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

Part 1: Overview and Fundamental Principles Version 2.0

QIF Version 2.0

ANSI/QIF Part 1–2014

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

Contents

Figures

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) was developed by domain experts from the manufacturing quality community representing a wide variety of industries and quality measurement needs. Contributors to this version 2.0 of the QIF standard include:

- Capvidia
- Honeywell Federal Manufacturing and Technology
- Lockheed Martin Missiles and Fire Systems,
- Metrosage
- Mitutoyo America
- National Institute of Standards and Technology
- Origin International Inc

This document was written by the QIF Working Group, and given final approval for ANSI review by the DMSC's Quality Measurement Standards (QMS) Committee. More information about DMSC's QIF effort can be found at www.qifstandards.org.

The QIF standard, version 2.0, 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 Definition (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

Parts 1 and 2 describe the overview and central concepts of the QIF standard. Parts 3 through 8 describe information models for the six *application areas* of QIF, namely MBD, Plans, Resources, Rules, Results, and Statistics.

The inaugural QIF standard, version 1.0, was published in 2013. This document is a component of the second release of the QIF suite of standards, denoted version 2.0. The QIF version 2.0 documents cancel and replace all documents of version 1.0. QIF version 2.0 is solely a product of the DMSC and its committees and working groups.

Each major release of the QIF standard is composed of several *Parts* documents. Individual *Parts* are designated with the version number of the major QIF revision, even when the document is new. QIF version 2.0 includes revisions of the version 1.0 documents Part 1 (Overview), Part 2 (QIF Library), Part 3 (Plans) and Part 4 (Results), and 4 new Parts that did not exist in version 1.0. Also, QIF Part 3 – 2013 was redesignated as Part 4 and its name was changed from QMPlans to QIF Plans. QIF Part 4 – 2013 was redesignated as Part 7 and its name was changed from QMResults to QIF Results.

This version of Part 1, designated QIF Part 1 – 2014, is a revision of the original standard QIF Part 1 – 2013. It's scope was expanded to cover the new application areas of MBD, Resources, Rules, and Statistics. It's content was revised to reflect changes made to the QIF information model.

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 data model. The html files facilitate viewing the complete data 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 data model description using mouse clicks.

Introduction

This Quality Information Framework (QIF) standard defines an integrated set of information models which enable the effective exchange of metrology data throughout the entire manufacturing quality measurement process – from product design to inspection planning to execution to analysis and reporting. This part of the QIF standard introduces the purpose and design approach behind QIF, as well as its content. The QIF information models are contained in files written in the XML Schema Definition Language (XSDL). The models for Version 2.0 consist of six application schema files plus a library of schema files containing information items used by all applications. The Library is described in Part 2 of this standard. Part 3 of this standard defines the Model Based Definition application model which deals with CAD data plus product manufacturing information (PMI). Part 4 defines the QIF Plans application model, which deals with plans for quality measurement. Part 5 defines the QIF Resources application model, which describes hardware and software resources used for inspection. Part 6 defines the QIF Rules application model which describes practices to be used in an inspection. Part 7 defines the QIF Results application model which deals with the results of part measurements. Part 8 defines the QIF Statistics application model which conveys analysis of multiple part inspections.

The QIF models include quality characteristics and measurement features as defined in the ASME Y14.5 specification and the Dimensional Measuring Interface Standard (DMIS). The QIF standard covers a wide variety of use cases including dimensional metrology inspection, first article inspection, reverse engineering, and discrete quality measurement.

Quality Information Framework (QIF) − An Integrated Model for Manufacturing Quality Information − Part 1: Overview and Fundamental Principles Version 2.0

1 Scope

1.1 Contents of this document

This document, Part 1, describes the general content and structure of the entire QIF information model. It describes the highest level data structures of QIF, that are expanded in Parts 2 through 8 using data dictionaries and XML schema files. All QIF XML schema files are bundled into a single compressed folder file called "QIF_2.0_XMLSchemaFiles.zip" that can be found at www.qifstandards.org.

This Part also describes practices for forming QIF instance files, called "documents", that support quality workflow scenarios. Its focus is to show how the QIF information model, and data formed into XML instance files, support the entire scope of model based definition manufacturing quality workflow. It describes how the information model is partitioned among the XML schema files and contains all terms used in the other seven Parts.

The purpose of this Part is to orient potential users of QIF to the organization of the information model to make their study of the details more rewarding and efficient. It should also help solution providers and users to evaluate QIF for their uses, without needing to go to the lowest technical details of the XML schemas. The information model narrative focuses on the approach to modeling the core data structures of QIF, which model the content of ANSI/ASME Y14.5M-1994, and the plans and results data elements defined in Dimensional Measuring Interface Standard (DMIS) 5.2. The material on XML practices describes consistent design practices to be used by QIF working groups who will be designing new schemas. It should also help data processing experts to write software that writes and reads manufacturing quality data using the XML schemas.

1.2 Scope of the QIF Version 2.0 information model

Figure 1 shows a high level view of the QIF information model for version 2.0 standardization. At the core of the QIF architecture is the reusable QIF library which contains definitions and components that are referenced by the application areas, thereby ensuring interoperability and extensibility. Around the QIF library core Figure 1 shows the six QIF application area information models, MBD, Plans, Resources, Rules, Results, and Statistics. The "QIF Execution" model is a placeholder for a future DMSC standard, that is not a part of QIF 2.0. Each QIF application model reuses the QIF Library. The order of generation of QIF data in an enterprise generally proceeds clockwise around the diagram, beginning with QIF MBD and ending with QIF Statistics. Use of the QIF information model does not place any requirements on a user's workflow architecture.

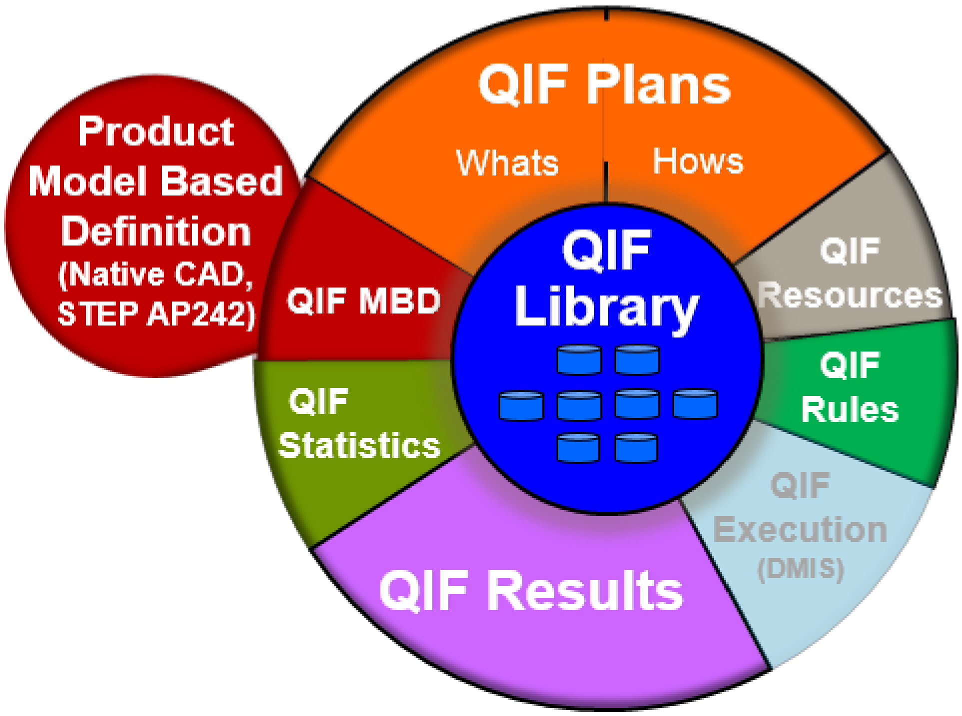


Figure 1 – QIF version 2.0 information architecture

The flow of QIF data starts with generation of CAD + PMI data exported as QIF MBD application data. Quality planning systems import the MBD and generate Plans (whats), then import Resources and Rules information and export Plans (whats and hows). Programming systems import Plans to generate DME-specific programs, or general instructions to guide inspection. Dimensional measurement equipment executes programs and evaluates characteristics of a single manufactured part or assembly and export the measurements as Results. Analysis systems, typically performing statistical process control, import single parts Results and generate analysis of multiple part batches as QIF Statistics data.

Users of the QIF information model are not required to implement the entire model. Any of the six application models may be used singly for exchange of quality data between software systems. Further, other data models and exchange formats can coexist in an enterprise with QIF data.

2 Conformance

Software programs that implement this specification to write QIF XML instance files must:

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

Software programs that implement this specification to read QIF instance files must:

- be able to read any valid QIF 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 Y14.5-1994 (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 Y14.41-2012, *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

4 Terms and definitions

For the purposes of this document, the following terms and definitions apply

4.1 Terms defined in ANSI Y14.5-2009

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

4.1.2 datum target

the designated points, lines, or areas that are used in establishing a datum

NOTE Datum targets are used in establishing a datum reference frame.

4.2 Terms defined in ANSI/DMIS 105.2-2009, Part 1

4.2.1 actual

referring to features or tolerances, the actuals are the results computed from measured or constructed features

NOTE Actuals are determined by the measuring device and do not exist until a measurement or construction has occurred.

NOTE Referring to parts and assemblies, the actuals are physical instances called actual components.

4.2.2 dimensional measuring equipment (DME)

a class of equipment used to inspect parts and evaluate dimensions and tolerances

NOTE DME includes, but is not limited to, coordinate measuring machines, video inspection equipment, optical comparators, robotic measuring devices, theodolites, photogrammetry, and laser-based measuring devices.

4.2.3 nominal

referring to a feature or tolerance, this is the "as-designed" value defined in the product definition of the part

NOTE The nominal is known prior to measurement and is the value to which the actuals are compared for computing out of tolerance conditions.

4.2.4 part coordinate system (PCS)

a datum reference frame associated with the part to be measured

4.3 Terms defined in XML Schema Part 0: Primer Second Edition, W3C Recommendation 28 October 2004

4.3.1 attribute

information represented as an XML attribute in an instance file, usually conforming to an *attribute* declaration in an XML schema

4.3.2 complexType

a value type that has elements or attributes

4.3.3 element

information represented as an XML element in an instance file, usually conforming to an *element* declaration in an XML schema

4.3.4 instance file

a file containing an information set intended to conform to an XML schema

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

4.4.1 attribute information item

information modeled using an *attribute* declaration in an XML schema

NOTE often shortened to "attribute".

4.4.2 element information item (often shortened to "element")

information modeled using an *element* declaration in an XML schema

4.4.3 enumeration

a term indicating that a single indivisible value follows

4.4.4 extension

a term in a complexType definition indicating that the type being defined is derived from a more general type

NOTE Extension is the XSDL mechanism used in QIF for building hierarchies of complex types. XSDL also provides for restrictions of complex types, but that is not used in QIF.

4.4.5 key
a constraint requiring that selected data fields exist and be unique

NOTE In XSDL, a key may require that a combination of data fields be unique, but QIF never uses more than a single data field.

4.4.6 keyref
a constraint requiring that there is a match between two sets of values in an instance file

4.4.7 schema (or XML schema)
a complete information model in the XML schema definition language

NOTE A schema is defined by one or more schema files.

4.4.8 schema document (or schema file)
a file containing a well-formed XML schema declaration

NOTE Such a file does not necessarily contain a complete information model; it may reference other schema files needed to make the model complete.

4.4.9 simple type
a value type that is indivisible or is a list of indivisible items
number example: 10
string example: 10 Downing Street
list example: 10 34 -7

NOTE In XSDL, *union* type is also simple, but it is not used in QIF.

4.4.10 string
a data type that is a sequence of Unicode characters

4.4.11 token
a string in which there is no leading or trailing white space, and the only white space occurring between visible characters is a single space character

4.5 Terms defined in the QIF standard

4.5.1 action
a plan element that gives information about what to measure or validate

4.5.2 action group
a plan element that organizes sets of actions

NOTE An action group can be an ordered group, unordered group, one-of group, partially ordered group, or pick-some group.

4.5.3 action method
a prescription of how an action is to be performed

NOTE Actions with action methods form the core of a measurement plan.

4.5.4 actual component
a physical instance of a component

NOTE In QIF the design of a wheel is a part; the design of the wheel placed at the front right of the design of a car is a component; the front right wheel of your car is an actual component.

4.5.5 actual component set
a set of actual components

4.5.6 application area
one of the six QIF workflow interface types, i.e., MBD, Plans, Resources, Rules, Results, and Statistics

4.5.7 aspect
one of four categories used to incrementally accumulate characteristic or feature data based on data source and data shareability

NOTE In QIF 2.0 there are four aspects of characteristics and features: definition, nominal, actual, and item.

definition aspect
the aspect of feature or characteristic nominal data categorized by its shareability among different features or characteristics

nominal aspect
the aspect of feature or characteristic nominal data particular to an individual feature or characteristic

item aspect
the aspect of feature or characteristic incremental data particular to an individual measurement of a feature or characteristic (in QIF 1.0 referred to as the "instance aspect")

actual aspect
the aspect of feature or characteristic data particular to the results of the measurement of a feature or characteristic

4.5.8 assembly
a number of parts or combination thereof that are joined together to perform a specific function and subject to disassembly without degradation of any of the parts

NOTE In QIF, assembly means the design of an assembly, and a physical instance of an assembly is called an actual component.

4.5.9 assembly path
a sequence with an id of the ids of components

NOTE An assembly path shows where in an assembly design a specific instance of a part design or a subassembly design is located

4.5.10 assignable cause
those causes of variation in a process which are not random and have some source which can be determined and perhaps eliminated

4.5.11 attribute characteristic
a characteristic described using attribute data, or using data that do not have numerical values

EXAMPLE color, malleability

4.5.12 attribute data
a result from a characteristic or property that is appraised only as to whether it does or does not conform to a given requirement (for example, go/no-go, accept/reject, pass/fail, etc.) [AS9102a]

4.5.13 bias
a measure of a gage's tendency toward specific values when compared to a master value

4.5.14 bill of characteristics (BoC)
a list of all the characteristics applied to a product.

4.5.15 boolean condition
a statement which can be unambiguously evaluated as true or false

EXAMPLE As applied in QIF Rules, "The feature is a cylinder" would evaluate to "true" if and only if the feature in question is a cylinder feature.

4.5.16 capability
the measure of a process's stability and centralization against a nominal value and tolerance values, also, the ability of a process to produce acceptable parts

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

4.5.18 characteristic item
a tolerance or specification applied to a feature or product that needs verification

4.5.19 checked
refers to an actual feature or characteristic being measured directly or being constructed from previously measured or constructed data

4.5.20 clipping plane
a bounding plane surface which abbreviates the intended display of data to that portion which lies on one or the other side of the plane

4.5.21 component
an instance of a part or an assembly aligned to its parent's space

4.5.22 composite feature
a feature composed of two or more sub-features which act as a functional group and to which shared characteristics may be applied

4.5.23 constructed feature
a feature that is computed from other features

NOTE Contrast to a measured feature, which is computed from measured point data.

4.5.24 control limits
statistical limits that represent boundary conditions of a process that is in control

NOTE Processes that are out of control are said to have unnatural causes or assignable causes of variation.

4.5.25 control points
a set of points that determine the shape of a NURBS object

4.5.26 control polygon
a polygon formed by the control points of a NURBS object

4.5.27 corrective action
a countermeasure that can be applied to an assignable cause of variation in order to reduce the likelihood of recurrence

4.5.28 corrective action plan
information related to the establishment of lists of assignable causes of variation and associated corrective actions in the manufacturing process

4.5.29 datum definition
definition of a datum label and optionally, its association with datum targets or feature instances

4.5.30 entity
the basic unit of information in a file

NOTE The term applies to single items which may be individual elements of geometry, collections of annotations to form dimensions, or collections of entities to form structured entities.

4.5.31 evaluation
refers to the process by which the status of a characteristic actual is determined from nominal, actual and specification limit information

4.5.32 event
an occurrence, usually unplanned, which may have an effect on the outcome of a measurement or inspection operation and which should be recorded

4.5.33 external product definition
a product shape definition not included in a QIFDocument
EXAMPLE a PDF file, a drawing, or a physical part.

4.5.34 gage repeatability and reproduceability (gage R&R)
a source of measurement variation based on the ability of a gage to repeat measurements on the same parts and the ability of two or more inspectors using this gage to achieve the same measurement results

4.5.35 generatrix
a curve to be swept to generate an extruded surface, or revolved to generate a surface of revolution

4.5.36 generic feature
a feature which can be referenced by a user-defined characteristic

NOTE Usually, this is a portion of the surface of a part that is not correctly described by any other feature type.

4.5.37 geometric
related to shape information: e.g., points, curves, surfaces, and volumes

4.5.38 geometric characteristic
a concept characterizing the size, form, orientation or location of a feature or of a component of a feature

EXAMPLE diameter, flatness, parallelism, or position.

4.5.39 inspection
a measurement of characteristics on a physical part to determine whether the features are within allowed tolerance, commonly in order to accept or reject the part

4.5.40 inspection traceability

information about the circumstances of a quality measurement process

4.5.41 internal product definition
a product shape definition contained in a QIFDocument

4.5.42 item
instance (deprecated)
an individual occurrence of an object

4.5.43 key characteristic
a characteristic of a feature, material, process, or part whose variation has a significant influence on product fit/function, safety/compliance, performance, service life, or manufacturability

NOTE A key characteristic can be identified by a designator and have a criticality class.

4.5.44 knot vector
an increasing sequence of real numbers which divides the parametric space of a NURBS into intervals (called spans)

NOTE The number of knots is equal to the number of control points plus the order. The knot vector is a uniform one if all knots are equally spaced and of multiplicity one.

4.5.45 linearity
the ability of a gage to accurately measure across a range of values

4.5.46 manufacturing traceability
information about the circumstances of a manufacturing process

4.5.47 measurand
an object, quantity, property, or condition to be measured for a specific purpose

NOTE Two examples of a measurand are the measurement of a shape feature to evaluate a specified characteristic (for example tolerance) and the measurement of a shape feature to establish a datum (for example primary datum) within the context of a datum reference frame. One could measure the same feature differently or apply a different substitute feature data fitting algorithm.

4.5.48 measure feature method
an action method for measuring a feature

4.5.49 measurement
an estimate of a dimension associated with a feature or features on a physical part generated using a physical device

4.5.50 measurement device
a hardware element that performs a measurement, or is used by an operator to perform a measurement

NOTE A measurement device will typically have an accuracy statement as well as calibration information. Measurement devices include CMMs and calipers.

4.5.51 measurement plan
a complete plan that contains information on what and how to measure

4.5.52 measurement resource
a class of things that facilitate or perform measurements

NOTE Things that facilitate measurement include fixtures, sensors, tooling, rotary tables, software, tapped hole location gages, chemical solutions, ovens, environmentally controlled rooms, etc.

4.5.53 non-dimensional quality data
data expressed as either attributes or variables, e.g., number of non-conformities like burrs or dents, color of paint, etc.

4.5.54 normal
perpendicular to a surface and pointing away from the surface material

4.5.55 normal vector
a unit vector perpendicular to a surface in 3 dimensions or a unit vector perpendicular to a curve in 2 dimensions

4.5.56 notable event
a description of a planned or unplanned event that inspection processes should monitor and should record if it occurs during inspection

4.5.57 note
an explanatory or descriptive statement in natural language

NOTE In QIF, notes provide information in addition to that which is formally modeled.

4.5.58 noted event
an event occurring and reported during inspection

4.5.59 part
one item, or two or more items joined together, that is not normally subject to disassembly without destruction or impairment of designed use

NOTE In QIF, part means the design of a part, and a physical instance of a part is called an actual component.

4.5.60 plan element
an action or an action group

NOTE A combination of actions and action groups can be structured in a directed hierarchical tree of actions.

4.5.61 plan note
descriptive information that applies to an entire measurement plan

4.5.62 plan root
the top level plan element (an action or action group) of a measurement plan

4.5.63 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.

4.5.64 process variation
the variation of a process in achieving the same characteristic values across time

4.5.65 product
a generic term for a part or an assembly

4.5.66 product and manufacturing information (PMI)
non-geometric attributes in 3D Computer Aided Design/Manufacturing/Inspection/Engineering (CAD/CAM/CAI/CAE) systems necessary for manufacturing product components or subsystems

NOTE PMI may include geometric dimensions & tolerances (GD&T), 3D annotation (text) and dimensions, surface finish, and material specifications.

4.5.67 production
a manufacturing process or operation designed to produce goods

4.5.68 qualification
a check on or the refinement of the calibration of a measurement device performed using a known artifact such as a tooling ball or gage block

4.5.69 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 may 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".

4.5.70 sampling method
a statistical method of grouping manufactured products for measurement

NOTE Key concepts include sample size (how many) and sampling frequency (how often).

4.5.71 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.

4.5.72 set

refers to an actual feature or characteristic being set to its nominal without any measurement taking place

4.5.73 stability

the ability of a gage to arrive at the same measurements against a master value over time

4.5.74 standard deviation

a measure of the dispersion or frequency distribution around a population mean or average

4.5.75 statistical study plan

information, defined in the QIF Statistics model, that provides a method for establishing the quality approach and criteria for data studies associated with measurement in manufacturing

NOTE This plan ensures the expectation that correct quality control information will be supplied with each manufactured lot.

4.5.76 statistical study results

information that contains the actual summary data that provides an overview of measured product quality

4.5.77 traceability

information about the circumstances of a quality measurement process or a manufacturing process

NOTE QIF defines five types of traceability. None of these is the classical traceability of measurements from an international standard through intermediate steps to a specific measurement.

4.5.78 trimming contour

a 3D contour which is used for trimming a surface

4.5.79 wire-frame

a method of geometric modeling in which a two- or three-dimensional object is represented by object edges

4.5.80 work instruction

information that provides instructions about actions or action methods to be used in executing a measurement plan

5 Symbols and abbreviated terms

ANSI	American National Standards Institute
ASCII	American Standard Code for Information Interchange
ASME	American Society of Mechanical Engineers
BREP	Boundary Representation
CAD	Computer-Aided Design
CAIPP	Computer-Aided Inspection Process Planning
CAM	Computer-Aided Machining or Computer-Aided Manufacturing
CAX	Computer-Aided Technologies
CMM	Coordinate Measuring Machine
CoP	Cloud of Points
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
MBD	Model Based Definition
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 Overview of the Quality Information Framework (QIF) information model

6.1 Purpose

The goal of the QIF specification is to facilitate interoperability of manufacturing quality data between system software components. Solving the metrology interoperability problem will benefit manufacturers by avoiding wasted resources spent on non-value-added costs of translating data between the different components of manufacturing quality systems. Users should gain flexibility in configuring quality systems and in choosing commercial components, and achieve effortless and accurate flow of data within their factory walls as well as with suppliers and customers. Solution providers should be able to eliminate their efforts previously spent in data translations, and there should be increased opportunities to sell their products and to improve and expand the features of their solutions.

