrategy element is the name of a point sampling strategy defined by the user.


complexType **PointPointSamplingStrategyType**

diagram	
children	PointSamplingStrategyEnum UserDefinedStrategy
used by	element IfThenPointRuleType/ThenPointStrategy
annotation	documentation The PointPointSamplingStrategyType defines a point sampling strategy for a point. The strategy is either an enumerated point strategy or a user defined strategy.

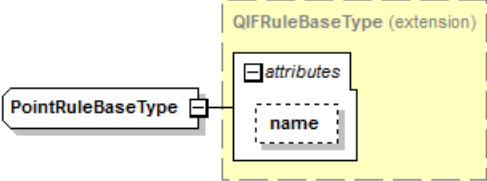
element **PointPointSamplingStrategyType/PointSamplingStrategyEnum**

diagram	
type	PointPointSamplingStrategyEnumType
properties	content simple
facets	Kind enumeration Value POINTS Annotation
annotation	documentation The PointSamplingStrategyEnum element is a commonly used point sampling strategy for a point.

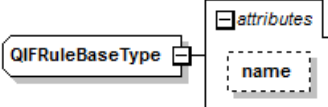
element **PointPointSamplingStrategyType/UserDefinedStrategy**

diagram	
type	xs:token
properties	content simple
annotation	documentation The UserDefinedStrategy element is the name of a point sampling strategy defined by the user.

complexType **PointRuleBaseType**

diagram						
type	extension of QIFRuleBaseType					
properties	base	QIFRuleBaseType				
	abstract	true				
used by	complexType	ElseRuleType IfThenFeatureRuleType				
attributes	Name	Type	Use	Default	Fixed	Annotation
	name	xs:token				documentation The optional name attribute is the name of the rule.
annotation	documentation The abstract PointRuleBaseType defines the base type for rules for selecting the number or density of hit points and/or a strategy for selecting the points.					

complexType **QIFRuleBaseType**

diagram						
properties	abstract	true				
used by	complexType	PointRuleBaseType				
attributes	Name	Type	Use	Default	Fixed	Annotation
	name	xs:token				documentation The optional name attribute is the name of the rule.
annotation	documentation The abstract QIFRuleBaseType defines the base type for all QIF rules.					

attribute **QIFRuleBaseType/@name**

type	xs:token
annotation	documentation The optional name attribute is the name of the rule.

complexType **QIFRulesType**

diagram	
children	Version RulesUnits FeatureRules DMESelectionRules
used by	element Rules
annotation	documentation The QIFRulesType defines a collection of sets of rules for selecting quality measurement items. Currently it contains a complete model of rules for selecting the quantity of hit points and/or the strategy for selecting hit points and/or a substitute feature algorithm. It also contains a stub model of rules for selecting DMEs. Rules sets for other types of decisions might be added.

element **QIFRulesType/Version**

diagram	
type	VersionType
properties	minOcc 0 maxOcc 1 content complex
children	TimeCreated SignOffs ThisInstanceQPid
annotation	documentation The optional Version element gives version information about the rules.

element **QIFRulesType/RulesUnits**

diagram	
type	OtherUnitsType
properties	minOcc 0 maxOcc 1 content complex
children	AreaUnit AngularUnit ForceUnit LinearUnit MassUnit PressureUnit SpeedUnit TemperatureUnit TimeUnit
annotation	documentation The optional RulesUnits element defines the units that apply in the FeatureRules. If this element does not exist, the units are the primary units given in the FileUnits element of the QIFDocument. Feature and characteristic information expressed elsewhere in other units must be converted to these units while using rules.

element **QIFRulesType/FeatureRules**

diagram	
type	<u>FeatureRulesType</u>
properties	minOcc 0 maxOcc 1 content complex

children	SamplingRigorMax IfThenElseFeatureRules MaxFeatureRules
annotation	documentation The optional FeatureRules element defines rules for selecting the quantity of hit points for a feature and/or the strategy to use for selecting the points and/or the substitute feature algorithm to use.