System wide interoperability is achieved by partitioning the information model between a QIF Library of common, reusable components, and six information models for unique application areas. The reusable library components are referenced throughout the comprehensive quality information model thereby ensuring interoperability and extensibility between any data producer and consumer that implements the QIF formats in their software.

DMSC's goal is to write the QIF specifications such that conformity of commercial software products can be assessed by a manufacturer or supplier (first party), a user or purchaser (second party), or an independent body (third party). The ability of developers to test against conformance criteria, and of users to evaluate products for conformance, are key to establishing widespread interoperability of commercial off the shelf software solutions.

6.2 Model based definition manufacturing quality workflow

The scope of the QIF 2.0 information model has expanded beyond the original content of version 1.0, with its emphasis on dimensional features and characteristics data, and the application areas of inspection plans and inspection result reports. QIF 2.0 now includes information models for six application areas, as shown in Figure 1.

Figure 2 shows a model-based quality workflow activity diagram and the use of QIF formats to convey information between computer-aided quality processes. The scope of QIF version 2.0 is all manufacturing design and quality information required to assess product quality, and also to improve manufacturing processes and product design. The work flow model shows five major activities of a quality metrology enterprise:

- Define Product
- Determine Measurement Requirements
- Define Measurement Process
- Execute Measurement Process
- Analyze & Report Quality Data

Activities may export and/or import quality information that can be formatted according to the QIF information model and XML encoding rules. The diagram does not show activities that generate manufacturing process information or that implement a manufacturing execution system. QIF information is conveyed in a product-neutral format and is modularized into six application areas plus the QIF Library. These features of QIF facilitate efficient flow of enterprise quality data in a way that does not specify or constrain a user's system architecture.

The *Define Product* activity generates a model-based definition of a part that can support an enterprise's digital product verification. The product definition contains geometry information plus semantically linked product manufacturing information (PMI). PMI commonly includes geometric dimensioning and tolerancing (GD&T) information, key characteristic criticalities, material, surface texture, roughness, color and hardness. GD&T includes associations between geometric elements of the product and dimensions, tolerances, and datums. The product definition is expressed using the QIF MBD information model.

The *Determine Measurement Requirements* activity imports QIF MBD information or its equivalent expressed in another format or formats. Based upon enterprise quality requirements and/or manufacturing process knowledge, measurement requirements for a part are generated as a set of measurement criteria also known as a bill of characteristic instances (BOC). A characteristic instance is typically a tolerance or specification applied to a feature or product that needs verification. The plan may also specify the measuring sequence and resources to be used, but is not required to do so. This BOC constitutes a high level quality plan of "what" needs to be inspected or verified, expressed as a QIF Plans (whats) information packet.

The activity that generates QIF Resources information is not shown, but can be called *Define Measurement Resources*. This activity defines enterprise hardware and software resources available, either serial-number specific, or generic, that can be harnessed to meet inspection requirements for individual part features. The QIF Resources data format can be used to specify resources required in an inspection plan, or the resources actually used in an inspection. The scope of the QIF Resources application model includes:

- definition of resources both hardware and software,
- DMEs,
- application software,
- fixtures,
- go-no-go gages,
- manual instruments.

The activity that generates QIF Rules is not shown, but can be called *Define Measurement Rules*. This activity generates inspection practices required by an enterprise to be used in-house or by contractors. QIF Rules data defines, for each possible feature type on a part, the information elements required to fully specify and constrain the measurement on that feature type. The information elements include things like measurement point density, measurement point pattern, and feature fitting algorithm. The QIF standard defines the generic format to express enterprise Rules, but does not contain specific rules. Information areas that are in scope include:

- product measurement point number or density and sampling method for each product feature type
- feature fitting algorithm for each feature type
- rule ID and corporate ownership.

The *Define Measurement Process* activity inputs resource and metrology knowledge, and the QIF Plans (what), and generates additional instructions on “how” to inspect or verify the bill of characteristic instances. The completed inspection plan is output as QIF Plans (whats and hows). The scope of the QIF Plans 2.0 application model includes:

- dimensional product information, e.g., geometric features, measurement features, nominal dimensions, measurement features, and tolerance values
- non-dimensional product information, e.g., product IDs, customer information, key contact, temperature, and roughness
- product characteristics
- traceability values and pointers
- work instructions
- CAD entity relationships.

The downstream activity *Execute Measurement Process* activity imports the QIF Plan, and if needed generates a detailed resource specific inspection program. The programs are machine-level measurement programs, formatted according to DMIS or some other measurement programming language, that provide equipment level commands to specific coordinate measuring machine (CMM) control units, to collect point data, fit features to data, and output feature and characteristic data. The workflow shows the export of non-QIF format subsequently translated according to the QIF Results information model. Measurement processes that adopt QIF will likely export results directly without translation. Measurement Execution can also include software solutions that issue instructions to human operators using calipers, go/no-go gauges, and specialized inspection equipment, and generate results data. Actual measurement values may be numerical or non-numerical. Measurement results may include not only raw measurement values, but also summary statistical or derived results (e.g., cylinder radius with standard deviation). Measurement results may also include description of the algorithmic means (e.g., least squares) by which the derived results are calculated. All necessary nominal (as designed) target values may also be included to allow reanalysis. Any other information relevant to the measurements is also in scope. This includes information called inspection traceability, which includes the shift, the equipment operator’s name, a description of the item measured, the date and time of the measurement, etc.

Finally, the measurement results for two or more parts are collected, analyzed, and reported by the activity *Analyze & Report Quality Data*. The output, expressed using the QIF Statistics model, is generally an analysis of a multi-part batch. QIF Statistics is designed to carry information to transport statistical quality control plans, corrective action plans and detailed summary quality statistics. It builds on the QIF Results framework through supporting multi-part measurement results that can apply to a number of quality study types beyond single or first article inspection. It is designed to haul information in an unambiguous form for pre-production, capability, and production quality studies. In addition it supports the full extent of measurement systems analysis studies including Gage R&R.

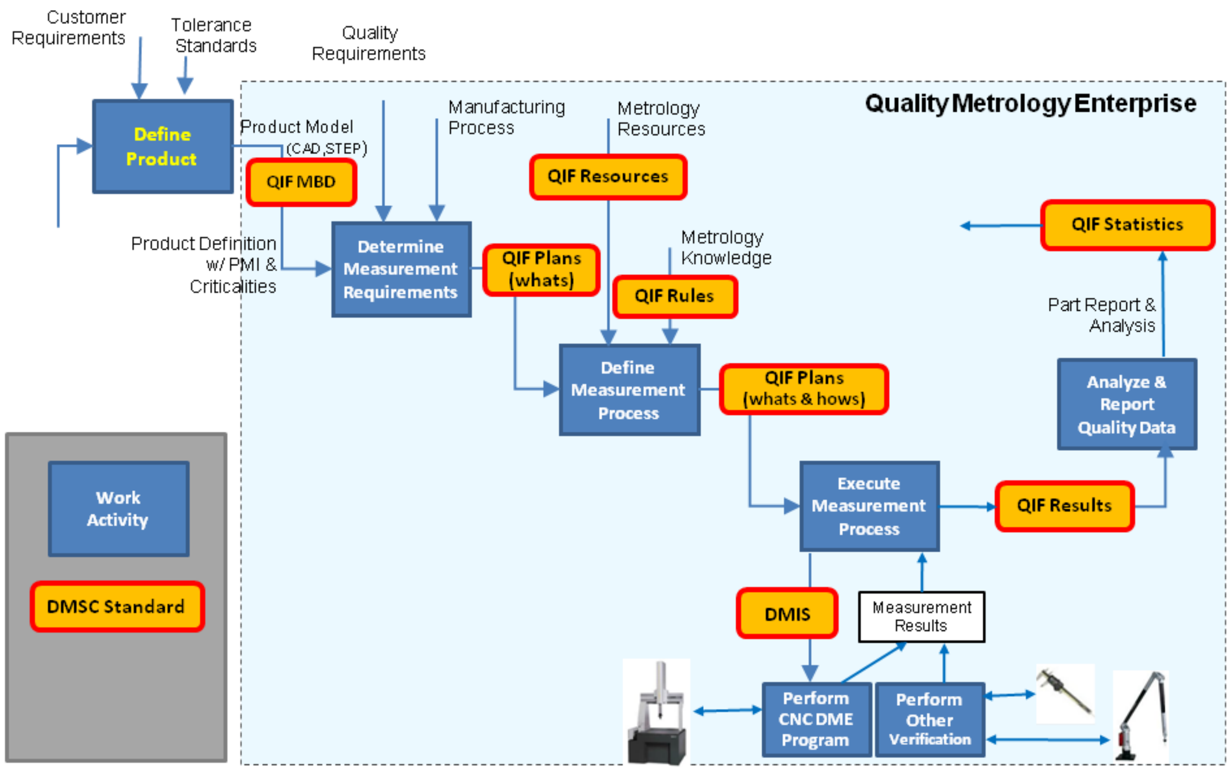


Figure 2 – QIF Model-Based Quality Workflow

Quality information generated in QIF format can be used as input by many other quality and manufacturing management components not shown in Figure 2, including, but not limited to, first article inspection plan and report generation, statistical process control (SPC), materials resource planning (MRP), measurement systems analysis (MSA), manufacturing execution systems (MES), and computer aided manufacturing (CAM).

The digital interface between Execute Measurement Process and the DME (dimensional measurement equipment) has been satisfied by the Dimensional Measuring Interface Standard (DMIS), ANSI/DMIS 105.2 Part 1-2009. DMIS can also be used as a numerical control part program for DMEs such as coordinate measuring machine (CMM).

An example of user defined flow of QIF information not shown in Figure 2 is the direct use by a human inspector of a QIF Plans file to inspect a part. The operator makes the decisions of

Measurement Programming like selecting the inspection device (e.g., caliper, fixtures, go-no go gages) and generating the inspection instruction details (e.g., number of points, placement of measurements).

6.3 QIF design requirements

There are three categories of requirements on the content of the QIF information model, and on how the data are encoded for exchange:

- **Business case functional requirements**. The design of the QIF information model is driven by the functional requirements of activities that import, process, and export manufacturing quality data. Requirements are expressed via natural language rules, scenarios, use case notation, identification of specifications and/or standards, and examples. The DMSC began with a baseline requirement to model ANSI/ASME Y14.5-1994, and the plans and results information in DMIS 5.2. The requirements list is evolving as the members of the quality community identify more workflow requirements. Functional requirements involve engineering data and workflow details, as well as business case justifications and requirements.

- **Interoperability requirements**. This is primarily the requirement on the QIF standard, that developers from different companies should be enabled, without cooperation, to develop software applications that will successfully exchange QIF data packages. This requirement is essential because solutions are developed globally by diverse developers, but also because QIF information must flow between companies, between original equipment manufacturers and contractors, and between primary vendors and their subcontractors. Interoperability requirements are met primarily through complete and accurate semantic annotations embedded within the XML schema files, and by publishing specifications and usage documents. QIF also includes data structures that allow writers of instance files to insert customized text data, which should only be used when their data does not match a defined QIF enumerated data type.

- **Computational requirements**. These are good practices of computer science that lead to efficient code and efficient use of computation resources. Examples include the use of class inheritance, and schema design practices that minimize the size of instance files.

6.4 QIF manufacturing functional requirements

Requirements that have been validated by inspection and examples in QIF 2.0 include ASME Y14.5-1994 and DMIS 5.2. Functional requirements met in QIF 2.0 include the following:

- Information requirements for QIF 2.0 encompass the principles of geometric dimensioning and tolerancing as described in ANSI/ASME Y14.5-1994, as well as workflow practices and quality management functions. Most of the requirements of the 2009 version are also met. DMSC plans to meet the full requirements of the 2009 version in the near future.

- QIF 2.0 encompasses all planning and results information defined in DMIS 5.2. QIF 2.0 also includes a small amount of tool and sensor data.

- QIF 2.0 instance files support all information defined for first article inspection reports as defined in the standard AS9102a [1].
- The neutral data format specifications are accompanied by fully defined semantics derived where applicable from other standards like DMIS 5.2, and AS9102a. The semantics ensure that data cannot be misinterpreted between sender and receiver of QIF instance files.
- Inspection results data formatted as QIF Results can be used both for a reverse engineering process where actual measurement data is stored without the presence of nominal information, and for a conventional measurement process where inspection is planned using nominal part feature data.
- QIF facilitates traceability of quality results to inspection and measurement processes, including identification of measurement devices and operators, software applications, and CAD models.
- QIF facilitates traceability of quality results to manufacturing processes by maintaining links, so that inspection data can be used to monitor, control, and improve manufacturing processes. The QIF scope includes support of manufacturing process and product validation testing.
- QIF Results data can be written to facilitate re-analysis of measured point data.
- QIF data supports quality systems based on model based definition, as well as systems that implement 2D drawing-based processes.

6.5 QIF information model design guidelines

A QIF design principle is to follow a decoupled normalized relationship model. As such, many relations between data elements (also known as instanced types) in the QIF schemas are made using identifiers rather than allowing a parent type to directly contain a child type (as you would find in a strictly hierarchical model). Each “object” or “instanced type” that needs to be referenced by another type is given a unique identification designator (id). The referencing object may then establish a relation by calling out the id rather than redefining the entire data element within itself. By decoupling the child from the parent we introduce the ability to reuse components of the relationship without duplicating definitions. This concept of decoupling is one of the pillars of QIF extensibility. It also leads to reducing instance file size compared to purely hierarchical design, and may lead to better data integrity because the risk of duplicating errors is reduced.

The QIF models represent measurement feature and characteristic objects with four aspects: item, definition, nominal, and actual. The relationships between data objects of each aspect type in an instance file are implemented using a relationship scheme described in more detail in section 6.7. Because the scope of QIF covers the entire workflow of enterprise quality systems, the aspects allow QIF instance files to describe nominal and measured GD&T quality data, as well as express the relationships and other data generated during the workflow of planning, execution, reporting, and analysis.

We will use the terms “data object” and “object” to mean a grouping of information in a QIF XML instance file, defined by elements of the QIF information model.

6.6 Overview of XML schema file modularity

A complete model built using XML Schema Definition Language (XSDL) is called a *schema*. However, the complete model may consist of information items from several different *schema* files. Typically, the complete model will be defined using a top level (or root) *schema* file which will use subordinate *schema* files. In XSDL, using definitions in other files is indicated by an *include* directive. The top level file *include*s subordinate files, and those subordinates may *include* other subordinates.

For example, **QIFDocument** is a complete application *schema*. The top-level *schema* file is QIFDocument.xsd. It includes the primary *schema* file for each of the six QIF application areas, and each of those *include*s subordinate files collected in the QIF Library. The net effect is that the **QIFDocument** *schema* contains everything defined in all the QIF *schema* files.

An XML data file conforming to an XML *schema* is called an instance file. Every instance file conforming to the QIF model will have **QIFDocument** as the root of a hierarchy of information.

The XML schema file hierarchy of QIF is shown in Figure 3.


```
QIF (directory)
   QIFApplications (directory)
      QIFDocument.xsd
      QIFMeasurementResources.xsd
      QIFPlan.xsd
      QIFProduct.xsd
      QIFResults.xsd
      QIFRules.xsd
      QIFStatistics.xsd
   QIFLibrary (directory)
      Auxiliary.xsd
      Characteristics.xsd
      Expressions.xsd
      Features.xsd
      GenericExpressions.xsd
      Geometry.xsd
      IntermediatesPMI.xsd
      Primitives.xsd
      PrimitivesPD.xsd
      PrimitivesPMI.xsd
      Statistics.xsd
      Topology.xsd
      Traceability.xsd
      Units.xsd
      Visualization.xsd
```

Figure 3 – QIF XML schema directory structure

The QIF Library is modularized by grouping related *type* and *element* definitions together in a *schema* file. Some of the *schema* files in the QIF Library serve multiple applications, and some serve only a single application:

- Expressions.xsd and GenericExpressions.xsd serve only QIFRules.
- Auxiliary.xsd, Geometry.xsd, Topology.xsd, and Visualization.xsd serve only QIFProduct.
- Statistics.xsd serves only QIFStatistics.

The QIF application models are each described in a separate Part document (QIF Parts 3 through 8). Part 2 of the QIF standard describes the contents of each *schema* file in the QIF Library.

6.7 Data structures

6.7.1 The QIFDocument element

QIFDocument is the highest level element of all QIF instance files. Its structure is illustrated in Figure 4. Many different sorts of instance file may be prepared. A file may include information for only a single application area, or it may combine applications. For example, a QIF Rules instance file will typically have only identifying information and a **Rules** *element* in the **QIFDocument**. On the other hand, if a product has been modeled in the **Product** *element* of a **QIFDocument**, an instance file containing a plan for that product is likely to have the same **Product** *element* and a **Plan** *element* in the **QIFDocument**.

A QIF Statistics instance file is likely to have at least a **Product** *element*, a **MeasurementsResults** *element*, and a **Statistics** *element*.

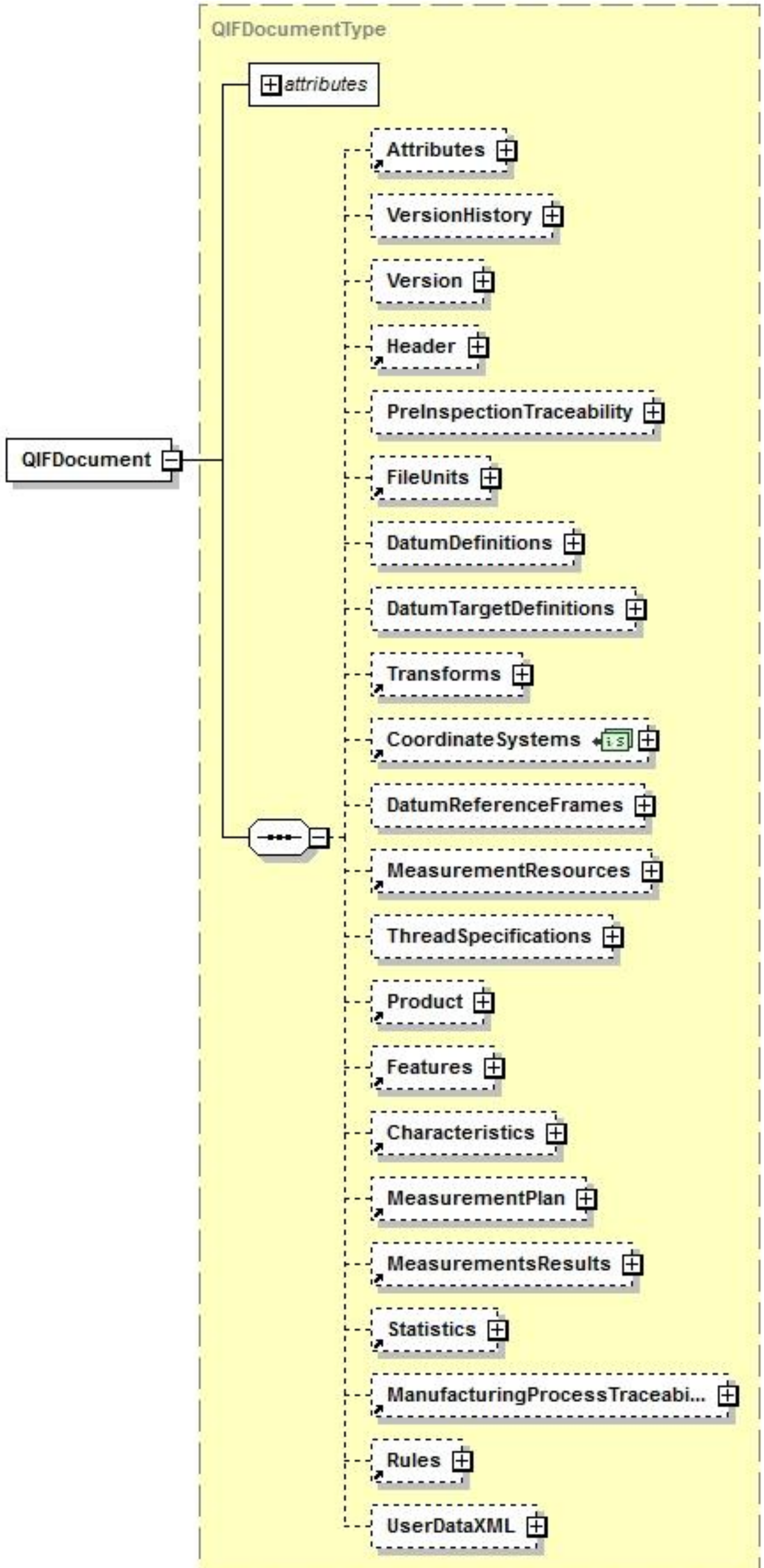


Figure 4 – Structure of the QIFDocument element

However a user has the option of putting related application information in a single instance file or keeping that information in separate instance files. A QIF Statistics instance file might contain only identifying information and a **Statistics** *element*. The product information referenced in the **Statistics** *element* might be in a second instance file that was generated when the product was designed, and the measurement results information referenced in the **Statistics** *element* might be in a third instance file that was generated when an instance of the product was measured. When related information from different applications is in a single file, it is connected using identifiers (**id**s) that are local to the file. When that information is in separate files, it is connected using a combination of local ids and QIF Persistent Identifiers (**QPId**s), which are universally unique.

A brief description of all sub-elements of QIF Document follows. All *element*s are optional, but a **QIFDocument** should contain at least a **Header** and one or more of **MeasurementResources**, **Product**, **MeasurementPlan**, **MeasurementsResults**, **Rules**, and **Statistics**. It is strongly suggested that every QIFDocument also have **Version** and **FileUnits**.