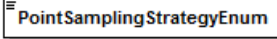
element **QIFRulesType/DMESelectionRules**

diagram	
type	DMESelectionRulesType
properties	minOcc 0 maxOcc 1 content complex
children	DMESelectionRule
annotation	documentation The optional DMESelectionRules element defines a stub for a set of rules for selecting DMEs.


complexType **SpherePointSamplingStrategyType**

diagram	
children	PointSamplingStrategyEnum UserDefinedStrategy
used by	element IfThenSphereRuleType/ThenPointStrategy
annotation	documentation The SpherePointSamplingStrategyType defines a point sampling strategy for a sphere. The strategy is either an enumerated sphere strategy or a user defined strategy.

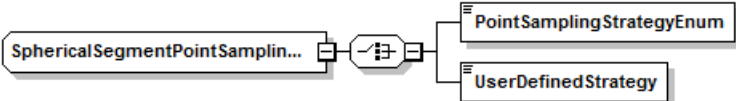
element **SpherePointSamplingStrategyType/PointSamplingStrategyEnum**

diagram																			
type	SpherePointSamplingStrategyEnumType																		
properties	content simple																		
facets	<table><tr><th>Kind</th><th>Value</th><th>Annotation</th></tr><tr><td>enumeration</td><td>ORTHOGONALGRID</td><td></td></tr><tr><td>enumeration</td><td>SPECIFIEDGRID</td><td></td></tr><tr><td>enumeration</td><td>STRATIFIED</td><td></td></tr><tr><td>enumeration</td><td>HELIX</td><td></td></tr><tr><td>enumeration</td><td>POINTS</td><td></td></tr></table>	Kind	Value	Annotation	enumeration	ORTHOGONALGRID		enumeration	SPECIFIEDGRID		enumeration	STRATIFIED		enumeration	HELIX		enumeration	POINTS	
Kind	Value	Annotation																	
enumeration	ORTHOGONALGRID																		
enumeration	SPECIFIEDGRID																		
enumeration	STRATIFIED																		
enumeration	HELIX																		
enumeration	POINTS																		
annotation	<div>documentation</div> <div>The PointSamplingStrategyEnum element is a commonly used point sampling strategy for a sphere.</div>																		

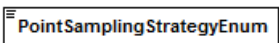
element **SpherePointSamplingStrategyType/UserDefinedStrategy**

diagram	
type	xs:token
properties	content simple
annotation	documentation The UserDefinedStrategy element is the name of a point sampling strategy defined by the user.


complexType **SphericalSegmentPointSamplingStrategyType**

diagram	
children	PointSamplingStrategyEnum UserDefinedStrategy
used by	element IfThenSphericalSegmentRuleType/ThenPointStrategy
annotation	documentation The SphericalSegmentPointSamplingStrategyType defines a point sampling strategy for a spherical segment. The strategy is either an enumerated spherical segment strategy or a user defined strategy.

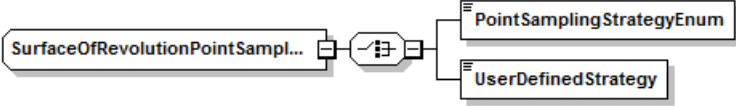
element **SphericalSegmentPointSamplingStrategyType/PointSamplingStrategyEnum**

diagram			
type	SpherePointSamplingStrategyEnumType		
properties	content	simple	
facets	Kind	Value	Annotation
	enumeration	ORTHOGONALGRID	
	enumeration	SPECIFIEDGRID	
	enumeration	STRATIFIED	
	enumeration	HELIX	
	enumeration	POINTS	
annotation	documentation The PointSamplingStrategyEnum element is a commonly used point sampling strategy for a spherical segment.		