- **Attributes** – provides the user with opportunity to attach typed information that is not modeled in QIF. It is described further in Section 6.21. The ***AttributesType*** and **Attributes** global *element* are defined in Primitives.xsd.
- **VersionHistory** – version history information about the file. The ***VersionHistoryType*** is defined in Intermediates.xsd.
- **Version –** version information about the file, including **TimeCreated**, **Signoffs**, and **ThisInstanceQPId**, a universally unique identifier for the file. QPIds are discussed in Section 6.11.1. ***QPIdType*** is defined in Primitives.xsd.
- **Header** – identification of the source of the file and the scope of the file. The ***QIFDocumentHeaderType*** and **Header** global *element* are defined in QIFDocument.xsd.
- **PreInspectionTraceability** – information about the provenance of the file before inspection has occurred. The ***PreInspectionTraceabilityType*** is defined in Traceability.xsd. Preinspection traceability is discussed briefly in Section 6.13.2 of Part 2.
- **FileUnits –** information about the units used in the file. Units are discussed in Section 6.15 of this part and in more detail in Section 6.14 of Part 2. The ***FileUnitsType*** and **FileUnits** global *element* are defined in Units.xsd.
- **DatumDefinitions –** a list of datum definitions. Datum definitions are discussed in Section 6.14.6.1. ***DatumDefinitionsType*** is defined in IntermediatesPMI.xsd.
- **DatumTargetDefinitions –** a list of datum targets. Datum target types are listed in Section 6.7 of Part 2. The ***DatumTargetDefinitionsType*** and the various datum target types are defined in IntermediatesPMI.xsd.
- **Transforms –** a list of transforms. Transforms are discussed in Section 6.13.3. The ***TransformListType*** and **Transform** global *element* are defined in IntermediatesPMI.xsd.

- **CoordinateSystems** – a list of coordinate systems. Coordinate systems are discussed in Sections 6.13.4, 6.13.5, and 6.13.6. The ***CoordinateSystemType***, ***CoordinateSystemListType*** and **CoordinateSystems** global *element* are defined in IntermediatesPMI.xsd.
- **DatumReferenceFrames –** a list of datum reference frames. Datum reference frames are discussed in Section 6.14.6. The ***DatumReferenceFramesType*** and the ***DatumReferenceFrameType*** are defined in IntermediatesPMI.xsd.
- **MeasurementResources –** information about measurement resources. Measurement resources are discussed in Part 5. The ***MeasurementResourcesType***, **MeasurementResources** global *element*, and other measurement resource types are defined in QIFMeasurementResources.xsd.
- **ThreadSpecifications –** a list of thread specifications. Thread specifications are discussed in Section 6.18. Thread specification types are defined in IntermediatesPMI.xsd. Thread characteristics and features are defined in Characteristics.xsd and Features.xsd, respectively.
- **Product –** information about parts and assemblies, both designs and physical instances of the designs. These are discussed in Section 6.9 of this document and in more detail in Part 3 and sections 6.1, 6.6, 6.9, 6.12, and 6.15 of Part 2. The ***ProductType*** and **Product** global *element* are defined in QIFProduct.xsd.
- **Features –** information about features. Features are discussed in Section 6.7.2, 6.7.5, 6.16, 6.17, and 6.19 of this document and in Section 6.4 of Part 2. The ***FeaturesAspectsListsType*** and **Features** global *element* are defined in Features.xsd.
- **Characteristics –** information about characteristics. Characteristics are discussed in Sections 6.7.3, 6.7.4, 6.7.5, and 6.20 of this document and in Section 6.2 of Part 2. The ***CharacteristicAspectsListsType*** and **Characteristics** global *element* are defined in Characteristics.xsd.
- **MeasurementPlan –** information about a measurement plan. Measurement plans are discussed briefly in Section 6.2 and in more detail in Part 4. The ***MeasurementPlanType*** and **MeasurementPlan** global *element* are defined in QIFPlan.xsd.
- **MeasurementsResults –** information about one or more measurement results. Each measurement results includes all information associated with a single product measurement such as actual feature, characteristic, and coordinate system transform information. Measurement results are discussed briefly in Section 6.11.2 and in detail in Part 7. The ***MeasurementsResultsType*** and **MeasurementsResults** global *element* are defined in QIFResults.xsd.
- **Statistics –** information about quality statistics aimed primarily at process control; both plans and results are included. Statistics is discussed briefly in Section 6.2 of this document and Section 6.11 of Part 2. Statistics is discussed in detail in Part 8. The ***StatisticsType*** and **Statistics** global *element* are defined in QIFStatistics.xsd.
- **ManufacturingProcessTraceabilities** – information about manufacturing processes. Manufacturing process traceability is discussed briefly in Section 6.13.5 of Part 2. The ***ManufacturingProcessTraceabilitiesType*** and the **ManufacturingProcessTraceabilities** global *element* are defined in Traceability.xsd.

- **Rules** – information about rules to be used in inspection planning and/or programming. Rules are discussed briefly in Section 6.2 and in detail in Part 6. Rules types are defined in QIFRules.xsd.
- **UserDataXML –** information in XML format modeled in some non-QIF namespace. The ***UserDataXMLType*** and the **UserDataXML** global *elemen*t are defined in Primitives.xsd. This is intended to be used only for data that cannot be modeled in QIF.

6.7.2 Four aspects of features data

Feature information is defined in the QIF library using four aspects: definition, nominal, item, and actual. In a QIF instance file each feature data object has a unique identifier, and relationships between objects are expressed by references to the identifiers as shown in Figure 5. These four library data types were designed to express quality information beyond the scope of solely inspection results reporting. The item aspect, in particular, includes information related to part design as well as information generated by planning activities.

Figure 5 – Reference connections among feature data objects in a QIF XML instance file. Solid lines show required references, dashed lines show optional references.

The four aspects of feature data will be illustrated with the simple example of a plate with four holes as shown in Figure 6.

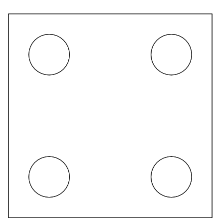


Figure 6 – A plate with four holes

This plate with holes can exist in several contexts: it may be a printed 2D drawing, it may be a 3D solid MBD, it may be a CMM inspection plan or program, it may be an actual physical part, it may be a CMM results report, or it may exist as all of the above. Regardless, in QIF the four holes would be considered as cylinder features. In the MBD or drawing contexts, nominal information for these cylinders will exist. In the physical part context actual information about the cylinders can exist if they are measured. So naturally, one would assume that QIF would only need to contain objects that define feature nominal and feature actual information. Such a simple approach can result in the redundant expression of data and may not mirror the design intent.

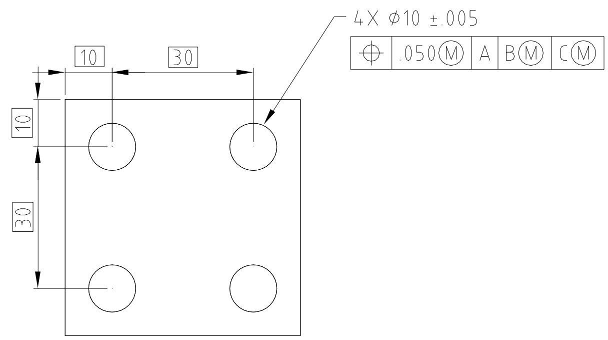


Figure 7 – A plate with four holes and GD&T

The design of the plate shown in Figure 7 illustrates that some nominal information about the holes is shared and some is not. Each cylinder has a unique nominal location defined by basic dimensions but all have a shared nominal diameter. QIF therefore splits the nominal information between a shareable feature definition and a non-shareable feature nominal.

A **feature definition** data object is intended to be reusable, in that it includes information (e.g., cylinder: diameter), that is independent of a specific instance of the feature (e.g., a specific hole in our example). A single feature definition can be referenced by many nominal feature objects. Only nominal feature objects may reference feature definition objects.

In the example, the cylinder definition with the shared diameter of the holes might look like this in a QIF XML instance file:

```
<CylinderFeatureDefinition id="22">
  <Diameter>10</Diameter>
</CylinderFeatureDefinition>
```

A **feature nominal** data object adds additional feature information to the feature definition by defining information unique to a particular instance of a feature. For example, an instance of the ***CylinderFeatureDefinitionType*** provides the diameter for a cylinder, while an instance of the ***CylinderFeatureNominalType*** references the ***CylinderFeatureDefinitionType***, and gives the location point and the axis vector and optional target points on a specific cylinder on a part to be measured.

Feature nominal data is unique to a particular feature in the context of a simple part. But in the context of multiple instances of a part in an assembly it can be shared. In this case the feature item's (see below) reference to the nominal will include its assembly path. The feature nominal has optional *elements* for **Name** and **Description** which may be used to identify and describe the feature in the component part context perhaps as it appears in a CAD system feature tree.

If a part is measured, then each cylinder will have unique location and size information. No feature actual data can be shared among feature instances.

The four holes in the example would have four feature nominals in a QIF XML file:

```
<CylinderFeatureNominal id="23">
  <FeatureDefinitionId>22</FeatureDefinitionId>
  <Axis>
    <AxisPoint>40 40 0</AxisPoint >
    <Direction>0 0 1</Direction>
  </Axis>
</CylinderFeatureNominal>
<CylinderFeatureNominal id="24">
  <FeatureDefinitionId>22</FeatureDefinitionId>
  <Axis>
    <AxisPoint>40 10 0</AxisPoint >
    <Direction>0 0 1</Direction>
  </Axis>
</CylinderFeatureNominal>
<CylinderFeatureNominal id="25">
  <FeatureDefinitionId>22</FeatureDefinitionId>
  <Axis>
    <AxisPoint>10 10 0</AxisPoint >
    <Direction>0 0 1</Direction>
  </Axis>
</CylinderFeatureNominal>
<CylinderFeatureNominal id="26">
  <FeatureDefinitionId>22</FeatureDefinitionId>
  <Axis>
    <AxisPoint>10 40 0</Z> </AxisPoint >
    <Direction>0 0 1</Direction>
  </Axis>
</CylinderFeatureNominal>
```

Each has a unique id and specifies a unique location but each references the same feature definition.

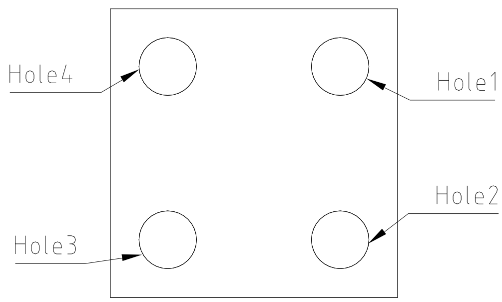


Figure 8 – A plate with four holes with names.

A **feature item** data object represents an instance of a feature at any stage of the metrology process - before or after a feature has been measured. The feature item data object provides: the name assigned to the feature (as in Figure 8), optional links to upstream CAD data, reference to a part definition object ID, and an optional reference to a nominal feature object.

If the same feature on the same physical part is measured several times, it is expected that a feature item data object will be defined for each measurement. Some other examples of workflow that cause instantiation of a feature item object include: any process at any time that requires a named feature, planning inspection of a part, free-form measurement of a part for reverse engineering purposes, bringing a legacy CMM report into QIF, or mining a legacy CMM program for nominal features and characteristics.

A QIF XML instance file might have feature items like:

```
<CylinderFeatureItem id="27">
  <FeatureNominalId>23</FeatureNominalId>
  <FeatureName>Hole_1</FeatureName>
  …
</CylinderFeatureItem>
<CylinderFeatureItem id="28">
  <FeatureNominalId>24</FeatureNominalId>
  <FeatureName>Hole_2</FeatureName>
  …
</CylinderFeatureItem>
<CylinderFeatureItem id="29">
  <FeatureNominalId>25</FeatureNominalId>
  <FeatureName>Hole_3</FeatureName>
  …
</CylinderFeatureItem>
```

```
<CylinderFeatureItem id="30">
  <FeatureNominalId>26</FeatureNominalId>
  <FeatureName>Hole_4</FeatureName>
  …
</CylinderFeatureItem>
```

Each item references a single nominal and assigns a feature name (the ellipses … indicate extra required *elements* outside the scope of this simple example).

Feature nominal objects can be defined at the whole product level in the global coordinate system or at the part or subassembly level in a local coordinate system whereas the feature item is always defined at the whole product level. The feature nominal can be referenced by several feature items, one for each part or subassembly instance in an assembly. When a feature item references such a feature nominal, the optional **asmPath** *attribute* on the **FeatureNominalId** *element* is used to unambiguously define the assembly path which takes the feature nominal from the part or subassembly context to the whole part context.

A QIF XML instance file of an assembly containing two instances of the 4-hole plate might have feature items like:

```
<CylinderFeatureItem id="27">
  <FeatureNominalId asmPath=”6”>23</FeatureNominalId>
  <FeatureName>Part1_Hole_1</FeatureName>
  …
</CylinderFeatureItem>
<CylinderFeatureItem id="28">
  <FeatureNominalId asmPath=”6”>24</FeatureNominalId>
  <FeatureName>Part1_Hole_2</FeatureName>
  …
</CylinderFeatureItem>
<CylinderFeatureItem id="29">
  <FeatureNominalId asmPath=”6”>25</FeatureNominalId>
  <FeatureName>Part1_Hole_3</FeatureName>
  …
</CylinderFeatureItem>
<CylinderFeatureItem id="30">
  <FeatureNominalId asmPath=”6”>26</FeatureNominalId>
  <FeatureName>Part1_Hole_4</FeatureName>
  …
</CylinderFeatureItem>
<CylinderFeatureItem id="31">
  <FeatureNominalId asmPath=”8”>23</FeatureNominalId>
  <FeatureName>Part2_Hole_1</FeatureName>
  …
</CylinderFeatureItem>
<CylinderFeatureItem id="32">
  <FeatureNominalId asmPath=”8”>24</FeatureNominalId>
  <FeatureName>Part2_Hole_2</FeatureName>
```

```
    …
  </CylinderFeatureItem>
  <CylinderFeatureItem id="33">
    <FeatureNominalId asmPath=”8”>25</FeatureNominalId>
    <FeatureName>Part2_Hole_3</FeatureName>
    …
  </CylinderFeatureItem>
  <CylinderFeatureItem id="34">
    <FeatureNominalId asmPath=”8”>26</FeatureNominalId>
    <FeatureName>Part2_Hole_4</FeatureName>
    …
  </CylinderFeatureItem>
```

A **feature actual** data object provides feature information that has been measured or constructed. For example, a ***CylinderFeatureActualType*** will contain the actual location, orientation, and size of a cylinder. The feature actual data object contains a reference to the associated feature item.

For an inspection that has been programmed from CAD data, the feature item may reference the related nominal feature object (which in turn, has a related feature definition object). For feature actual data generated during a reverse engineering process, a QIF XML instance file may not contain nominal feature data. In which case the feature actual would reference a feature item which does not reference a feature nominal.

The four holes when measured might look like this in a QIF XML instance file:

```
  <CylinderFeatureActual id="31">
    <FeatureItemId>27</FeatureItemId>
    <Axis>
      <AxisPoint>40.002 39.994 0</AxisPoint>
      <Direction>-0.001 0 1</Direction>
    </Axis>
    <Diameter>10.003</Diameter>
  </CylinderFeatureActual>
  <CylinderFeatureActual id="32">
    <FeatureItemId>28</FeatureItemId>
    <Axis>
      <AxisPoint>39.967 10.011 0</AxisPoint>
      <Direction>0.009 -0.009 0.999</Direction>
    </Axis>
    <Diameter>10.005</Diameter>
  </CylinderFeatureActual>
  <CylinderFeatureActual id="33">
    <FeatureItemId>29</FeatureItemId>
    <Axis>
      <AxisPoint>10.002 10.013 0</AxisPoint>
      <Direction>0.001 0 1</Direction>
    </Axis>
```

```
    <Diameter>9.996</Diameter>
  </CylinderFeatureActual>
  <CylinderFeatureActual id="34">
    <FeatureItemId>30</FeatureItemId>
    <Axis>
      <AxisPoint>9.987 40.013 0</AxisPoint>
      <Direction>0 -0.004 1</Direction>
    </Axis>
    <Diameter>10.007</Diameter>
  </CylinderFeatureActual>
```

Each references the corresponding cylinder nominal and each has a unique measurement result.

6.7.3 Four aspects of characteristics

As with features, characteristics have four aspects defined in the QIF library: definition, nominal, item, and actual. Data objects of each aspect type can be linked in a QIF XML instance file to express the semantics of GD&T and quality workflow using the scheme described in section 6.5. As with features, the aspects cover a quality workflow scope wider than solely results reporting.

The **characteristic definition** is the part of a characteristic that can be shared among different characteristics. An example would be a standard diameter tolerance; one manufacturer, for instance, has a standard diameter tolerance for sheet metal parts of (+.25/-.04) mm regardless of the diameter. As another example, one often sees tolerances specified for dimensions based on the number of decimal places: dimensions to one decimal place are ±.2 mm, those to two decimal places are ±.05 mm, etc.

In the example shown in Figure 7, a QIF XML instance file representation of the diameter tolerance as a characteristic definition might look like:

```
  <DiameterCharacteristicDefinition id="40">
    <Tolerance>
      <MaxValue>0.005</MaxValue>
      <MinValue>-0.005</MinValue>
      <DefinedAsLimit>false</DefinedAsLimit>
    </Tolerance>
  </DiameterCharacteristicDefinition>
```

The **characteristic nominal** is that part of a characteristic that is not shared among different characteristics, not to be confused with sharing a characteristic among several features. An example would be a diameter tolerance for a set of holes in a pattern all with the same diameter. That shared diameter becomes the target value in the nominal characteristic. Very often, each characteristic definition will only be referenced by a single characteristic nominal, the pair together representing one call-out on a print such as the diameter with tolerance in Figure 7.

In the example, an instance file containing the diameter characteristic nominal might look like:

```
<DiameterCharacteristicNominal id="41">
  <CharacteristicDefinitionId>40</CharacteristicDefinitionId>
  <TargetValue>10</TargetValue>
</DiameterCharacteristicNominal>
```

This single diameter characteristic nominal can be shared among the four holes.

The **characteristic item** is the mechanism used to apply a tolerance to an individual feature. Our plate with holes would have one characteristic item for each hole because each hole will have an individual tolerance condition.

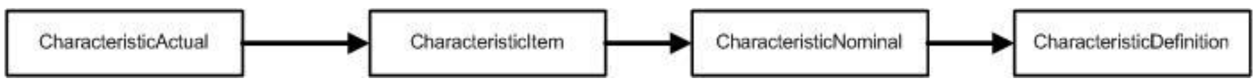


Figure 9 – References among characteristic data objects in a QIF XML instance file.

Each characteristic item references the single shared characteristic nominal, and in turn that characteristic nominal references a single characteristic definition which may or may not be referenced by other characteristic nominals. Unlike feature items which may optionally reference a feature nominal, characteristic items are required to reference a characteristic nominal.

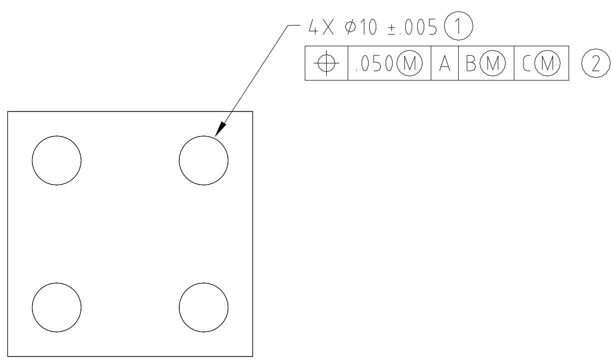


Figure 10 – A plate with ballooned tolerances.

The idea of a characteristic item may at first seem a bit redundant but it allows for a unique identifying name, or key characteristic identifier to be assigned on a per-feature basis. Our diameter tolerance might be “ballooned” on the print (as in Figure 10) as key characteristic number 1, then each characteristic item could be labeled with key characteristics of 1A, 1B, 1C or 1_1, 1_2, 1_3 depending on company standards.

From the example, here is what a QIF XML instance file with key characteristics might look like:

```
<DiameterCharacteristicNominal id="41">
  <CharacteristicDefinitionId>40</CharacteristicDefinitionId>
  <TargetValue>10</TargetValue>
  <KeyCharacteristic>
    <Designator>1</Designator>
  </KeyCharacteristic>
</DiameterCharacteristicNominal>
…
<DiameterCharacteristicItem id="42">
  <Name>Hole_1_diam</Name>
  <KeyCharacteristic>
    <Designator>1_1</Designator>
  </KeyCharacteristic>
  <CharacteristicNominalId>41</CharacteristicNominalId>
</DiameterCharacteristicItem>
<DiameterCharacteristicItem id="43">
  <Name>Hole_2_diam</Name>
  <KeyCharacteristic>
    <Designator>1_2</Designator>
  </KeyCharacteristic>
  <CharacteristicNominalId>41</CharacteristicNominalId>
</DiameterCharacteristicItem>
<DiameterCharacteristicItem id="44">
  <Name>Hole_3_diam</Name>
  <KeyCharacteristic>
    <Designator>1_3</Designator>
  </KeyCharacteristic>
  <CharacteristicNominalId>41</CharacteristicNominalId>
</DiameterCharacteristicItem>
<DiameterCharacteristicItem id="45">
  <Name>Hole_4_diam</Name>
  <KeyCharacteristic>
    <Designator>1_4</Designator>
  </KeyCharacteristic>
  <CharacteristicNominalId>41</CharacteristicNominalId>
</DiameterCharacteristicItem>
```

The optional **KeyCharacteristic** *element* on the characteristic nominal defines the shared key characteristic designator (balloon number) from the call-out and the **KeyCharacteristic** *elements* on each characteristic item define the individual key characteristic designators for the diameter of the four individual holes.

The **characteristic actual** is the evaluation of the characteristic based on direct measurement or from feature measurement data. There will be one characteristic actual for each measured characteristic. Just as with feature actual information, there is no shareable information among characteristic actuals.

For the example, an instance file with actuals might look like:

```
<DiameterCharacteristicActual id="46">
  <Status>
    <CharacteristicStatusEnum>PASS</CharacteristicStatusEnum>
  </Status>
  <CharacteristicItemId>42</CharacteristicItemId>
  <Value>10.003</Value>
</DiameterCharacteristicActual>
<DiameterCharacteristicActual id="47">
  <Status>
    <CharacteristicStatusEnum>PASS</CharacteristicStatusEnum>
  </Status>
  <CharacteristicItemId>43</CharacteristicItemId>
  <Value>10.005</Value>
</DiameterCharacteristicActual>
<DiameterCharacteristicActual id="48">
  <Status>
    <CharacteristicStatusEnum>PASS</CharacteristicStatusEnum>
  </Status>
  <CharacteristicItemId>44</CharacteristicItemId>
  <Value>9.996</Value>
</DiameterCharacteristicActual>
<DiameterCharacteristicActual id="49">
  <Status>
    <CharacteristicStatusEnum>FAIL</CharacteristicStatusEnum>
  </Status>
  <CharacteristicItemId>45</CharacteristicItemId>
  <Value>10.007</Value>
</DiameterCharacteristicActual>
```

6.7.4 Default tolerances and characteristics

There are four types of dimension: a basic dimension like $\boxed{40.0}$ (a dimension to define the nominal location of a feature to which a geometric tolerance is applied), a reference dimension like (40.0) (a dimension which can be calculated from other dimensions but which is provided for easy reference), a directly toleranced dimension like 40.0±0.1 (either as specification limits or as a bidirectional tolerance about a nominal value), and a dimension which is not directly toleranced like 40.0.