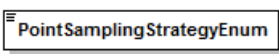
element **SphericalSegmentPointSamplingStrategyType/UserDefinedStrategy**

diagram	
type	xs:token
properties	content simple
annotation	documentation The UserDefinedStrategy element is the name of a point sampling strategy defined by the user.


complexType **SurfaceOfRevolutionPointSamplingStrategyType**

diagram	
children	PointSamplingStrategyEnum UserDefinedStrategy
used by	element IfThenSurfaceOfRevolutionRuleType/ThenPointStrategy
annotation	documentation The SurfaceOfRevolutionPointSamplingStrategyType defines a point sampling strategy for a surface of revolution. The strategy is either an enumerated surface of revolution strategy or a user defined strategy.

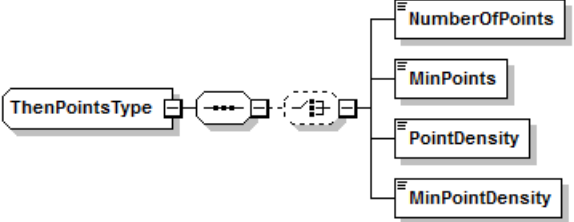
element **SurfaceOfRevolutionPointSamplingStrategyType/PointSamplingStrategyEnum**

diagram			
type	SurfaceOfRevolutionPointSamplingStrategyEnumType		
properties	content	simple	
facets	Kind	Value	Annotation
	enumeration	ORTHOGONALGRID	
	enumeration	BIRDCAGE	
	enumeration	SPECIFIEDGRID	
	enumeration	STRATIFIED	
	enumeration	HELIX	
	enumeration	POINTS	
annotation	documentation The PointSamplingStrategyEnum element is a commonly used point sampling strategy for a surface of revolution.		

element **SurfaceOfRevolutionPointSamplingStrategyType/UserDefinedStrategy**

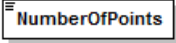
diagram	
type	xs:token
properties	content simple
annotation	documentation The UserDefinedStrategy element is the name of a point sampling strategy defined by the user.

complexType **ThenPointsType**


diagram	
children	NumberOfPoints MinPoints PointDensity MinPointDensity
used by	elements IfThenFeatureRuleType/ThenPoints ElseRuleType/ThenPoints

annotation	documentation The ThenPointsType defines an optional choice of four alternative methods for specifying the quantity of hit points to be used. In the two density alternatives: (1) for curves, the value is in points per unit length in linear units; (2) for surfaces, the value is in points per unit area in area units.
------------	---

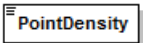
element ThenPointsType/NumberOfPoints

diagram	
type	xs:unsignedInt
properties	content simple
annotation	documentation The NumberOfPoints element is the number of points to be used. A larger or smaller number of points may not be used. This element is in an optional choice.

element ThenPointsType/MinPoints

diagram	
type	xs:unsignedInt
properties	content simple
annotation	documentation The MinPoints element is the minimum number of points to be used. A smaller number of points may not be used. A larger number of points may be used. This element is in an optional choice.

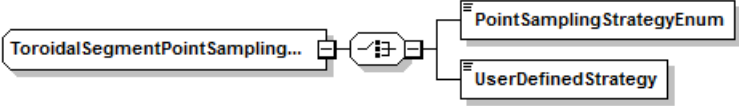
element ThenPointsType/PointDensity

diagram							
type	PositiveDecimalType						
properties	content simple						
facets	<table><tr><th>Kind</th><th>Value</th><th>Annotation</th></tr><tr><td>minExclusive</td><td>0</td><td></td></tr></table>	Kind	Value	Annotation	minExclusive	0	
Kind	Value	Annotation					
minExclusive	0						
annotation	<p>documentation</p> <p>The PointDensity element is the density of points to be used. A larger or smaller point density may not be used, except as is unavoidable because the number of points is an integer. This element is in an optional choice.</p>						

element ThenPointsType/MinPointDensity

diagram	<div><div><div>MinPointDensity</div></div></div>						
type	PositiveDecimalType						
properties	content simple						
facets	<table><tr><td>Kind</td><td>Value</td><td>Annotation</td></tr><tr><td>minExclusive</td><td>0</td><td></td></tr></table>	Kind	Value	Annotation	minExclusive	0	
Kind	Value	Annotation					
minExclusive	0						
annotation	<div>documentation</div> <div>The MinPointDensity element is the minimum density of points to be used. A smaller point density may not be used. A larger point density may be used. This element is in an optional choice.</div>						


complexType **ToroidalSegmentPointSamplingStrategyType**

diagram	
children	PointSamplingStrategyEnum UserDefinedStrategy
used by	element IfThenToroidalSegmentRuleType/ThenPointStrategy
annotation	documentation The ToroidalSegmentPointSamplingStrategyType defines a point sampling strategy for a toroidal segment. The strategy is either an enumerated toroidal segment strategy or a user defined strategy.