This latter dimension, although not explicitly toleranced, is often subject to an “unless otherwise specified” default tolerance. Such default tolerances are often called “box tolerances” because they can appear in a box at the corner of a drawing and often take the form:

Unless otherwise specified	
X.X	±.2 mm
X.XX	±.05 mm
X.XXX	±.005 mm
XX°	±1°

In QIF characteristics are explicitly typed; a length nominal cannot use a diameter definition to define its tolerance. Because these default tolerances cross type boundaries (the same tolerance may apply to a length, diameter, distance, coordinate etc.) QIF has a mechanism for storing and referencing un-typed default tolerances. Instead of defining a tolerance directly in a characteristic definition such as:

```
<DiameterCharacteristicDefinition id="40">
  <Tolerance>
    <MaxValue>0.005</MaxValue>
    <MinValue>-0.005</MinValue>
    <DefinedAsLimit>false</DefinedAsLimit>
  </Tolerance>
</DiameterCharacteristicDefinition>
```

The tolerance values can be defined by reference to a default tolerance:

```
<DiameterCharacteristicDefinition id="40">
  <Tolerance>
    <DefinitionId>20</DefinitionId>
    <DefinedAsLimit>false</DefinedAsLimit>
  </Tolerance>
</DiameterCharacteristicDefinition>
```

where the **DefinitionId** *element* references a default tolerance found in the **DefaultToleranceDefinitions** *element*. Such a default tolerance has the form:

```
<LinearToleranceDefinition id="20">
  <Tolerance>
    <MaxValue>0.005</MaxValue>
    <MinValue>-0.005</MinValue>
  </Tolerance>
</LinearToleranceDefinition>
```

This tolerance can be referenced by any linear value characteristic. The default tolerance for angular value characteristics is defined in a separate *element*:

```
<AngularToleranceDefinition id="21">
  <Tolerance>
    <MaxValue>1</MaxValue>
    <MinValue>-1</MinValue>
  </Tolerance>
</AngularToleranceDefinition>
```

The relationship between a default tolerance definition and the box tolerance annotation can be made more explicit with a note contained in the optional **Attributes** *element*:

```
<LinearToleranceDefinition id="20">
  <Attributes>
    <AttributeStr name=”Applicability”, value=”X.XXX”\>
  <\Attributes>
  <Tolerance>
    <MaxValue>0.005</MaxValue>
    <MinValue>-0.005</MinValue>
  </Tolerance>
</LinearToleranceDefinition>
```

A similar concept is that of a default characteristic applied to all non-directly toleranced features of a part, for example, “Unless otherwise specified all surfaces | ⌓ | 0,5 | A | B | C |”. Such contextual tolerances are stored in a special **DefaultCharacteristicDefinitions** *element* which is a list of characteristic definitions.

At the PMI stage of design, default tolerances and characteristics can be indicated simply by their presence in the **DefaultToleranceDefinitions** *element* and the **DefaultCharacteristicDefinitions** *element*. The connections between these tolerances and the features to which they apply can be made later at the metrology planning stage.

6.7.5 Relationships between the aspects

Three of the four aspects of characteristics are instantiated in the global **Characteristics** *element* of type ***CharacteristicApectsListsType***. This type has an *element* for listing instances of each of the following aspects: **CharacteristicDefinitions**, **CharacteristicNominals**, and **CharacteristicItems**. The fourth, **CharacteristicActuals** is found on a per-measured-part basis in the **MeasurementResults** *element* of **MeasurementsResults**. This allows for multiple part measurements to be reported in a single QIF instance file. If more than one part has been measured then there will be more than one characteristic actual referencing the same characteristic item.

To traverse all characteristics in a QIF XML instance file, the starting point would be the **CharacteristicItems** sub-*element* of the **Characteristics** *element*.

The four aspects of features are similarly instantiated in the global **Features** *element* of type ***FeatureAspectsListsType*** and in the **FeatureActuals** *element* on a per-measured-part basis in the **MeasurementResults** *element* of **MeasurementsResults**. ***FeatureAspectsListsType*** has an *element* for listing instances of **FeatureDefinitions**, **FeatureNominals**, and **FeatureItems**. Of the three, only **FeatureItems** *element* is not optional. The feature item will come into existence either at the planning stage with reference to nominal (and definition) data or come into existence as actuals are measured during a reverse-engineering use case.

To traverse all features in a QIF XML instance file, the starting point would be the **FeatureItems** sub-*element* of the **Features** *element*.

6.7.5.1 Connecting characteristics and features

The connections among the four aspects for both characteristic aspects as a group and feature aspects as a group have been discussed earlier. In general, the actual references an item, the item references a nominal, and the nominal references the definition. The connection between characteristics and features is accomplished with the **FeatureItemIds** *element* on the ***CharacteristicItemBaseType*** from which all characteristic item types are derived.

The **FeatureItemIds** *element* allows a characteristic item to reference the feature item or feature items of the feature or features to which it applies. When more than one feature is referenced by the **FeatureItemIds** *element* it is because more than one feature is required to evaluate the tolerance, such as a distance or angle between two features. To determine the characteristic-feature relationships in a QIF instance file, the starting point would be the **CharacteristicItems** sub-*element* of the **Characteristics** *element*.

There is a secondary method of connecting characteristics and features for use at the PMI design stage when neither feature items nor characteristic items yet exist. Product manufacturing information (characteristics) are attached to the model based definition (features) by the **FeatureNominalIds** *element* on the ***CharacteristicNominalBaseType***.

Unlike the **FeatureItemsIds** *element* which references the minimum number of features necessary for the measurement of a single characteristic item, the **FeatureNominalIds** *element* can reference the many feature nominals to which the characteristic nominal applies. This one-to-many reference for a single characteristic nominal will give rise to as many characteristic items as there are features (or feature pairs, etc.).

From the example, in a QIF XML instance file containing both characteristics and features, the connections between characteristic items and feature items and characteristic nominals and feature nominals might look like this:

```
<DiameterCharacteristicNominal id="41">
  <CharacteristicDefinitionId>40</CharacteristicDefinitionId>
  <FeatureNominalIds N="4">
    <Id>23</Id>
    <Id>24</Id>
    <Id>25</Id>
    <Id>26</Id>
  </FeatureNominalIds>
  <TargetValue>10</TargetValue>
  <KeyCharacteristic>
    <Designator>1</Designator>
```

```
    </KeyCharacteristic>
  </DiameterCharacteristicNominal>
  …
  <DiameterCharacteristicItem id="42">
    <Name>Hole_1_diam</Name>
    <FeatureItemIds N=”1”>
      <Id>27</Id>
    </FeatureItemIds>
    <CharacteristicNominalId>41</CharacteristicNominalId>
    </DiameterCharacteristicItem>
  <DiameterCharacteristicItem id="43">
    <Name>Hole_2_diam</Name>
   <FeatureItemIds N=”1”>
     <Id>28</Id>
    </FeatureItemIds>
    <CharacteristicNominalId>41</CharacteristicNominalId>
  </DiameterCharacteristicItem>
  <DiameterCharacteristicItem id="44">
    <Name>Hole_3_diam</Name>
    <FeatureItemIds N=”1”>
      <Id>29</Id>
    </FeatureItemIds>
    <CharacteristicNominalId>41</CharacteristicNominalId>
  </DiameterCharacteristicItem>
  <DiameterCharacteristicItem id="45">
    <Name>Hole_4_diam</Name>
    <FeatureItemIds N=”1”>
      <Id>30</Id>
    </FeatureItemIds>
    <CharacteristicNominalId>41</CharacteristicNominalId>
  </DiameterCharacteristicItem>
```

(Key characteristic designators have been removed in the above example.)

In the context of a single part measurement this connection between the characteristic item and feature item would be sufficient to imply the connection between the single characteristic actual and single feature actual. But, if the results of more than one part measurement are included in a single QIF document this connection is no longer sufficient. Therefore, to explicitly make the connection between a characteristic actual and the actual feature or features to which it applies the **FeatureActualIds** *element* is used.

From the example, in a QIF XML instance file containing both characteristics and features, the connections between characteristic actuals and feature actuals might look like this:

```
  <DiameterCharacteristicActual id="46">
    <Status>
      <CharacteristicStatusEnum>PASS</CharacteristicStatusEnum>
    </Status>
    <CharacteristicItemId>42</CharacteristicItemId>
```

```
      <FeatureActualIds N="1">
        <Id>31</Id>
      </FeatureActualIds>
      <Value>10.003</Value>
    </DiameterCharacteristicActual>
    <DiameterCharacteristicActual id="47">
      <Status>
        <CharacteristicStatusEnum>PASS</CharacteristicStatusEnum>
      </Status>
      <CharacteristicItemId>43</CharacteristicItemId>
      <FeatureActualIds N="1">
        <Id>32</Id>
      </FeatureActualIds>
      <Value>10.005</Value>
    </DiameterCharacteristicActual>
    <DiameterCharacteristicActual id="48">
      <Status>
        <CharacteristicStatusEnum>PASS</CharacteristicStatusEnum>
      </Status>
      <CharacteristicItemId>44</CharacteristicItemId>
      <FeatureActualIds N="1">
        <Id>33</Id>
      </FeatureActualIds>
      <Value>9.996</Value>
    </DiameterCharacteristicActual>
    <DiameterCharacteristicActual id="49">
      <Status>
        <CharacteristicStatusEnum>FAIL</CharacteristicStatusEnum>
      </Status>
      <CharacteristicItemId>45</CharacteristicItemId>
      <FeatureActualIds N="1">
        <Id>34</Id>
      </FeatureActualIds>
      <Value>10.007</Value>
    </DiameterCharacteristicActual>
```

6.7.5.2 Connections at various stages of the metrology process

The diagrams in this section illustrate the various possible connections among feature aspects, among characteristic aspects, and between characteristic aspects and feature aspects at various stages of the metrology process when feature data is present.

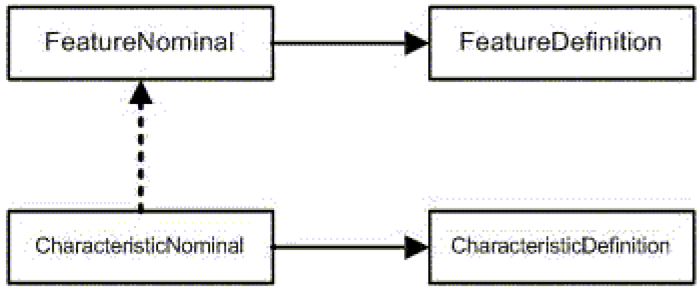


Figure 11 – Connections at the PMI Stage

Figure 11 shows the possible connections at the product manufacturing information stage of the metrology process. In this stage all information necessary to associate PMI (dimensions and tolerances, etc.) with features on a part is known. The dotted arrow between CharacteristicNominal and FeatureNominal indicates that the associated QIF *element* (**FeatureNominalIds**) is an optional *element*. It is nevertheless required for this use case in order to associate a characteristic with the feature or features to which it applies.

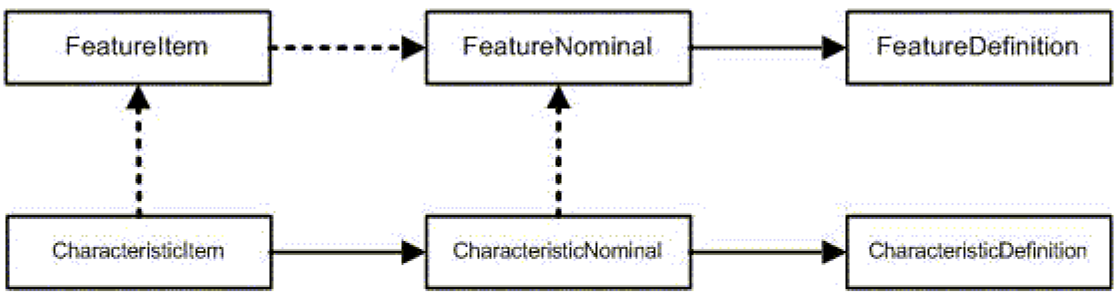


Figure 12 – Connections at the Planning Stage

Figure 12 shows the additional connections that come into being at the planning stage of metrology. Existing connections from the PMI stage persist and remain unchanged. Feature items and characteristic items which represent individual feature measurements and characteristic evaluations reference the data from the PMI stage. Again dotted arrows represent optional elements which are nonetheless required at the planning stage.

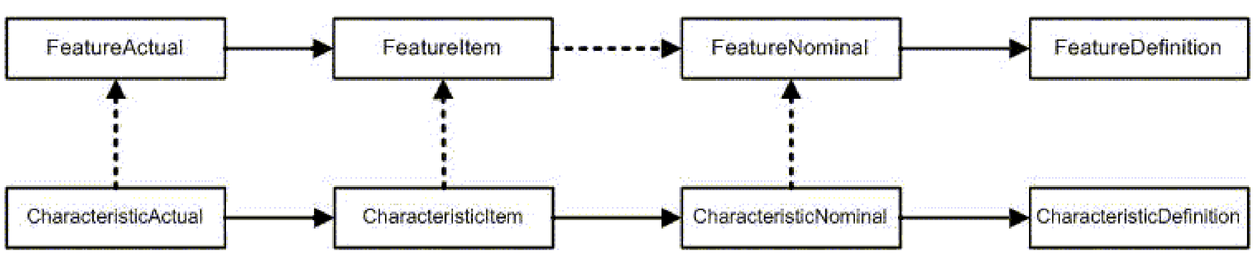


Figure 13 – Connections at the post-measurement stage

Figure 13 shows the new connections that arise at the post measurement stage. The feature and characteristic information and connections from the planning and PMI stages persist and remain unchanged and additional connections between actual characteristics evaluations and the measured features to which they apply and between actuals and their associated items are made. Again dotted arrows represent optional elements which are nonetheless required at the post-measurement stage.

Figure 13 shows the post-measurement connections when both feature and characteristic data are present. If only characteristic information is present then Figure 9 shows the connections between the four characteristic aspects at the post-measurement stage. Similarly, if only feature information, both nominal and actual, is present then Figure 5 shows the connections between the four feature aspects at the post-measurement stage.

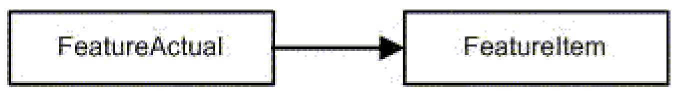


Figure 14 – Connections for reverse engineering

From the solid arrows in Figure 9 we see that characteristic nominal information is necessary when characteristic actual information is present. The dotted arrow in Figure 5 shows that this is not the case with feature information. Figure 14 shows the single connection between feature actual and feature item information in the reverse engineering use case. In this scenario features are measured without nominal information often with the goal of creating a drawing or electronic model for a part where one does not exist.

6.7.5.3 Parent Features

The **FeatureItem** *element* (defined in the ***FeatureItemBaseType***) contains an optional sub-*element* called **ParentFeatureItemId**. The purpose of this *element* is to connect a feature item created after the initial planning stage with an existing feature in a measurement plan.

Occasionally, a feature item in a measurement plan cannot be measured when the plan is to be executed and a new derivative feature is measured in its place. Some examples include:

- A cylinder in a thin walled material, perhaps with break-away from a punching operation, is measured as a circle.
- A part rests on a datum feature making it inaccessible and the surface plate is measured to simulate the datum surface.
- The edge of a sheet-metal part is inaccessible with available probe configurations so a gage block is held against the edge and measured.
- A countersink is not measured directly but instead a spherical tooling ball is placed in the hole and measured.

6.8 Hierarchy of required information

6.8.1 QIF use of optional elements

The QIF information model contains many optional data elements. Sometimes these elements represent data for special cases which do not arise in most applications. But mostly, data elements have been made optional to allow for a variety of use-cases while avoiding the possibility of the necessity to create fake or dummy data for a required data element that is outside the scope of a particular use case. For example, a caliper can only measure the size of an actual cylinder feature so required XYZ center location and IJK vector orientation elements for the actual cylinder would be outside the scope of the caliper-based use case.

There are situations where different use cases require that all data elements of a QIF type be optional. A caliper can measure the size of a hole but not the location or orientation, and a manual CMM with a conical hard probe can measure the location and orientation but not the size of a hole. Therefore, the location, orientation and size are all optional elements of the actual cylinder in QIF. The presence of such optional data in the QIF information model in no way indicates that the optional data is unimportant.

Wherever possible, data elements in QIF that are co-requisite are placed together in a data type so that all the data that is required is mandatory. An *element* of that data type may be optional. The result is that either all or none of the data is present. The QIF feature and characteristic aspects which isolate sharable information like size from non-sharable information like location sometimes result in co-requisite information appearing in two different data types. In such cases the annotation describing the data type will make reference to any co-requisite data elements.

The hierarchies of required information for various aspects of the QIF information model are discussed in the individual sections pertaining to those aspects.

6.8.2 Example: diameter characteristic

Consider the measurement of the diameter of a hole. The diameter might have a nominal representation and an associated tolerance which needs to be evaluated: or the diameter might be measured for the purposes of reverse engineering. Several examples are provided to illustrate how various levels of detail can be accommodated in QIF.

Common to the examples where a diameter is specified with a tolerance are the characteristic nominal and definition. Consider the diameter specification from Figure 10: ⌀10±.005. The tolerance is contained in the characteristic definition:

```
<DiameterCharacteristicDefinition id="40">
  <Tolerance>
    <MaxValue>0.005</MaxValue>
    <MinValue>-0.005</MinValue>
    <DefinedAsLimit>false</DefinedAsLimit>
  </Tolerance>
</DiameterCharacteristicDefinition>
```

and the nominal is contained in the characteristic nominal


```
<DiameterCharacteristicNominal id="41">
  <CharacteristicDefinitionId>40</CharacteristicDefinitionId>
  <TargetValue>10</TargetValue>
</DiameterCharacteristicNominal>
```

as described in section 6.7.2.

6.8.2.1 Measurement with a hard gage

Suppose that a hard gage is used to determine the tolerance condition. The gage consists of two pins: a smaller pin which must fit in the hole and a larger pin that must not. The result of the inspection is a simple pass/fail evaluation.

The actual results of the measurement might appear as follows in a QIF XML instance file:

```
<DiameterCharacteristicActual id="46">
  <Status>
    <CharacteristicStatusEnum>PASS</CharacteristicStatusEnum>
  </Status>
  <CharacteristicItemId>42</CharacteristicItemId>
</DiameterCharacteristicActual>
```

The actual diameter is not known and the optional **Value** *element* does not appear in the instance file.

6.8.2.2 Measurement with a caliper

If the same hole is measured with a caliper not only can the tolerance condition be determined but the actual diameter is available.

The actual results of the measurement might appear as follows in a QIF XML instance file:

```
<DiameterCharacteristicActual id="46">
  <Status>
    <CharacteristicStatusEnum>PASS</CharacteristicStatusEnum>
  </Status>
  <CharacteristicItemId>42</CharacteristicItemId>
  <Value>10.003</Value>
</DiameterCharacteristicActual>
```

6.8.2.3 Measurement with a coordinate measuring machine

If the same hole is measured with a coordinate measuring machine then even more data about the size can become available. In addition to the actual best-fit diameter, the minimum and maximum diameters might be reported.

The actual results of the measurement might appear as follows in a QIF XML instance file:

```
<DiameterCharacteristicActual id="46">
  <Status>
    <CharacteristicStatusEnum>PASS</CharacteristicStatusEnum>
```

```
    </Status>
    <CharacteristicItemId>42</CharacteristicItemId>
    <Value>10.003</Value>
    <MaxValue>10.004</MaxValue>
    <MinValue>10.001</MinValue>
  </DiameterCharacteristicActual>
```

6.8.2.4 Reverse engineering

If the same hole is measured with a caliper but no nominal information or tolerance information is available then the measurement results must be captured as a feature rather than as a characteristic.

The actual results of the measurement might appear as follows in a QIF XML instance file:

```
<CylinderFeatureActual id="31">
  <FeatureItemId>27</FeatureItemId>
  <Diameter>10.003</Diameter>
</CylinderFeatureActual>
```

Note that the **FeatureItemId** *element* is still present. The feature item will contain the feature name and material disposition (inner or outer) but will not reference a feature nominal.

6.8.2.5 Limit dimensions

A characteristic may be described by specification limits like `10.005/9.995` in which case the example diameter characteristic definition might appear as follows in a QIF XML instance file:

```
<DiameterCharacteristicDefinition id="40">
  <Tolerance>
    <MaxValue>10.005</MaxValue>
    <MinValue>9.995</MinValue>
    <DefinedAsLimit>true</DefinedAsLimit>
  </Tolerance>
</DiameterCharacteristicDefinition>
```

Because the size is defined by the limits, the target nominal diameter need not be specified, in which case the example diameter characteristic nominal might appear as follows in a QIF XML instance file:

```
<DiameterCharacteristicNominal id="41">
  <CharacteristicDefinitionId>40</CharacteristicDefinitionId>
</DiameterCharacteristicNominal>
```

Note that the optional **TargetValue** element is not present.

6.8.2.6 Basic dimensions

A characteristic may be described with a basic dimension like $\boxed{10}$, a nominal without tolerance limits. If a basic diameter is to be measured then the diameter characteristic definition might appear as follows in a QIF XML instance file:

```
<DiameterCharacteristicDefinition id="40">
  <NonTolerance>MEASURED</NonTolerance>
</DiameterCharacteristicDefinition>
```

The diameter characteristic nominal target value would specify the basic dimension:

```
<DiameterCharacteristicNominal id="41">
  <CharacteristicDefinitionId>40</CharacteristicDefinitionId>
  <TargetValue>10</TargetValue>
</DiameterCharacteristicNominal>
```

The diameter might be measured with a caliper but with no specification limits, no tolerance evaluation can be performed. The results of a measurement of a basic diameter might appear as follows in a QIF XML instance file:

```
<DiameterCharacteristicActual id="42">
  <Status>
    <CharacteristicStatusEnum>BASIC</CharacteristicStatusEnum>
  </Status>
  <CharacteristicNominalId>40</CharacteristicNominalId>
  <Value>10.003</Value>
</DiameterCharacteristicActual>
```

6.9 Actual parts and assemblies

A detailed description of the QIF data model for product definition including parts and assemblies can be found in QIF Part 3: Model Based Definition (MBD) Information Model and XML Schema File Section 7.4 Product Structure. What follows is an overview of that structure for the purpose of connecting actual data to nominal product definitions.

The QIF data model represents a product as one or more components. A component is an instance of a part or assembly placed into the context of the final product or a higher order assembly. The root component represents the whole, final product and is the root of a tree of components representing the assembly structure. An assembly path defines the path from the root component to an individual component in a product design. This tree structure allows for each part or sub-assembly to be defined once and then be placed into the final product or sub–assembly with a transformation matrix. A product consisting of a single part will have one component.

Each assembly path has a real world counterpart in the actual product. For example the concept of a “right rear wheel” exists both in an automobile design and on an actual automobile.

Therefore when actual data is associated with the product definition it is by reference to this assembly path. An assembly path may be shared by many objects in a QIF document therefore they are stored in one place: the **AsmPaths** *element*. Each assembly path has a QIF id by which it is referenced.

The simplest example showing the connection between an actual product and the product design is that of a single part. The product definition might be:

```
  <Product>
    <PartSet N="1">
      <Part id="1">
        <Name>WING_MIR_REENF</Name>
…
      </Part>
    </PartSet>
    <ComponentSet N="1">
      <Component id="2">
        <Part>
          <Id>1</Id>
        </Part>
      </Component>
    </ComponentSet>
    <AsmPaths N="1">
      <AsmPath id="3">
        <ComponentIds N="1">
          <Id>2</Id>
        </ComponentIds>
      </AsmPath>
    </AsmPaths>
  </Product>
```

A single part is defined which is used in a single component. The assembly path to that component contains one id.

It is the assembly path of the product design component that is associated with the actual component. Actual components are located in the **ActualComponentSet** *element* of the **MeasurementsResults** *element*:

```
 <MeasurementsResults>
 …
     <ActualComponentSet N="1">
       <ActualComponent id="4">
         <SerialNumber>X124578-17</SerialNumber>
         <Status>
           <InspectionStatusEnum>PASS</InspectionStatusEnum>
         </Status>
         <AsmPathId>3</AsmPathId>
       </ActualComponent>
     </ActualComponentSet>
 …
  </MeasurementsResults>
```

Each actual component has a QIF id which allows it to be referenced by one or more measurement results (the same product sample may be measured more than once) and/or by individual feature actuals:

```
 <MeasurementsResults>
 …
    <MeasurementResults id="86">
…
       <MeasuredFeatures>
         <FeatureActuals>
           <PointFeatureActual id="20">
             <FeatureItemId>19</FeatureItemId>
             <Location>2466.9 774.31 944.84</Location>
             <ActualComponentId>4</ActualComponentId>
           </PointFeatureActual>
 …
       </MeasuredFeatures>
       <ActualComponentIds N="1">
         <Id>4</Id>
       </ActualComponentIds>
    </MeasurementResults>
…
  </MeasurementsResults>
```

6.10 Checking connections between data objects

In QIF instance files, many connections between objects are made using an identifier and a reference to the identifier. For example the connection from a feature actual to a feature item is made by putting an identifier in the feature item and a reference to the identifier in the feature actual. All local (i.e. not significant outside the instance file) identifiers in the QIF schemas are of type ***QIFIdType***. All references to local identifiers are of type ***QIFReferenceType***. Both ***QIFIdType*** and ***QIFReferenceType*** are unsigned positive integers without leading zeros. For example, 1 and 3079 are valid, but 001 and +3079 are not. References that are significant between instance files are discussed in the next section.

In a QIF XML instance file, the reference contains the same symbol as the identifier of the object being referenced. For example, if the value of the **id** attribute of an instance of ***CylinderFeatureItemType*** is 26 then the value of the **FeatureItemId** element of an instance of ***CylinderFeatureActualType*** that uses that feature item is also 26.

QIF seeks to ensure that connections made using identifiers and references join the correct types of objects. For example, a reference from a cylinder feature actual to its item must identify a cylinder feature item, not any other type of object such as a transformation or a cone feature definition.

To ensure that identifier/reference pairs make matches between objects of the correct types, the QIF schemas contain several hundred *key*/*keyref* pairs, one for each variety of identifier/reference pair. *Key* and *keyref* are standard parts of the XML schema definition language (XSDL). Readily available XML instance file checkers will check whether or not *key*/*keyref* constraints are satisfied in an instance file governed by an XML schema. A detailed description of how *key* and *keyref* work may be found in books about the XML schema language such as [2]. In simplistic terms, a *key*/*keyref* pair locates two places in an (upside down) tree of objects that must contain identical information items. The two places are identified by describing the two paths that go downward to them from a common starting point. The *key*/*keyref* pair is located in a schema file at the common starting point.

For example, the CylinderFeatureActualToItemKeyref:

- is located in the **QIFDocument** *element* (the common starting point)
- references the CylinderFeatureItemKey
- has the xpath "t:**MeasurementsResults**/t:**MeasurementResults**/t:**MeasuredFeatures**/ t:**FeatureActuals**/t:**CylinderFeatureActual**", and
- has the field **FeatureItemId**.

The "t:" used with each *element* name in the xpath is a prefix standing for the QIF namespace and is required by the rules of XSDL in xpaths in any schema that uses a namespace. The header of each QIF schema file declares that the "t:" prefix will be used for the http://qifstandards.org/xsd/qif2 namespace. Almost any other prefix would have worked as well; "t:" is nice and short.

The CylinderFeatureItemKey:

- is also located in the **QIFDocument** *element* (the common starting point)
- has the xpath "t:**Features**/t:**FeatureItems**/t:**CylinderFeatureItem**", and
- has the field **@id**

The @ sign means the **id** field is an *attribute* rather than an *element*.

This sounds complex, but if we look at an instance file, it's fairly straightforward. Here is a snippet from an abbreviated QIFDocument instance file.

```
<QIFDocument>
  ...
  <MeasurementsResults ...>
    <MeasurementResults ...>
      ...
      <MeasuredFeatures>
        <FeatureActuals>
          <CylinderFeatureActual id="53">
            <FeatureItemId>26</FeatureItemId>
            <Axis> ... </Axis>
            <Diameter>2.5</Diameter>
          </CylinderFeatureActual>
          <CylinderFeatureActual id="54">
            <FeatureItemId>27</FeatureItemId>
            <Axis> ... </Axis>
            <Diameter>2.0</Diameter>
          </CylinderFeatureActual>
```

```
          ...
        </FeatureActuals>
      </MeasuredFeatures>
    </MeasurementResults>
  </MeasurementsResults>
  ...
  <Features>
    <FeatureItems>
      <CylinderFeatureItem id="26">
        <FeatureNominalId>33</FeatureNominalId>
        <FeatureName>top inner cylinder</FeatureName>
        ...
      </CylinderFeatureItem>
      <CylinderFeatureItem id="27">
        <FeatureNominalId>34</FeatureNominalId>
        <FeatureName>'dial' outer cylinder</FeatureName>
        ...
      </CylinderFeatureItem>
    </FeatureItems>
  </Features>
  ...
<QIFDocument>
```

When a *key*/*keyref* checking system reads this instance file and comes to the <QIFDocument> part, it knows it needs to check the **CylinderFeatureItemKey** and the **CylinderFeatureActualToItemKeyref** (since they are located in the **QIFDocument** *element*). To check the *key*, the checker follows every **Features/FeatureItems/CylinderFeatureItem** path down to the **id** *attribute*. In the snippet above, there are two such paths, one ending in "26", the other ending in "27". The checker puts the two **id**s into a collection of **id**s of **CylinderFeatureItem**s, checking to make sure that each of these is different from those already in the collection. To check the *keyref*, the checker follows every **MeasurementsResults**/**MeasurementResults**/**MeasuredFeatures**/ **FeatureActuals**/**CylinderFeatureActual** path down to the **FeatureItemId** *element*. In the snippet above, there are two such paths with *elements*, one with the value 26, the other with value 27. The checker checks that each of the values is in its collection of **id**s of **CylinderFeatureItem**s. The snippet above passes these checks.

The *key*/*keyref* pair mechanism is very good at catching errors in matching identifiers and references, but it is not foolproof. In all cases, when there are many objects of the same type, no automated check (such as *key*/*keyref*) will know which one is intended. For example, if an instance file has several instances of ***CylinderFeatureItemType*** and several instances of ***CylinderFeatureActualType*** (as in the snippet above), only the builder of the instance file will know which item (26 or 27 in the snippet) is supposed to go with which actual. In other cases, there is no way to define a *key* that discriminates between similar types of object because they are mixed together at the same location (in a set of transformations containing both actual transformations and nominal transformations, for example).

6.11 Tracking information through the product lifecycle

QIF is constructed to enable a seamless flow of information from upstream applications to downstream applications and to enable tracking information through a product's lifecycle.

6.11.1 UUIDs and QPIds

The primary mechanism used by QIF for identifying the same information in different places is the use of a persistent universally unique identifier (UUID), as standardized in ISO/IEC 9834-8. UUIDs have that name because the chances of generating two that are the same anywhere in our part of the universe are vanishingly small. Computer libraries for generating UUIDs conforming to the standard are widely available in many computer languages.

Using UUIDs, non-communicating systems can identify information uniquely. That information can be combined later into a single application or database without needing to resolve identifier conflicts.

As a number, a UUID is a 128 bit unsigned integer. As a text string in an instance file, a UUID is represented by 32 hexadecimal digits displayed in five groups separated by hyphens in the form 8-4-4-4-12 for a total of 36 characters (32 alphanumeric characters and four hyphens). An example of a UUID string is 550e8400-e29b-0518-a716-445664449c0b. The letters a through e are hexadecimal digits representing the numbers 10 through 15. Either lower case letters or upper case letters may be used in QIF for those digits.

The null UUID, which is equivalent in practice to no UUID since it is not unique, is one that has all 128 bits set to zero. In text form, that is written 00000000-0000-0000-0000-000000000000.

UUIDS used in QIF are called QPIds (pronounced "cupids"), a short form of QIF Persistent Identifier. Where a QPId is used to identify a file or *element*, its data type is ***QPIdType***, and it must be created by a well known UUID generator. Where a QPId is used to reference a file or *element*, its data type is ***QPIdReferenceType***, and it must match an existing QPId.

6.11.2 QIF data flow

Version 2.0 of QIF has six application areas. A **QIFDocument** instance file can contain any one of the applications or any combination of them. The most natural combinations in a single instance file are those that contain a sequence along the workflow shown in Figure 2, for example a **Product** *element* from the Define Product activity, a **QIFPlan** *element* from the Define Measurement Process activity, and a **MeasurementsResults** *element* from the Execute Measurement Process activity. Those *element*s would reference information common to all three of them contained elsewhere in the **QIFDocument** such as:

- file units
- datum definitions and datum reference frames
- measurement resources
- feature definitions, nominals, and items
- characteristic definitions, nominals, and items.

At the time a downstream process is started (measurements are starting to be taken, for example), a **QIFDocument** instance file containing the output of the upstream process is very likely to exist (a **QIFDocument** with **Product** and **MeasurementPlan** *element*s, for example); call it the QIFPlan file. When measurements are taken and it is desired to put the data in a **QIFDocument** instance file, the data must not simply be added to the QIFPlan file. A new **QIFDocument** instance file (call it the QIFResults file) should be created containing a **MeasurementsResults** *element*. Most *element*s of the **QIFDocument** in the QIFPlan file may be copied without change into the **QIFDocument** in the QIFResults file, but any **Version**, **VersionHistory**, or **Header** *element* must be new.

The way in which local identifiers (ids) are handled when a new downstream file is created from an old upstream file is up to the application building the new file. The local identifiers from the upstream process might be preserved or they might be changed. In any event, no system processing QIF instance files should rely on local identifiers remaining the same between files.

6.11.3 Using QPIds in QIF

To provide a method of uniquely identifying QIFDocument instance files and objects in them, QPIds may optionally be assigned as shown in Figure 15 – QPIdType Elements. To avoid confusing QPIds with ids, QPIds are always given as *element*s, and ids are always given as *attribute*s. The figure shows the *element* name. The value of each of the *element*s listed is of ***QPIdType***. Where the *element* is in a base type, all of the derived types (which is very many for features and characteristics) will also have the *element*.

QIFDocument.xsd
ThisInstanceQPId in the **Version** *element* of the ***QIFDocumentType***

QIFMeasurementResources.xsd
ThisInstanceQPId in the **Version** *element* of the ***MeasurementResourcesType***

QIFPlan.xsd
ThisInstanceQPId in the **Version** *element* of the ***MeasurementPlanType***

QIFProduct.xsd
QPId in ***ProductDefinitionBaseType***
QPId in ***ComponentType***
ThisInstanceQPId in the **Version** *element* of the **File** *element* of the **Header** *element* of the ***ProductType***

QIFResults.xsd
ThisResultsInstanceQPId in ***MeasurementResultsType***
ThisInstanceQPId in the **Version** *element* of the ***MeasurementsResultsType***

QIFRules.xsd
ThisInstanceQPId in the **Version** *element* of the ***QIFRulesType***

QIFStatistics.xsd
ThisStatisticalStudyPlanInstanceQPId in ***StatisticalStudyPlanBaseType***
ThisStatisticalStudyResultsInstanceQPId in ***StatisticalStudyResultsBaseType***
ThisInstanceQPId in the **Version** *element* of the ***StatisticsType***

Features.xsd
QPId in ***FeatureItemBaseType***

Characteristics.xsd
QPId in ***CharacteristicItemBaseType***

Figure 15 – QPIdType Elements

All uses of QPIds are optional. Using QPIds, however, is the only reliable method of connecting files across the workflow shown in Figure 2. It is strongly recommended that QPIds be used. A *key* in QIFDocument.xsd (when applied by an instance file processor) checks that all instances of ***QPIdType*** in an instance file have the correct format and are unique within the file.

When a downstream QIF instance file is created from an upstream QIF instance file, although the **ThisInstanceQPId** must change, if *element*s such as **characteristics**, **features**, and **product** are copied into the downstream file, the QPIds should be copied as is. If any part of a file section identified by a QPId (other than local ids) is changed after copying, the QPId should be changed, too. A changed local id should result in a changed QPId only if the local id change results in a change in the structure obtained when all local id references have been resolved.

QPIds uniquely identify QIF instance files and objects in the files. In addition, references to QPIds may be used to connect sections of QIF instance files (in different files or in the same file) as shown in Figure 16. Where QPIds are used in QIFPlan.xsd they indicate the rules that were used in developing the plan and/or the rules to be used in refining the plan. Where QPIds are used in QIFStatistics.xsd, they serve to identify results from external QIF instance files. Where QPIds are used in Traceability.xsd they serve to identify the plan that was used for **MeasurementResults** and for **Statistics**. A reference to a QPId is always made using an instance of ***QPIdReferenceType***.

QIFPlan.xsd
  RulesToUseQPId in ***MeasurementPlanType***
  RulesUsedQPId in ***MeasurementPlanType***

QIFStatistics.xsd
  ResultsQPId in ***StatisticalStudyResultsBaseType***

Traceability.xsd
  ReferencedQIFPlanInstance in ***InspectionTraceabilityType***
  ReferencedQIFPlanInstance in ***PreInspectionTraceabilityType***

Figure 16 – QPIdReferenceType Elements

The references shown in the figure are made using the ***QPIdFullReferenceType*** which has two *element*s, each of which is of ***QPIdReferenceType***. The required **ItemQPId** *element* is the QPId of the referenced item. Each optional **DocumentQPId** *element* is the QPId (found in the **Version**) of a **QIFDocument** that contains the item. For example, the **RulesToUseQPId** in a measurement plan might be as follows.

```
<RulesToUseQPId>
    <ItemQPId>ed43400a-29bf-4ec6-b96c-e2f846eb6f00</ItemQPId>
    <DocumentQPId>fd43400a-29bf-4ec6-b96c-e2f846eb6ff6</DocumentQPId>
</RulesToUseQPId>
```

The **DocumentQPId** *element* is included so that the item will be easier to find.

Another use of references to QPIds occurs in the **VersionHistory** of the ***QIFDocumentType***, which may be used to describe earlier versions of a QIF document. The information given there for each earlier version has an optional **QPIdReference** *element* of type ***QPIdReferenceType***.

6.12 Linking PMI information to product shape models

The QIF design provides for connecting detailed shape information in internal or external product models to features, characteristics, parts, and assemblies in a QIFDocument instance file. The ***PartType*** and the ***AssemblyType*** both have an optional **DefinitionInternal** *element* and an optional **DefinitionExternal** *element*.

Each part and assembly in a QIF instance file may correspond to an internal model. The internal model is as described in Part 3 of the QIF standard.

Each part and assembly in a QIF instance file may correspond to multiple external models, for example, a CAD file and two drawings.

In the case of an external CAD model, the identifiers in the model must be at least minimally persistent in the sense that each time a given file is loaded into a system that can handle it, the same identifier should be attached to each information item as is attached any other time the file is loaded. It may be assumed that identifiers in a digital drawing that is not a CAD model are persistent.

QIF supports the following types of external models for parts and assemblies in the *element*s of the **DefinitionExternal** *element*.

Parts
PrintedDrawingType
DigitalDrawingType
DigitalModelType
PhysicalModelType

Assemblies
PrintedDrawingType
DigitalDrawingType
DigitalModelType
PhysicalModelType

To connect digital drawing or digital model entities to QIF features and characteristics, the digital entities must be identified, and the association from a feature or characteristic to one or more digital entities must be made. Printed drawings and physical models do not have entity identifiers that could be connected to features or characteristics.

Identifying external digital entities is done in the **DefinitionExternal** element, which is of ***DefinitionExternalType***. The **DigitalDrawing** and **DigitalModel** *element*s of the ***DefinitionExternalType*** each have a subordinate **Entities** *element* that is a list of **Entity** *element*s of ***EntityExternalType***. The ***ExternalEntityType*** has **EntityId**, **Name**, and **Description** *element*s and an **id** *attribute* of ***QIFIdType***. The value of the **EntityId** is a string that is a persistent identifier in the external model. That string is created by the creator of the external model, not the author of the QIF file. The **id** *attribute* is used to identify the entity in other parts of the QIF instance file. For each information item in the external model that has a persistent identifier and is to be referenced elsewhere in the file, an **Entity** is defined.

The connection of an external digital entity to a feature or characteristic is made in the nominal feature or nominal characteristic, both of which contain an **EntityExternalIds** *element* that is a list of the QIF ids for the associated external entities.

Internal entities are connected to features and characteristics in exactly the same way, except that the ids of the internal entities are in an **EntityInternalIds** *element*.

An abbreviated **Part** section from a QIF instance file is shown below. It has a one model (with **id** 64). The snippet shows two **Entity**s (with **id**s 68 and 69). One of those refers to the **EntityId** "TopDistanceRightPlane", and the other refers to the **EntityId** "TopDistanceLeftPlane".

```
<Part>
  ...
  <DefinitionExternal>
    ...
    <DigitalModel id="64">
      ...
      <Entities>
        <Entity id="68">
          <EntityId>TopDistanceRightPlane</EntityId>
        </Entity>
        <Entity id="69">
          <EntityId>TopDistanceLeftPlane</EntityId>
        </Entity>
      </Entities>
    </DigitalModel>
  </DefinitionExternal>
</Part>
```

The association from a feature to one or more CAD entities is made in the **Features** section of a QIF instance file (and similarly for characteristics). This is done by listing the **EntityExternalId**s that correspond to each feature instance. Here is a snippet from a QIF instance file in which a single opposite planes feature references the planes in the preceding snippet.

```
<Features>
  ...
  <FeatureNominals>
    ...
    <OppositePlanesFeatureNominal id="34">
      ...

      <EntityExternalIds>
        <Id>68</Id>
        <Id>69</Id>
      </EntityExternalIds>
      ...
    </OppositePlanesFeatureNominal>
    ...
```

```
  </FeatureNominals>
</Features>
```

6.13 QIF handling of transforms, transformations, and coordinate systems

6.13.1 Coordinate Spaces

In QIF, the locations and orientations of all features, both nominal and actual, are defined in three-dimensional, right-handed, Cartesian coordinate spaces.

The location and orientation of nominal features can be defined in the coordinate space of the product component to which they belong allowing nominal feature information to be defined once for a part and shared across multiple instances of the same part in an assembly. Components (parts and assemblies) are combined to form the product and in the process the nominal feature information undergoes coordinate transformations until it is in the coordinate space of the root component, i.e. that of the product itself. In many situations the product will consist of a single component in which case all nominal feature information is defined in a single coordinate system.

The coordinate space of the root component is analogous to the global or world coordinate system in a CAD system: it is the de facto Cartesian coordinate space to which all other Cartesian spaces can be related.

The location and orientation of actual features are defined in this root component coordinate space. Actual feature information can be transformed from this root component coordinate space into other coordinate spaces using transformation information stored in QIF.

6.13.2 Transformation matrix

A transformation matrix is used to map (X, Y, Z) coordinates and (I, J, K) unit vectors between Cartesian coordinate spaces. Component transformations map the location and orientation of a component to its location and orientation in an assembly.

In QIF a transformation matrix takes the form of a 4 by 3 matrix consisting of a 3 by 3 rotation matrix defining real number triplets representing the orientation of one coordinate space in another, and the origin of the of one coordinate space in another defined as a real number triplet. Both the rotation matrix and the origin are optional allowing for definions of rotation-only transforms, translation-only transforms, and the identity transform (no rotation or translation) as well as combined rotation and translation transforms.

XYZ points, IJK direction vectors, and transformation matrices of local coordinate spaces can all be defined in a common coordinate system. The origin of the transform is the XYZ location of the local coordinate space in the common coordinate space. Each row of the rotation matrix is the direction of an axis of the local coordinate space in the common coordinate space.

Such a transformation matrix is defined by the ***CoordinateSystemCoreType*** which has two *elements* defining the rotation (as a 3x3 matrix) and the origin (as a triplet):

Rotation: a 3x3 matrix representing the X, Y and Z axis directions of the local coordinate space in the common coordinate space having 3 sub-*elements*:

- **XDirection**: a unit vector defining the IJK direction of the X-axis direction of the local coordinate space in the common coordinate space.
- **YDirection**: a unit vector defining the IJK direction of the Y-axis direction of the local coordinate space in the common coordinate space.
- **ZDirection**: a unit vector defining the IJK direction of the Z-axis direction of the local coordinate space in the common coordinate space.

Origin: a point defining the origin of the local coordinate space in the common coordinate space.