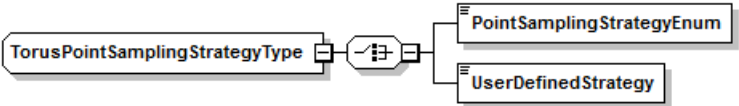
element **ToroidalSegmentPointSamplingStrategyType/PointSamplingStrategyEnum**

diagram	<div><div><div>PointSamplingStrategyEnum</div></div></div>																					
type	SurfaceOfRevolutionPointSamplingStrategyEnumType																					
properties	content simple																					
facets	<table><tr><td>Kind</td><td>Value</td><td>Annotation</td></tr><tr><td>enumeration</td><td>ORTHOGONALGRID</td><td></td></tr><tr><td>enumeration</td><td>BIRDCAGE</td><td></td></tr><tr><td>enumeration</td><td>SPECIFIEDGRID</td><td></td></tr><tr><td>enumeration</td><td>STRATIFIED</td><td></td></tr><tr><td>enumeration</td><td>HELIX</td><td></td></tr><tr><td>enumeration</td><td>POINTS</td><td></td></tr></table>	Kind	Value	Annotation	enumeration	ORTHOGONALGRID		enumeration	BIRDCAGE		enumeration	SPECIFIEDGRID		enumeration	STRATIFIED		enumeration	HELIX		enumeration	POINTS	
Kind	Value	Annotation																				
enumeration	ORTHOGONALGRID																					
enumeration	BIRDCAGE																					
enumeration	SPECIFIEDGRID																					
enumeration	STRATIFIED																					
enumeration	HELIX																					
enumeration	POINTS																					
annotation	<div>documentation</div> <div>The PointSamplingStrategyEnum element is a commonly used point sampling strategy for a toroidal segment.</div>																					


element **ToroidalSegmentPointSamplingStrategyType/UserDefinedStrategy**

diagram	
type	xs:token
properties	content simple
annotation	documentation The UserDefinedStrategy element is the name of a point sampling strategy defined by the user.


complexType **TorusPointSamplingStrategyType**

diagram	
children	PointSamplingStrategyEnum UserDefinedStrategy
used by	element IfThenTorusRuleType/ThenPointStrategy
annotation	documentation The TorusPointSamplingStrategyType defines a point sampling strategy for a torus. The strategy is either an enumerated torus strategy or a user defined strategy.

element **TorusPointSamplingStrategyType/PointSamplingStrategyEnum**

diagram			
type	SurfaceOfRevolutionPointSamplingStrategyEnumType		
properties	content	simple	
facets	Kind	Value	Annotation
	enumeration	ORTHOGONALGRID	
	enumeration	BIRDCAGE	
	enumeration	SPECIFIEDGRID	
	enumeration	STRATIFIED	
	enumeration	HELIX	
	enumeration	POINTS	
annotation	documentation The PointSamplingStrategyEnum element is a commonly used point sampling strategy for a torus.		

element **TorusPointSamplingStrategyType/UserDefinedStrategy**

diagram	
type	xs:token
properties	content simple
annotation	documentation The UserDefinedStrategy element is the name of a point sampling strategy defined by the user.

Bibliography

[1] SAE AS9102a (2004-01), *Aerospace First Article Inspection Requirement*

[2] Walmsley, Priscilla., 2002. *Definitive XML Schema*. Prentice Hall, Upper Saddle River, NJ, USA.