The three vectors defined by the **XDirection**, **YDirection** and **ZDirection** *elements* are orthonormal: each is of unit length and each is perpendicular to the other two. Furthermore they are right-handed in the sense that the right-hand vector cross product of the vector defined by the **XDirection** *element* with the vector defined by the **YDirection** *element* is equal to the vector defined by **ZDirection** *element*, the right-hand vector cross product of the vector defined by the **YDirection** *element* with the vector defined by the **ZDirection** *element* is equal to the vector defined by **XDirection** *element*, and the right-hand vector cross product of the vector defined by the **ZDirection** *element* with the vector defined by the **XDirection** *element* is equal to the vector defined by **YDirection** *element*. Or in short:

$$\|\mathbf{Xdirection}\| = 1$$

$$\|\mathbf{Ydirection}\| = 1$$

$$\|\mathbf{ZDirection}\| = 1$$

$$\mathbf{XDirection} \cdot \mathbf{YDirection} = 0$$

$$\mathbf{YDirection} \cdot \mathbf{ZDirection} = 0$$

$$\mathbf{ZDirection} \cdot \mathbf{XDirection} = 0$$

$$\mathbf{XDirection} \times \mathbf{YDirection} = \mathbf{ZDirection}$$

$$\mathbf{YDirection} \times \mathbf{ZDirection} = \mathbf{XDirection}$$

$$\mathbf{ZDirection} \times \mathbf{XDirection} = \mathbf{YDirection}$$

This results in a transformation which can translate and rotate but neither skew nor scale axis systems.

To map (or transform) a point in a local space with coordinates (x, y, z) to a point in the common space with coordinates (X, Y, Z) where the local space is defined by the transformation matrix with both rotation and translation:

$$\begin{bmatrix} X_i & X_j & X_k \\ Y_i & Y_j & Y_k \\ Z_i & Z_j & Z_k \\ O_x & O_y & O_z \end{bmatrix}$$

the following calculations are used:

$$X = (X_i)x + (Y_i)y + (Z_i)z + O_x,$$

$$Y = (X_j)x + (Y_j)y + (Z_j)z + O_y, \quad (1)$$

$$Z = (X_k)x + (Y_k)y + (Z_k)z + O_z.$$

The mapping of vectors between common and local spaces is similar. To map the vector with components (i, j, k) in local space to the vector with components (I, J, K) in common space the following calculations are used:

$$I = (X_i)i + (Y_i)j + (Z_i)k,$$

$$J = (X_j)i + (Y_j)j + (Z_j)k, \quad (2)$$

$$K = (X_k)i + (Y_k)j + (Z_k)k.$$

Conversely, to map points from common space into local space the inverse transformation is used. To map the point (X, Y, Z) in common space to the point (x, y, z) in local space the following calculations are used:

$$x = (X_i)\ (X\text{-}O_x) + (X_j)\ (Y\text{-}O_y) + (X_k)\ (Z\text{-}O_z),$$

$$y = (Y_i)\ (X\text{-}O_x) + (Y_j)\ (Y\text{-}O_y) + (Y_k)\ (Z\text{-}O_z), \quad (3)$$

$$z = (Z_i)\ (X\text{-}O_x) + (Z_j)\ (Y\text{-}O_y) + (Z_k)\ (Z\text{-}O_z).$$

And to map the vector (I, J, K) in common space to the vector (i, j, k) in local space the following calculations are used:

$$i = (X_i)I + (X_j)J + (X_k)K,$$

$$j = (Y_i)I + (Y_j)J + (Y_k)K, \quad (4)$$

$$k = (Z_i)I + (Z_j)J + (Z_k)K.$$

A transformation matrix with both rotation and translation might look like this in a QIF instance file:

```
<Transform id=”2”>
  <Rotation>
    <XDirection>0.8660254037844 0 -0.5</XDirection>
    <YDirection>0 1 0</YDirection>
    <ZDirection>0.5 0 0.8660254037844</ZDirection>
```

```
    </Rotation>
    <Origin>5.5179491924311 0.5 3</Origin>
  </Transform>
```

If the transformation matrix is a rotation only matrix then the optional **Origin** *element* is missing. In this case, the origin offset is $O_x = O_y = O_z = 0.0$ and formula (1) simplifies to:

$$X = (X_i)x + (Y_i)y + (Z_i)z,$$

$$Y = (X_j)x + (Y_j)y + (Z_j)z, \quad (1a)$$

$$Z = (X_k)x + (Y_k)y + (Z_k)z.$$

Formula (2) for the mapping of vectors between common and local spaces is unchanged.

Formula (3) for the mapping points from common space into local space simplifies to:

$$x = (X_i)\,X + (X_j)\,Y + (X_k)\,Z,$$

$$y = (Y_i)\,X + (Y_j)\,Y + (Y_k)\,Z, \qquad (3a)$$

$$z = (Z_i)\,X + (Z_j)\,Y + (Z_k)\,Z.$$

And formula (4) for mapping the vector (I, J, K) in common space to the vector (i, j, k) in local space remains unchanged.

A rotation-only transformation matrix might look like this in a QIF instance file:

```
  <Transform id="2">
    <Rotation>
      <XDirection>0.8660254037844 0 -0.5</XDirection>
      <YDirection>0 1 0</YDirection>
      <ZDirection>0.5 0 0.8660254037844</ZDirection>
    </Rotation>
  </Transform>
```

If the transformation matrix is a translation only matrix then the optional **Rotation** *element* is missing. In this case, the rotation matrix is the 3 by 3 identity matrix and formula (1) simplifies to:

$$X = x + O_x,$$

$$Y = y + O_y, \quad (1b)$$

$$Z = z + O_z.$$

If there is no rotation then vectors remain unchanged and formula (2) simplifies to:

$$I = i,$$

$$J = j, \qquad (2b)$$

$K = k.$

Formula (3) for mapping points from common space into local space simplifies to:

$x = X – O_x,$

$y = Y – O_y,$ (3b)

$z = Z – O_z.$

Because vectors remain unchanged and formula (4) simplifies to:

$i = I,$

$j = J,$ (4b)

$k = K.$

A translation-only transformation matrix might look like this in a QIF instance file:

```
<Transform id="2">
  <Origin>5.5179491924311 0.5 3</Origin>
</Transform>
```

If the transformation matrix has neither translation nor rotation then both the optional **Origin** and **Rotation** *elements* are missing and points and vectors remain unchanged in the transformation:

Formula (1) simplifies to:

$X = x,$

$Y = y,$ (1c)

$Z = z.$

Formula (3) simplifies to:

$x = X,$

$y = Y,$ (3c)

$z = Z.$

And the vector transformations are described by formulas (2b) and (4b).

A transformation matrix with neither rotation or translation might look like this in a QIF instance file:

```
<Transform id="2"/>
```


6.13.2.1 Transformation matrix example

Figure 17 shows a part with two Cartesian axis systems for the common space and a local space. The Cartesian coordinates and axis vector direction of a hole center are shown in both the common and local spaces.

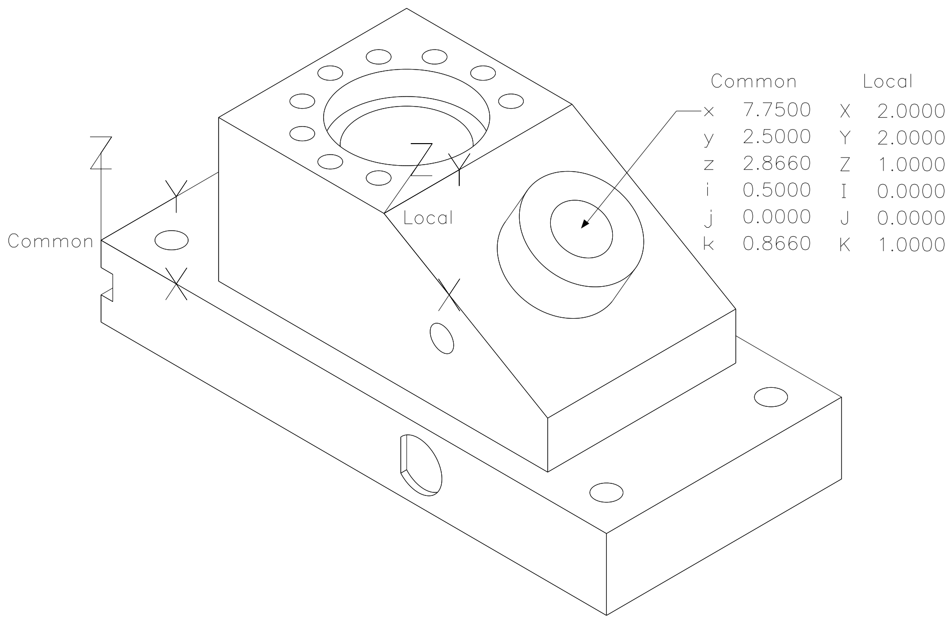


Figure 17 – Transformation matrix example.

The origin of the local space in common space coordinates is (O_x = 5.5179, O_y = 0.5000, O_z = 3.0000). The X-axis of the local space in terms of common space is (X_i = 0.8660, X_j = 0.0000, X_k = -0.5000), the Y-axis of the local space in terms of common space is (Y_i = 0.0000, Y_j = 1.0000, Y_k = 0.0000), and the Z-axis of the local space in terms of common space is (Z_i = 0.5000, Z_j = 0.0000, Z_k = 0.8660) giving the 4x3 transformation matrix:

$$\begin{bmatrix} 0.8660 & 0.0000 & -0.5000 \\ 0.0000 & 1.0000 & 0.0000 \\ 0.5000 & 0.0000 & 0.8660 \\ 5.5179 & 0.5000 & 3.0000 \end{bmatrix}$$

To transform the hole center in local coordinates (2.000, 2.000, 1.000) to coordinates in the common space the formula set (1) is used:

$$X = (0.8660)(2.0000) + (0.0000)(2.0000) + (0.5000)(1.000) + 5.5179 = 7.7500$$

Y = (0.0000)(2.0000) + (1.0000)(2.0000) + (0.0000)(1.000) + 0.5000 = 2.5000

Z = (-0.5000)(2.0000) + (0.0000)(2.0000) + (0.8660)(1.000) + 3.0000 = 2.8660

To transform the hole axis vector in local coordinates (0.000, 0.000, 1.000) to coordinates in the common space the formula set (2) is used:

I = (0.8660)(0.0000) + (0.0000)(0.0000) + (0.5000)(1.000) = 0.5000

J = (0.0000)(0.0000) + (1.0000)(0.0000) + (0.0000)(1.000) = 0.0000

K = (-0.5000)(0.0000) + (0.0000)(0.0000) + (0.8660)(1.000) = 0.8660

The transform of the hole center from common space coordinates (7.7500, 2.500, 2.8660) to local coordinates uses formula set (3):

x = (0.8660)(7.7500-5.5179) + (0.0000)(2.5000-0.5000) + (-0.5000)(2.8660-3.0000) = 2.0000

y = (0.0000)(7.7500-5.5179) + (1.0000)(2.5000-0.5000) + (0.0000)(2.8660-3.0000) = 2.0000

z = (0.5000)(7.7500-5.5179) + (0.0000)(2.5000-0.5000) + (0.8660)(2.8660-3.0000) = 1.0000

And the transform of the hole axis vector from common space coordinates (0.5000, 0.0000, 0.8660) to local coordinates uses formula set (4):

i = (0.8660)(0.5000) + (0.0000)(0.0000) + (-0.5000)(0.8660) = 0.0000

j = (0.0000)(0.5000) + (1.0000)(0.0000) + (0.0000)(0.8660) = 0.0000

k = (0.5000)(0.5000) + (0.0000)(0.0000) + (0.8660)(0.8660) = 1.0000

6.13.3 Transforms

The ***CoordinateSystemCoreType*** provides the minimum mathematical description for a coordinate transformation. The ***TransformMatrixType*** is derived from the ***CoordinateSystemCoreType*** adding the ability to define the units, accuracy, etc. of the origin point.

The ***TransformInstanceType*** is derived from the ***TransformMatrixType*** and so contains the 4x3 transformation matrix information but in addition has an **id** *attribute*, and **Name** and **Attributes** *elements*.

The **id** *attribute* allows the transform to be referenced and therefore allows a single transformation matrix to be shared by several objects.

The optional **Name** and **Attributes** *elements* allow for the transform to be identified, have arbitrary information and data added.

6.13.4 Coordinate systems

The ***CoordinateSystemType*** defines a coordinate axis system on a measurement device. A coordinate system on a measurement device typically involves aligning actual features on a real part to their nominal counterparts either using a holding fixture that physically interfaces with the actual alignment features, or by measuring the alignment features and performing the alignment mathematics in software.

In addition to aligning actual features to their nominal counterparts a coordinate system on a measurement device establishes a coordinate axis system and coordinate origin associated with the alignment. This results in two transformation matrices: one which defines the axis system on the nominal part with respect to the common space, and one which defines the axis system on the actual part with respect to the common space. The nominal and actual transformation matrices are stored in separate types to allow for multiple actual part measurements to share the same nominal transformation information.

A single coordinate system can coincide with the coordinate space of the root component. This coordinate system can be identified with the optional **CommonCoordinateSystemId** *element* of the ***CoordinateSystemListType***.

6.13.4.1 Nominal and actual transforms

The ***CoordinateSystemType*** has an optional *element*, **NominalTransform** and the ***CoordinateSystemActualTransformType*** has an *element* **ActualTransform** both of ***TransformMatrixType***, which hold the nominal and actual versions of the transformation matrix. The actual transform is stored separately on a per measurement results basis and is associated with the corresponding nominal coordinate system by id.

The meaning of and differences between these two transformation matrices is perhaps best illustrated by example.

The example in Figure 17 shows two Cartesian axis systems. In the context of QIF, if these two axis systems are established by aligning to actual features, they can become coordinate systems. The coordinate system labelled “common” is established by a set of actual part features and coincides with the root component space. Both the nominal and actual transformation matrices for this coordinate system will be the identity rotation matrix with a (0, 0, 0) origin offset:

```
  <CoordinateSystem id=”87”>
    <NominalTransform>
      <Rotation>
        <XDirection>1 0 0</XDirection>
        <YDirection>0 1 0</YDirection>
        <ZDirection>0 0 1</ZDirection>
      </Rotation>
      <Origin>0 0 0</Origin>
    </NominalTransform>
  </CoordinateSystem>
…
  <Transform>
    <ActualTransform>
      <Rotation>
        <XDirection>1 0 0</XDirection>
```

```
        <YDirection>0 1 0</YDirection>
        <ZDirection>0 0 1</ZDirection>
      </Rotation>
      <Origin>0 0 0</Origin>
    </ActualTransform>
   <CoordinateSystemId>87</CoordinateSystemId>
  </Transform>
```

or equivalently (using the ability to omit the optional **Rotation** *element* if there is no rotation, and omit the optional **Origin** *element* if there is no origin offset):

```
 <CoordinateSystem id="87">
   <NominalTransform/>
…
</CoordinateSystem>
…
  <Transform>
    <ActualTransform/>
    <CoordinateSystemId>87</CoordinateSystemId>
  </Transform>
```

If the coordinate system labelled "local" is established by a different set of actual features than those used to establish the coordinate system labelled "common" then (unless the actual part is a perfect representation of the nominal part) the actual transformation will not exactly match the nominal transformation:

```
  <CoordinateSystem id="88">
    <NominalTransform>
      <Rotation>
        <XDirection>0.866025 0 -0.5</XDirection>
        <YDirection>0 1 0</YDirection>
        <ZDirection>0.5 0 0.866025</ZDirection>
      </Rotation>
      <Origin>5.517949 0.5 3</Origin>
    </NominalTransform>
…
  </CoordinateSystem>
…
   <Transform>
     <ActualTransform>
      <Rotation>
        <XDirection>0.867865 0.009721 -0.496705</XDirection>
        <YDirection>-0.009058 0.999952 0.003743</YDirection>
        <ZDirection>0.496717 0.001251 0.867912</ZDirection>
      </Rotation>
      <Origin>5.517937 0.499941 2.999991</Origin>
    </ActualTransform>
    <CoordinateSystemId>88</CoordinateSystemId>
```

```
</Transform>
```

6.13.4.2 Alignment operations

A coordinate system on a measurement device is established using nominal and actual features by performing a set of alignment operations. This set is represented in the ***CoordinateSystemType*** by the **AlignmentOperations** *element,* which is of ***AlignmentOperationsType***. The ***AlignmentOperationsType*** is a list of **AlignmentOperation** *element*s, each of which is nominally of ***AlignmentOperationBaseType***. However, the **AlignmentOperation** *element* is the head of a substitution group of *element*s (described in the following subsection) that are of more specific types of alignment operation. *Element*s that are members of the substitution group must be used instead of **AlignmentOperation** *element*s. All of the more specific alignment operation *types* derive from the ***AlignmentOperationBaseType*** which has the required *element* **SequenceNumber** used to order alignment operations.

Any number of alignment operations can exist in a coordinate system. Only the transformation matrices (nominal and actual) of the accumulated effect of all alignment operations is stored on the coordinate system. If the transformation matrix information is required for the individual steps in a real alignment process, then a QIF coordinate system instance must be generated for each step.

6.13.4.2.1 *Alignment operation types*

The **PrimaryAlignment** *element* (which is of ***PrimaryAlignmentOperationType)*** is used to describe an alignment operation where a coordinate axis is made to align exactly with an alignment feature normal or axis.

The **SecondaryAlignment** *element* (which is of ***SecondaryAlignmentOperationType*)** is used to describe an alignment operation where a coordinate axis is made to align with an alignment feature normal or axis as exactly as possible with the constraint that a previously defined primary axis remains unchanged. In best practice the feature normal or axis of the secondary alignment feature is often nominally orthogonal to the primary axis but need not be so as long as it is not parallel to the primary axis.

The **ActualOffset** *element* (which is of ***ActualOffsetAlignmentOperationType*)** is used to describe the establishment of coordinate origins based on the location of alignment features.

The **NominalOffset** *element* (which is of ***NominalOffsetAlignmentOperationType*)** is used to describe the offset of a coordinate origin by a nominal numerical value.

The **NominalRotation** *element* (which is of ***NominalRotationAlignmentOperationType*)** is used to describe the rotation about a coordinate axis by a nominal numerical angle.

The **DatumPrecedence** *element* (which is of ***DatumPrecedenceAlignmentOperationType*)** is used to describe an alignment based on datum precedence and degrees-of-freedom rule.

The **BestFitAlignment** *element* (which is of ***BestFitAlignmentOperationType*)** is used to describe a best-fit alignment from a set of alignment features.

The **Machine** *element* (which is of ***MachineCoordinateSystemOperationType*)** is used to describe a switch to the integral coordinate system of a measurement device.

6.13.5 CAD coordinate systems

The ***CADCoordinateSystemType*** which has a **CoordinateSystemCore** *element* of type ***CoordinateSystemCoreType*** is used to define a coordinate system graphical element in a model based definition. These coordinate systems can have graphical representations and may be used for user interface purposes transforming between local and CAD global coordinates and to define sketch planes.

A CAD coordinate system may be associated with a PMI coordinate system by populating the **InternalCADCoordinateSystemId** *element* of the ***CoordinateSystemType*** with the id of an instance of ***CADCoordinateSystemType***. A similar link to a coordinate system defined in a CAD file using a non-QIF format can be established by using the **ExternalCADCoordinateSystemId** *element*.

6.13.6 Coordinate system lists

Coordinate systems representing alignments for metrology purposes are collected as a list of **CoordinateSystem** *elements* (of type ***CoordinateSystemType***) in the ***CoordinateSystemListType***. It is this type that also contains the **CommonCoordinateSystemId** *element* which identities the single coordinate system which matches the root component coordinate system equivalent to the CAD global coordinate system.

The actual transforms associated with these coordinate systems are collected as a list of **Transform** *elements* (of type ***CoordinateSystemActualTransformType***) in the ***CoordinateSystemsActualTransformsType***. Each **MeasurementResults** *element* has a sub-*element* **CoordinateSystemsActualTransforms** (of type ***CoordinateSystemsActualTransformsType***) which holds the actual transform list; the results of a single product measurement.

CAD coordinate systems representing CAD user interface coordinate system graphical elements are collected as a list of **CoordinateSystem** *elements* (of type ***CADCoordinateSystemType***) in the ***CoordinateSystemSetType***.

6.14 Feature control frames

A feature control frame like [⌖|Ø.05Ⓜ|A|BⓂ|CⓂ] typically consists of several components: the characteristic type defined by the GD&T symbol [⌖]; the tolerance zone size with zone shape, material condition and other modifiers [Ø.05Ⓜ]; and a datum reference frame consisting of datum labels with material condition or material boundary modifiers [A|BⓂ|CⓂ]. Feature control frames for form tolerances do not include a datum reference frame. The meaning of the feature control frame can be further refined using other symbols, notes or leader line modifiers like ⌀.

6.14.1 Geometric tolerance characteristic types

The GD&T symbol contained in the leftmost box of the feature control frame of a geometric tolerance defines which QIF types are to be used to describe the characteristic. The defined types follow; ***XXX*** in a type name represents ***Definition***, ***Nominal***, ***Actual***, or ***Item***,

Form characteristics:

- ○ CircularityCharacteristicXXXType
- ⌭ CylindricityCharacteristicXXXType
- ⏥ FlatnessCharacteristicXXXType
- ⏤ StraightnessCharacteristicXXXType

Location characteristics:

- ◎ ConcentricityCharacteristicXXXType
- ⌖ PositionCharacteristicXXXType
- ⌯ SymmetryCharacteristicXXXType

Orientation characteristics:

- ∠ AngularityCharacteristicXXXType
- ∥ ParallelismCharacteristicXXXType
- ⊥ PerpendicularityCharacteristicXXXType

Profile charactertistics:

- ⌒ LineProfileCharacteristicXXXType
- ⌓ SurfaceProfileCharacteristicXXXType

Runout charactertistics:

- ↗ CircularRunoutCharacteristicXXXType
- ⌰ TotalRunoutCharacteristicXXXType

In addition there is a ***PointProfileCharacteristicXXXType*** to implement a vector tolerance at a single measured point. Several point profile characteristics can be placed in a characteristic grouping to be evaluated simultaneously.

6.14.2 Tolerance zone size

The size of the tolerance zone is defined in the definition base type for all geometric tolerance characteristics: ***GeometricCharacteristicDefinitionBaseType***.

Depending on the characteristic type, the size of the tolerance zone may be subject to a bonus tolerance. This is indicated in the feature control frame by the use of a material condition modifier symbol which indicates the material condition at which the tolerance zone applies. When a particular characteristic type allows for the use of a material condition modifier, the required **MaterialCondition** *element* must be used. The **MaterialCondition** *element* can have one of four values:

REGARDLESS (RFS)	when the Ⓢ symbol appears after the tolerance
LEAST (LMC)	when the Ⓛ symbol appears after the tolerance
MAXIMUM (MMC)	when the Ⓜ symbol appears after the tolerance
NONE	when no material condition modifier symbol appears after the tolerance

In the case of orientation characteristics, the amount of bonus applied to the tolerance zone may be limited by defining the maximum tolerance size in the feature control frame. For example the size of the perpendicularity tolerance zone in | ⊥ | Ø.05Ⓜ | Ø.075MAX | A | is limited to .075 regardless of the available bonus. In QIF the optional **MaximumToleranceValue** is used to indicate the maximum size of an orientation characteristic tolerance zone.

A tolerance zone which varies in size must use the optional **ZoneLimit** *element* to define the limiting points for the variable tolerance zone. The tolerance zone starts with a value of **ToleranceValue** at the **FromPoint** and changes linearly in size to the value of **ToPointToleranceValue** at the **ToPoint**.

6.14.3 Zone shape

The shape of the tolerance zone is defined by symbols preceding the tolerance zone size in the feature control frame. These fall into three broad categories:

… | Ø.05…	Diametrical (cylindrical)
… | SØ.05…	Spherical
… | .05…	Non-diametrical

The orientation of a tolerance zone implied by the orientation of the feature to which a feature control frame is applied, or by the placement of a feature control frame on a drawing, or by a combination of the two, can be explicitly defined by the optional **ZoneOrientationVector** *element*.

The application of a position characteristic diametrical zone to an elongated feature like the round ends of a slot is indicated with the optional **ElongatedZone** *element*.

The application of a position characteristic non-diametrical zone to the boundary of a feature by use of a BOUNDARY note on the feature control frame is indicated with the optional **BoundaryZone** *element*.

A position characteristic diametrical or non-diametrical zone may apply to either a three-dimensional feature or to a two-dimensional feature (a planar section of a real three-dimensional feature). The dimensionality of the zone is defined by the optional **Dimensionality** *element*. A spherical zone is always three dimensional; so when used on a spherical zone the value of the **Dimensionality** *element* is fixed.

6.14.4 Zone extents

The extents of the tolerance zone in a feature control frame are naturally defined by the extents of the feature to which the feature control frame applies. This default behavior can be modified in the following ways: using a projected tolerance zone, defining the upper disposition of the tolerance zone, using chain lines on a drawing to define the extents of the tolerance zone, using a note like A↔B and point identifiers on the drawing, using other standard notes, or using a leader line modifier like ⌀.

The use of a projected tolerance zone indicated by the Ⓟ symbol in a feature control frame, is defined by the optional **ProjectedToleranceZone** *element*. Regardless of whether the length of the projected tolerance zone is defined numerically in the feature control frame or as a dimension on the drawing, the **ProjectedToleranceZone** *element* defines the length of the projected tolerance zone.

A profile characteristic tolerance zone is by default centered on the nominal feature to which it is applied. This behavior may be modified by chain lines on the drawing or by using the upper disposition symbol Ⓤ in the feature control frame. Both these methods are defined by the optional **OuterDisposition** *element* which defines the size of the tolerance zone outside the material which can vary from zero to the whole tolerance zone.

Zone limits defined by chain lines or by identified points used in notes like A↔B are both implemented in QIF with the optional **ZoneLimit** *element*. This element contains two sub-*elements*: **FromPoint** and **ToPoint** which together define the extents of the tolerance zone. The plane in which the zone limits are defined is given by the **NormalDirection** *element* which is the vector normal to the drawing view. Any ambiguity about the path to follow between the from-point and to-point around the part is removed by the **StartDirection** *element*. When the from-point and to-point are identified by labels on a drawing for use in a A↔B style note the optional **Name** sub-*element* on the **FromPoint** and **ToPoint** *elements* is used.

Standard feature control frame notes or leader line modifiers which control the zone extents are handled by various optional *elements* in QIF:

EACH ELEMENT	**EachElement** *element*
EACH RADIAL ELEMENT	**EachRadialElement** *element*
ALL AROUND or ⌀ ALLAROUND	**ExtentType** *element* with **ExtentEnum** sub-*element* set to ALLAROUND
ALL OVER ALLOVER	**ExtentType** *element* with **ExtentEnum** sub-*element* set to ALLOVER

6.14.5 Other Zone Modifiers

Other tolerance zone modifiers defined by GD&T symbols are shown below with their corresponding optional QIF *element*:

⟨ST⟩ statistical tolerance	**StatisticalCharacteristic**
Ⓕ free state	**FreeState**

6.14.6 Datum reference frames

The datum reference frame is defined by the required **DatumReferenceFrameId** *element* on all geometric tolerance characteristic types except for form characteristics. Because the same datum reference frame may be found in several feature control frames, a datum reference frame is instantiated from the ***DatumReferenceFrameType*** with a QIF **id**, placed in the **DatumReferenceFrames** *element*, and referenced by the feature control frame via its QIF **id**.

The datum reference frame which the **DatumReferenceFrameId** *element* references contains a hierarchy of information to define datum labels, material condition or boundary modifiers and datum precedence. In cases where a non-form geometric tolerance characteristic has no datum reference frame, the **DatumReferenceFrameId** *element* must still be present but the datum reference frame it references will be empty.

6.14.6.1 Datum Definitions

A datum definition defines a datum label and optionally associates it with datum targets or feature items. Because datum definitions can be shared among several datum reference frames or used in coordinate systems they are instantiated from the ***DatumDefinitionType*** with a QIF **id**, placed in the **DatumDefinitions** *element*, and referenced by their **id**s.

The datum label is defined by the **DatumLabel** *element*. This is typically meant to be a single datum identifier like ▶A used to identify a datum feature on a part. Compound datums like A-B that use two or more datum identifiers are only found in feature control frames and are handled in QIF with the ***CompoundDatumType***. In practice, a compound datum may be handled with the ***CompoundFeatureXXXType*** or as a constructed feature, in which case a datum label like A–B may be assigned to the compound feature.

The optional **FeatureNominalIds** *element* is used to reference the feature or features which comprise the datum. The optional **DatumTargetIds** *element* is used to reference the datum target or targets associated with the datum.

6.14.6.2 Datum with precedence

The ***DatumWithPrecedenceType*** is the mechanism by which a datum reference frame is composed of datums. Each framed box in a datum reference frame corresponds to a **Datum** *element* of type ***DatumWithPrecedenceType***.

The ***DatumWithPrecedenceType*** has the required **Precedence** *element* which is used to order datums, simple or compound, or datum features into a datum reference frame. The first datum in a datum reference frame will use the **PrecedenceEnum** *element* with a value set to PRIMARY, the second datum will used SECONDARY, the third TERTIARY, etc.

If a simple datum like A is used, then the **SimpleDatum** *element* of ***DatumType*** is chosen which references the QIF **id** of a datum definition with the **DatumDefinitionId** *element*. The **MaterialModifier** *element* is used to apply a material condition or material boundary modifier to the datum. And the **ReferencedComponent** defines whether it is the actual or nominal component of the datum feature associated with the datum that is used.

For a compound datum like A-B the **CompoundDatum** *element* of ***CompoundDatumType*** is chosen. The ***CompoundDatumType*** has two or more **Datum** *elements* of ***SequencedDatumType*** which are used to order simple datums of type ***DatumType***.

The ***DatumWithPrecedenceType*** also supports a datum feature without a datum definition and therefore no datum label. Such a construct will never be seen in a feature control frame. It is included in QIF to handle DMIS and other languages which allow a reference directly to a datum feature. For example, in DMIS:

```
T(PERP1)=TOL/PERP,0.05,FA(PLANE1)
```

As opposed to:

DATDEF/DAT(A),FA(PLANE1)

T(PERP1)=TOL/PERP,0.05,DAT(A)

When using a feature actual as a datum feature, the **ActualDatumFeature** *element* is chosen. When using a feature nominal as a datum feature the **NominalDatumFeature** *element* is chosen. Both reference the QIF **id** of a feature item but only the **ActualDatumFeature** allows for a material condition modifier.

6.14.6.3 Datum reference frame type

The ***DatumReferenceFrameType*** can have zero to five **Datum** *elements* of type ***DatumWithPrecedenceType***.

The case where zero **Datum** *elements* are present corresponds to a feature control frame with no datum reference frame, like ⌓ 0.5.

The ***DatumReferenceFrameType*** can optionally reference a coordinate system with the **CoordinateSystemId** element. The coordinate system referenced by its QIF **id** can optionally contain nominal and actual transformations. These are not to be confused with the **DRFTransformActualId** which is a reference to a transform which represents the change in the actual coordinate system associated with a mobile datum reference frame.

6.15 QIF handling of units

The QIF design seeks to handle units simply and unambiguously in instance files, especially to allow quantities to appear without explicit units specified for each value. QIF uses a scheme where primary/default and alternate units are specified once in an instance file. Quantities using the default units can occur in instance files without an explicit attribute giving the name of the unit. Alternate units can be assigned to individual quantities by including an attribute giving the name of the unit. All unit names must be unique for a given unit type.

Primary and alternate units are specified in a QIF instance file by using the **FileUnits** *element* of the QIFDocumentType. The **FileUnits** *element* is defined in Units.xsd. The **FileUnits** *element* specifies a primary unit for each of the unit types used in the instance file, and optional alternate units. If any quantity of a given unit type appears in an instance file, the corresponding unit type must appear in the **PrimaryUnits** or the **OtherUnits** of the **FileUnits**. Common XML file checkers will signal an error if this rule is violated.

For example, an instance file might give a diameter as follows:

<Diameter>7.5</Diameter>

If the **LinearUnit** in the **PrimaryUnits** is millimeter, the line above would mean that the diameter is 7.5 millimeters. This association occurs because the **Diameter** *element* is of ***LinearValueType*** in the schema.

The default unit for all unit types is the SI unit (meter, radian, kelvin, etc.). If it is desired to have a primary unit type not be a SI unit, a **UnitConversion** *element* should be included in the declaration of the primary unit. The **UnitConversion** *element* gives an **Offset** and a multiplication **Factor** that may be used to convert values of the primary unit type to values in terms of SI units.

For example, the meter is the SI unit for length. If a user wants to use the millimeter as the primary length unit in an instance file, the user puts the following lines into the **FileUnits** portion of the instance file:

```
<PrimaryUnits>
  <LinearUnit>
    <SIUnitName>meter</SIUnitName>
    <UnitName>millimeter</UnitName>
    <UnitConversion>
      <Factor>0.001</Factor>
      <Offset>0</Offset>
    </UnitConversion>
  </LinearUnit>
 ...
</PrimaryUnits>
```

In the **FileUnits** portion of the instance file, wherever a unit is declared, the name of the SI unit may be given regardless of whether it is the primary unit or not. If the **UnitConversion** is not included in the instance file, the **UnitName** just serves as an alias for the SI unit. For example if the unit type is **LinearUnit**, the **SIUnitName** must be meter if it is used, but the **UnitName** might be meter or m, or anything else the user likes. If the **UnitConversion** is included in the instance file, naming the SI unit makes it clear what units result from applying the conversion. The conversion is always accomplished using the equation:

SI = ((X plus Offset) times Factor)

where SI is the value in SI units, and X is the value in declared units.

The **FileUnits** *element* also includes an **OtherUnits** sub-*element* for specifying alternate units. For example, a **LinearUnit** named inch could be defined in the **OtherUnits** *element* as follows:

```
<OtherUnits>
  <LinearUnit>
    <SIUnitName>meter</SIUnitName>
    <UnitName>inch</UnitName>
    <UnitConversion>
      <Factor>0.0254</Factor>
      <Offset>0</Offset>
    </UnitConversion>
  </LinearUnit>
  ...
</OtherUnits>
```


If a quantity in an instance file is represented using an alternate unit, the name of the unit type must be given. If the definition for inch just given is used in an instance file, a diameter of 5 inches in an instance file would be expressed as follows:

```
<Diameter linearUnit=”inch”>5</Diameter>
```

6.16 Modeling slots in QIF

6.16.1 Introduction

The QIF library contains two feature types, opposite lines feature defined by the ***OppositeLinesFeatureXXXType*** and opposite planes feature defined by the ***OppositePlanesFeatureXXXType***, which are designed to accommodate a variety of real features commonly referred to as slots, grooves, ribs, webs or blocks. These features have the following characteristics in common: they are features of size with a center-line or center-plane about which two straight or flat sides are symmetrically opposed (the use of complex draft on an opposite planes feature may affect strict symmetry). Collectively these two feature types will be referred to as opposite sides features.

If the sides of a real feature are not flat, then the extruded cross section feature defined by the ***ExtrudedCrossSectionFeatureXXXType*** would be a more appropriate QIF feature for representing the real feature. The opposite lines feature might still be a suitable representation of a planar section of such a feature provided the criterion of symmetrically opposed straight sides is met.

The relationship between the opposite lines feature and the opposite planes feature is much like the relationship between a circle and a cylinder: the opposite lines feature is a two-dimensional planar section of a real feature and the opposite planes feature is a three-dimensional representation of a real feature.

6.16.2 Internal and external

An opposite sides feature can be either internal or external as indicated by the value of the required **InternalExternal** *element*. An internal feature is one defined by the removal of material from the bulk of a part. The center-line/plane of the feature will typically be in open space with the surface normal vectors of the opposite sides pointing generally in the direction of the center-line/plane. These real features are commonly referred to as slots or grooves. Conversely, an external feature is one defined by the bulk of a part. The center-line/plane of the feature will typically be inside the material of the part with the surface normal vectors of the opposite sides pointing generally in the direction away from the center-line/plane. These real features are commonly referred to as ribs, webs or blocks.

6.16.3 Location and size

The location and orientation of an opposite lines feature is defined by the center-line represented by a center-point and the axis unit vector defined by the **StartPoint** and **Vector** *elements* respectively of the required **CenterLine** *element*, and a unit vector representing the normal of the plane in which the feature lies defined by the required **Normal** *element*. The size of the opposite lines feature is given by the required **Width** *element* which applies in a direction perpendicular to the center-line axis in the plane of the feature. The feature may also have a length defined by the optional **Length** *element* which applies in the direction of the center-line axis.

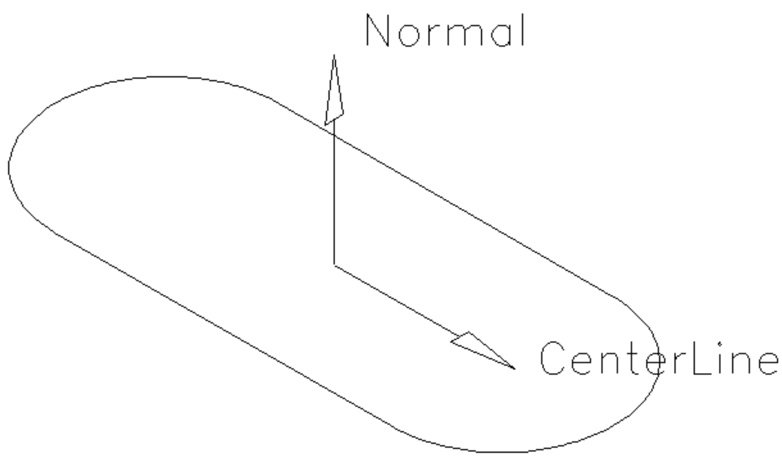


Figure 18 – An opposite lines feature with round closed ends.

Figure 18 shows the relationship between the **CenterLine** *element* which defines both the location and the orientation of the axis of the round-ended slot, and the **Normal** *element* which defines the plane in which the feature lies.

The location and orientation of an opposite planes feature is defined by the center-plane defined by a center-point and the center-plane's normal unit vector given by the **Point** and **Normal** *elements* of the required **CenterPlane** *element*. The size of the opposite planes feature is defined by the required **Width** *element* which applies in the direction along the center-plane normal vector. The feature may also have a length defined by the optional **Length** *element* which applies in the direction of the co-requisite **LengthVector** *element*. Furthermore, the feature may have a depth defined by the optional **Depth** *element* which applies in the direction of the co-requisite **DepthVector** *element*.

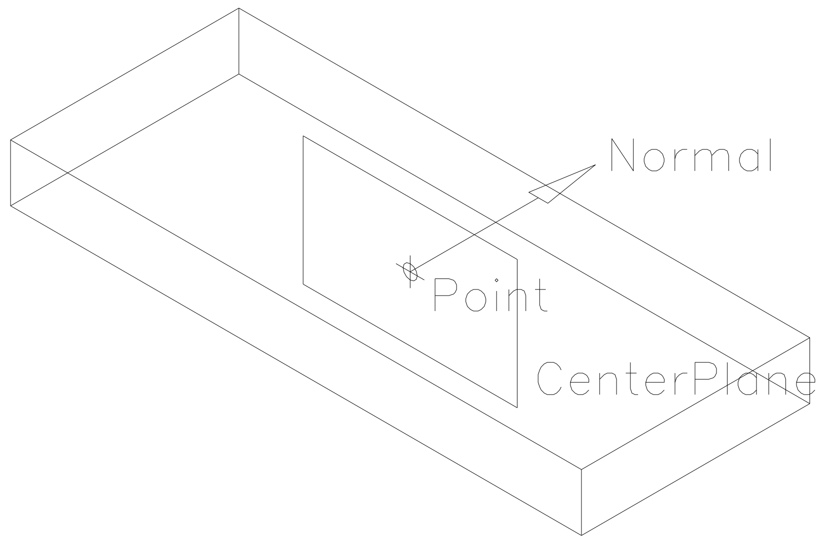


Figure 19 – An opposite planes feature with flat closed ends.

Figure 19 shows the relationship of the **CenterPlane** *element* with sub-elements **Point** and **Normal** which define the orientation of the main sides of a flat-ended slot. In order to define the orientation of the ends of the slot, the optional **LengthVector** *element* must be specified (not shown in Figure 19). The **LengthVector** *element* would have the same orientation as the **CenterLine** in Figure 18.

For both opposite lines and opposite planes features with closed ends the length is from the material boundary at one end to the material boundary at the other end measured along the center-line vector or along the length vector in the center-plane. The length is not between radius centers when a feature has rounded ends. When a feature has one or two open ends then the length can be the distance to a virtual material boundary equivalent to the real material boundary created by placing a flat block over the open end, or it can be an indication of the measureable region of the feature. In all cases the length is symmetrically disposed about the center-point of the feature.

6.16.4 End types

The opposite sides feature types also have an end-type as defined by the required **EndType** *element*, a choice between one of the enumerated values of the **SlotEndEnum** *element*: either ROUND, FLAT, OPEN or UNDEFINED, and a user-defined string in the **OtherSlotEnd** *element*. If the end-type of the opposite sides feature is unknown then the UNDEFINED enumerated end-type is used. If the end-type is known but not covered by any of the enumerated end-types, then a string is used to describe the end-type.

Figure 18 and Figure 19 show examples of the ROUND and FLAT end types respectively. Figure 22 below shows an example of the OPEN end type (upper right).

The shape of the ends of a ROUND and FLAT opposite sides feature can be further modified by using the optional **EndRadius1** and **EndRadius2** *elements* which apply in the directions against and along the center-line axis/length vector respectively. The default condition for round-ended features is to have a circular end tangent to both sides (the actual ends may be described by circular arcs for opposite lines or by cylindrical or conical segments for opposite planes but in a cross-section at any depth the circular ends will be tangent to both sides). By using the end radii *elements*, circular cross-section ends that are not tangent to the sides can be specified. The size of the end is given by the **EndRadius** *element* which must be larger than the radius of a tangent end. Whether the end expands beyond the width like a dumbbell shape or not is given by the value of the optional **Expanded** *element*.

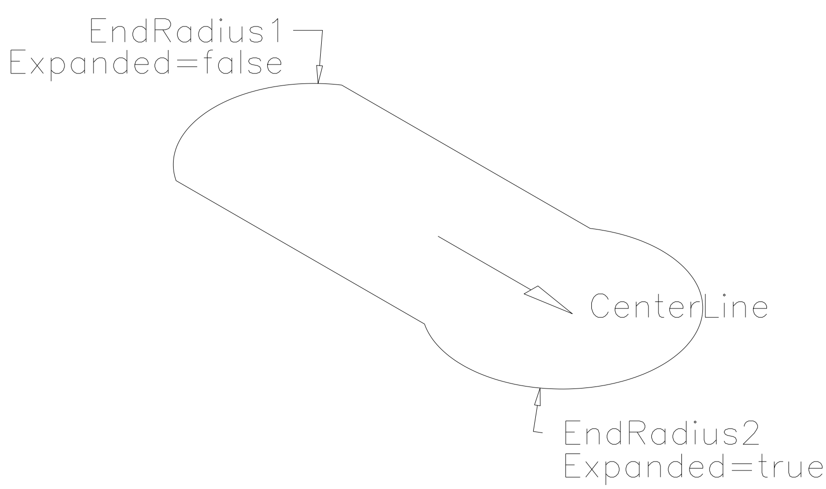


Figure 20 – A slot with non-tangent round ends.

Figure 20 shows a round-ended slot with non-tangent ends. The relationship between the ends modified by the **EndRadius1** and **EndRadius2** *elements* and the **LengthVector** or **CenterLine** *elements* is shown. In the example the values of the two end radii are equal, but one is expanded and the other is not. The center of the slot is midway between the extremes of the slots in the axis direction, and not at the midpoint of the centers of the circular ends.

For flat ended opposite sides features the end radii can be used to apply a fillet radius to an otherwise flat end. The end radius must be small enough to leave a portion of the flat end.

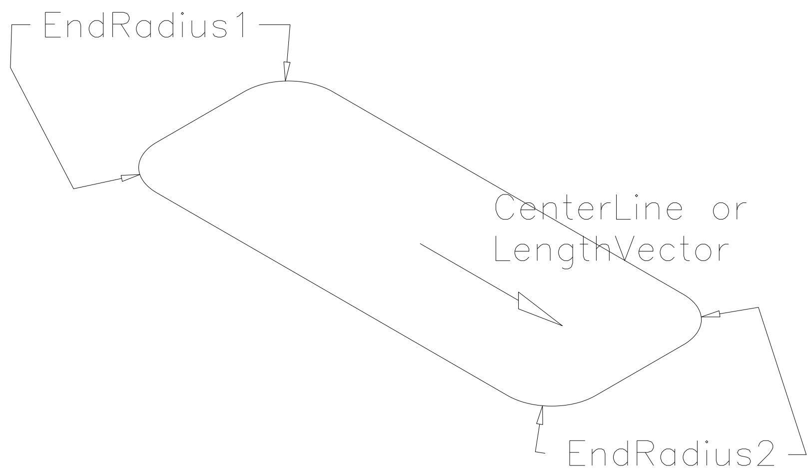


Figure 21 – A flat-ended slot with rounded corners..

Figure 21 shows a flat-ended slot with rounded corners. In the example the two end radii values are equal (and both must be smaller than half the width of the slot).

An opposite sides feature may have one closed end and one open end. The presence of a single open end is indicated by the optional **SingleOpenEnd** *element*. In the case of an opposite lines feature the center-line axis must point towards the open end, and for the opposite planes feature the co-requisite **LengthVector** *element* points toward the open end.

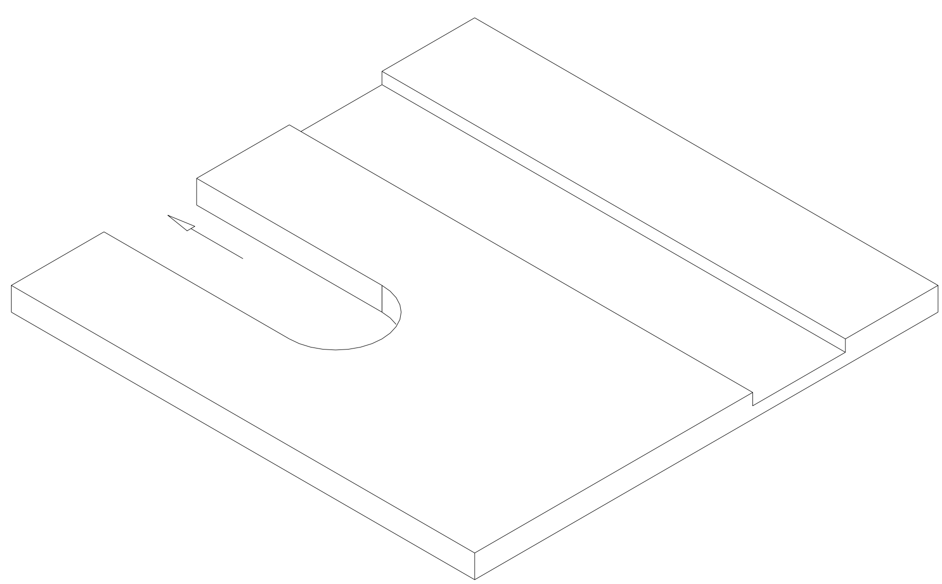


Figure 22 – Opposite planes features with open ends.

Figure 22 shows two slots with open ends. The lower left slot would have end type ROUND and use the optional **SingleOpenEnd** *element*. The arrow shows the direction of the **LengthVector** *element* towards the open end. The upper right slot would have end type OPEN because both ends of the slot are open.

6.16.5 Bottom types

The opposite planes feature can also define a bottom type with the optional **Bottom** *element*, a choice between one of the enumerated values of the **BottomEnum** *element*: either BLIND, THROUGH or UNDEFINED, and a user-defined string in the **OtherBottom** *element*. If the bottom-type of the opposite planes feature is unknown then the UNDEFINED enumerated bottom-type is used. If the bottom-type is known but not covered by any of the enumerated bottom-types, then a string is used to describe the bottom-type.

The slot in the lower left of Figure 22 is an example of the THROUGH bottom type; the slot in the upper right is an example of the BLIND bottom type.

6.16.6 Taper

An opposite sides feature may be tapered. If the optional **TaperAngle** *element* is populated with a non-zero value then the width of the opposite sides feature changes along the center-line vector/length vector. The sign of the taper angle defines whether the feature gets larger in the direction of the vector (positive) or gets smaller in the direction of the vector (negative). When an opposite sides feature is tapered the width applies at the center-point.

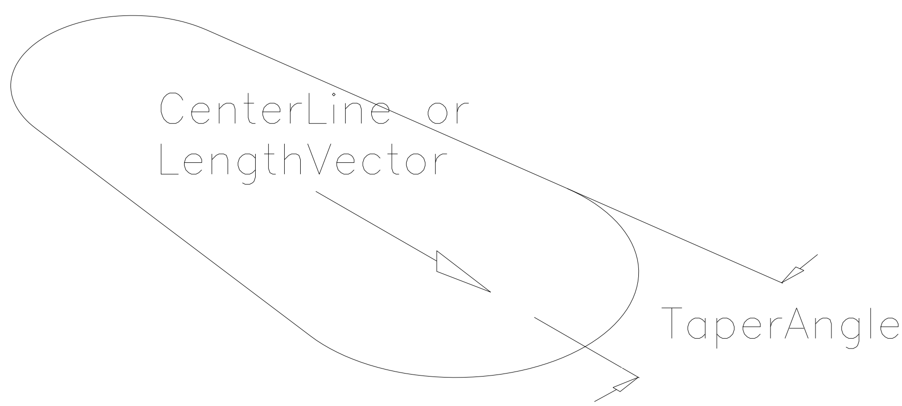


Figure 23 – A tapered slot.

Figure 23 shows the relationship between the taper angle and the center line or length vector of a tapered slot. The example shows a slot with a ROUND end type. The center of the slot is midway between the extremes of the slots in the axis direction, and not at the midpoint of the centers of the circular ends.

6.16.7 Draft

The opposite planes feature may have a draft angle defined by the optional **DraftAngle** *element*. A positive draft angle means the feature will open up (get larger) in the direction defined by the co-requisite **DepthVector** *element* or the co-requisite **DraftVector** *element*, a negative draft angle means the feature will close up (get smaller). When an opposite planes feature is drafted the width and length apply at the center-point. The **DraftVector** *element* overrides the **DepthVector** *element* when the draft vector is not perpendicular to the length vector, the axis vector, or both.

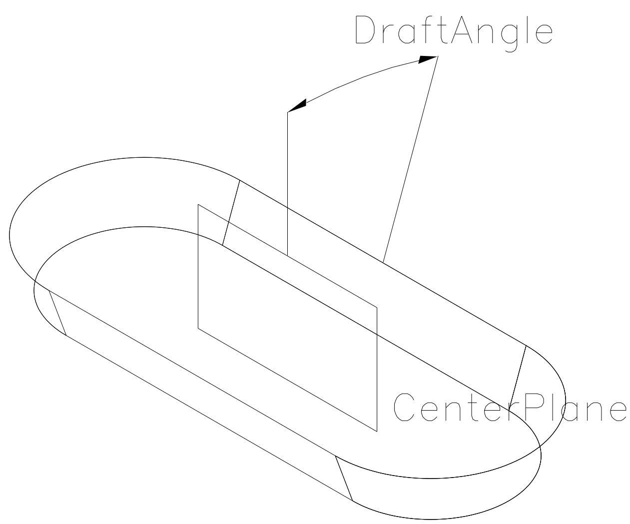


Figure 24 – A slot with draft.

Figure 24 shows the relationship between the draft angle and the center plane of a slot with draft.

6.16.8 Feature actual

The above descriptions of required, optional and co-requisite data were particular to nominal opposite sides features. All data elements for actual features are optional to handle different use cases.

In addition to actual elements which correspond directly with nominal elements like actual width and length corresponding with nominal width and length respectively, there are actual data elements for minimum and maximum of size values.

6.17 Modeling cones and conical segments in QIF

6.17.1 Introduction

The QIF library contains two feature types, cone feature defined by the ***ConeFeatureXXXType*** and conical segment feature defined by the ***ConicalSegmentFeatureXXXType***, for defining features with a single, conical surface. The cone feature is meant to describe a feature of size; a feature to which a location characteristic may apply which in turn may acquire bonus from a material condition modifier. The conical segment feature is meant to describe a feature which is not a feature of size; a feature to which a profile characteristic or a radius characteristic at a given cross section may be applied.

The *elements* necessary to describe these two feature types are the same with the exception that the sweep extents for a cone feature are optional while for a conical segment they are required. In the case of a cone feature the optional **Sweep** *element* might be used to exclude a portion of the conical surface to avoid a keyway. For a conical segment the required **Sweep** *element* defines the extent of the feature which may be the corner of a rounded pocket with draft.

6.17.2 Location, orientation and angle

The minimum information necessary to define the conical surface is its location, orientation and angle. Rather than forcing the location for the cone to be its vertex, the location of cone can be defined anywhere along its axis at a specified diameter. The reason for this is twofold. If the angle of a cone is very small so that the cone is almost cylindrical, the vertex is unstable with respect to small changes in angle. This instability can be removed by defining the location at a diameter rather than at the vertex. The feature to which a cone substitute feature fitting algorithm is to be applied may in fact nominally be a cylinder in which case the location of the vertex is indeterminate.

The locating point for a cone or conical segment is defined by the **AxisPoint** sub-*element* of the **Axis** *element*. The orientation of the cone or conical segment is defined by the **Direction** sub-*element* of the **Axis** *element*. The cone's axis direction defined by the **Direction** sub-*element* always points towards the widening end of the cone The **Diameter** *element* is the diameter of the cone in a plane perpendicular to the direction vector at the axis point. The cone or conical segment can be defined by either its full included angle (the angle between opposite sides) or its half angle (the angle between a side and its central axis) by choosing between the **FullAngle** or **HalfAngle** *element* respectively.

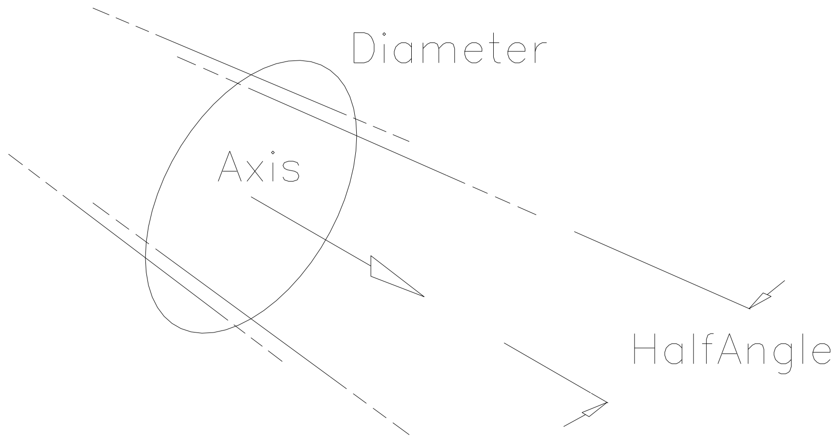


Figure 25 – An unbounded cone located at a reference diameter and defined by its half angle.

Figure 25 shows a cone with its location defined at a diameter and with its half angle. This cone definition is stable at small angles and handles the degenerate case of a cylinder.

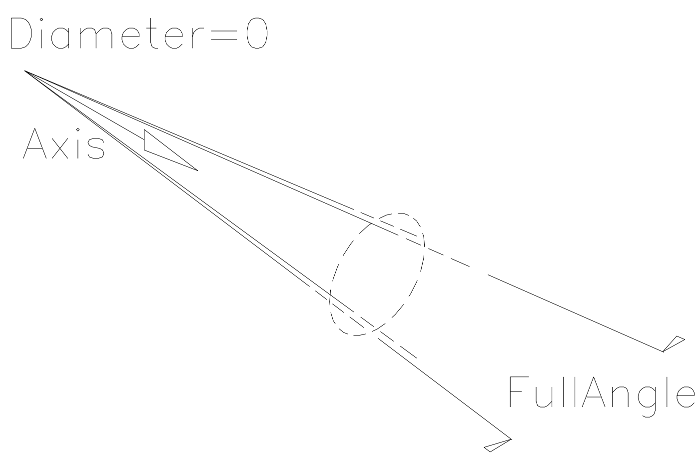


Figure 26 – An unbounded cone located at its vertex and defined by its full angle.

Figure 26 shows a cone with its location defined at its vertex and with its half angle. This cone definition is equivalent to that of many CMM systems including DMIS.

6.17.3 Linear extents

The simple definitions illustrated in Figure 25 and Figure 26 show unbounded cones with undefined extents in the direction away from the vertex. The linear extents of a cone or conical segment along its axis can be defined with the optional **LargeEndDistance** and **SmallEndDistance** *elements*. These distances can be positive or negative depending on their relationship with the locating point. If the end is along the axis vector from the locating point then the distance will be positive. If the end is in a direction against the axis vector from the locating point then the distance will be negative.

If both the optional **LargeEndDistance** and **SmallEndDistance** *elements* are given then the cone is a conical frustum (or a segment of a frustum): a truncated cone without a pointed end. If the optional **LargeEndDistance** *element* is present but the **SmallEndDistance** *element* is missing then the cone or conical segment has a pointed end.

The effect of the locating point on the sign and value of the **LargeEndDistance** and **SmallEndDistance** *elements* is shown in Figure 27, Figure 28, Figure 29, and Figure 30. Not shown is the case where the diameter at the locating point is larger than either the small or large ends in which case both distances would be negative. (In all diagrams the required full or half angle element is not shown.)

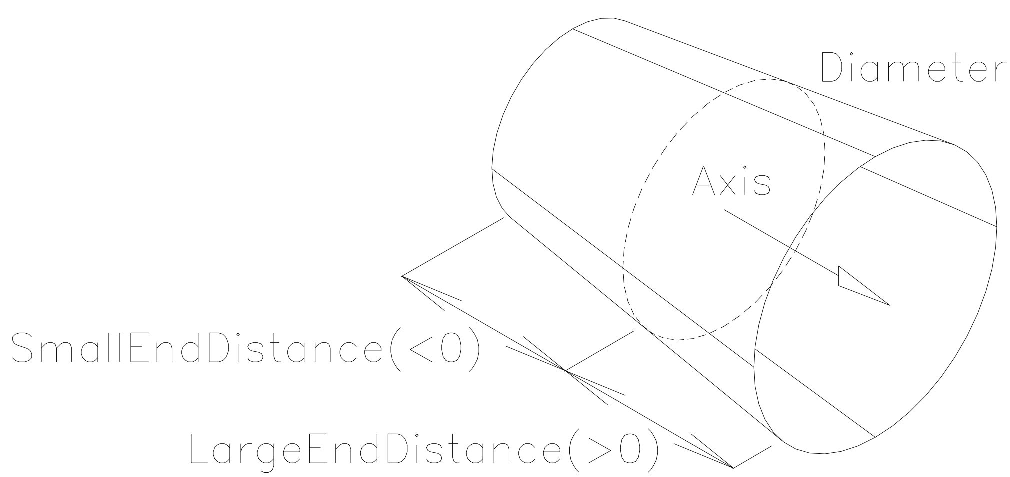


Figure 27 – A bounded, truncated cone located at a reference diameter

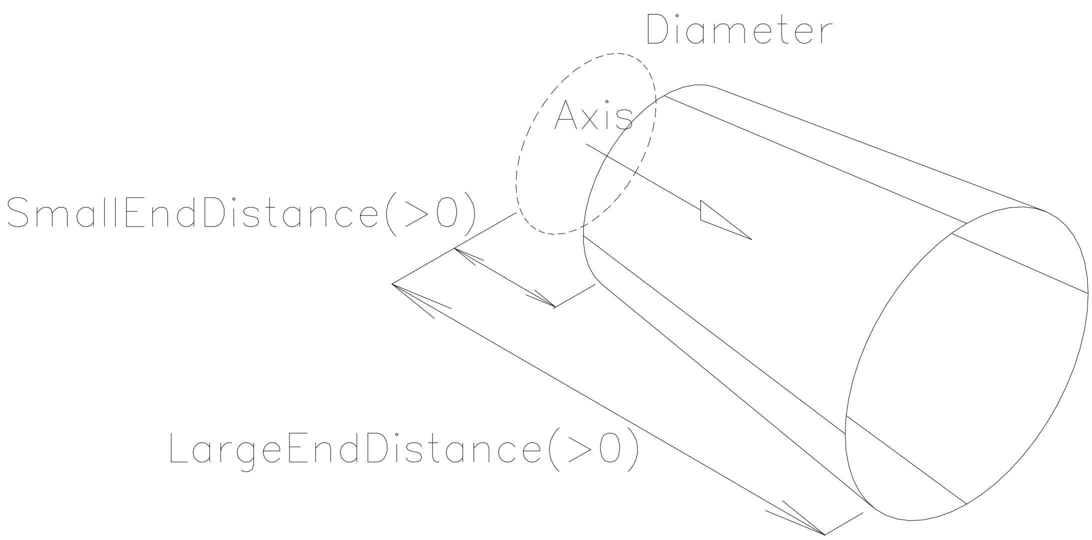


Figure 28 – A bounded, truncated cone located at a virtual reference diameter

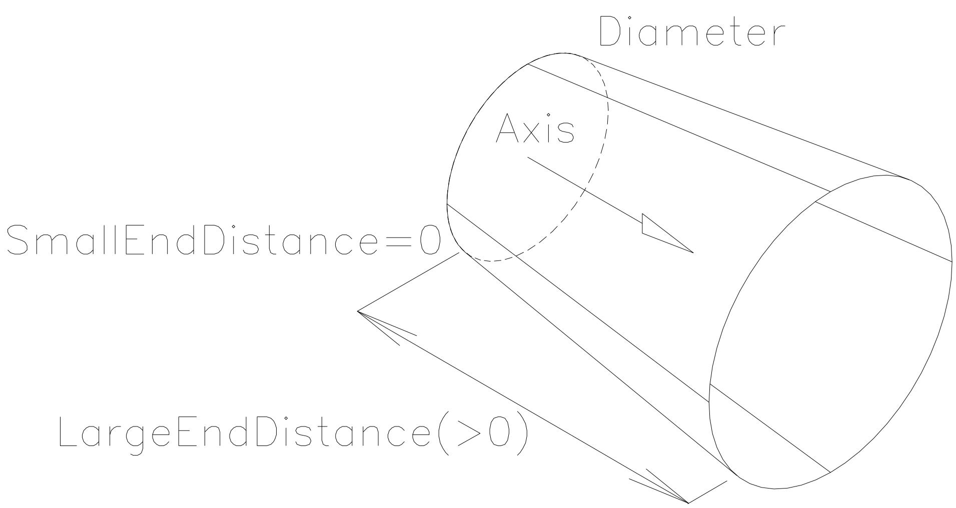


Figure 29 – A bounded, truncated cone located at its small end

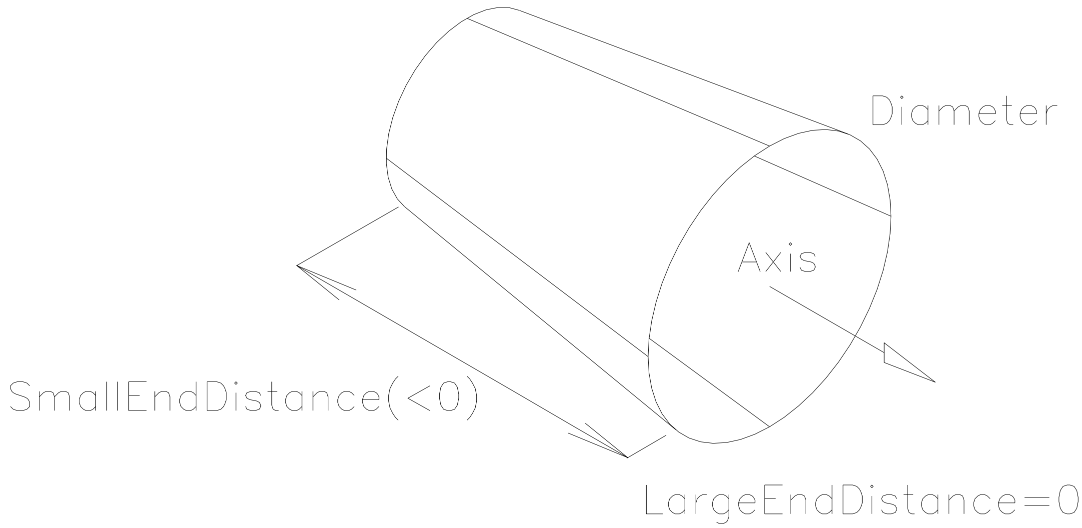


Figure 30 – A bounded, truncated cone located at its large end

Figure 31 and Figure 32 show the bounding of a cone defined with its locating point at its vertex.

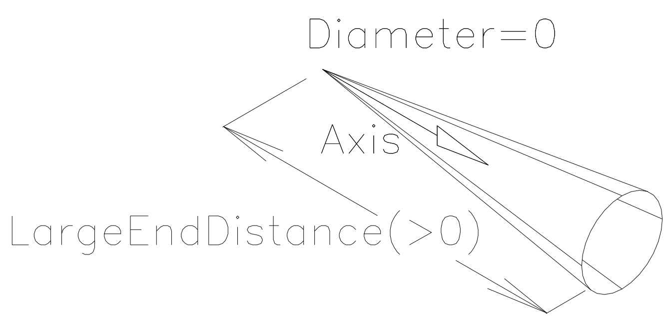


Figure 31 – A bounded pointed cone located at its vertex

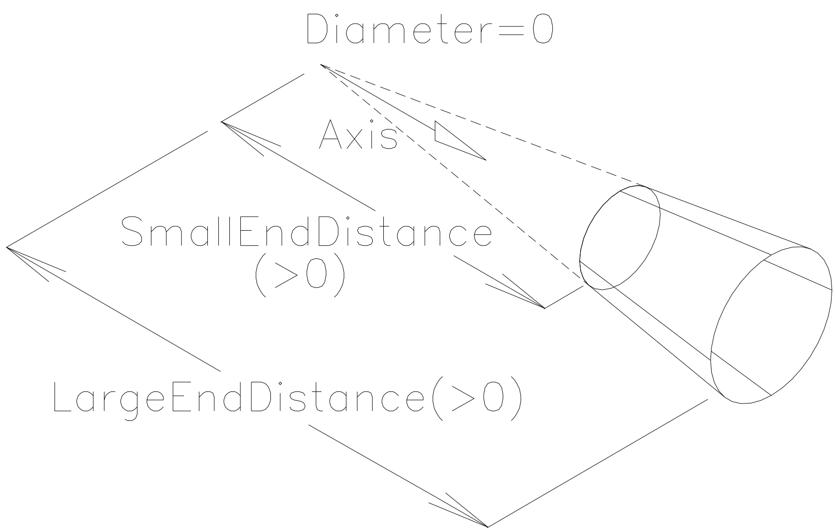


Figure 32 – A bounded truncated cone located at its vertex

6.18 Modeling threads in QIF

Features and characteristics often have elements that essentially define the same information. For example a cylinder has a nominal **Diameter** *element* on the ***CylinderFeatureDefinitionType*** and a diameter characteristic has a nominal **TargetValue** via the ***LinearCharacteristicNominalBaseType*** from which ***DiameterNominalCharacteristicType*** derives. Both the diameter and the target value are nominal representations of the size of the feature. Most often these two values will be identical but use cases exist where the manufacturing size and the tolerance evaluation size might be different. For example if a diameter has a specified size of 10.0 +0.5/-0.0 and the manufacturing process targeted the nominal value of 10.0 any undersize variation would result in an out of tolerance condition.

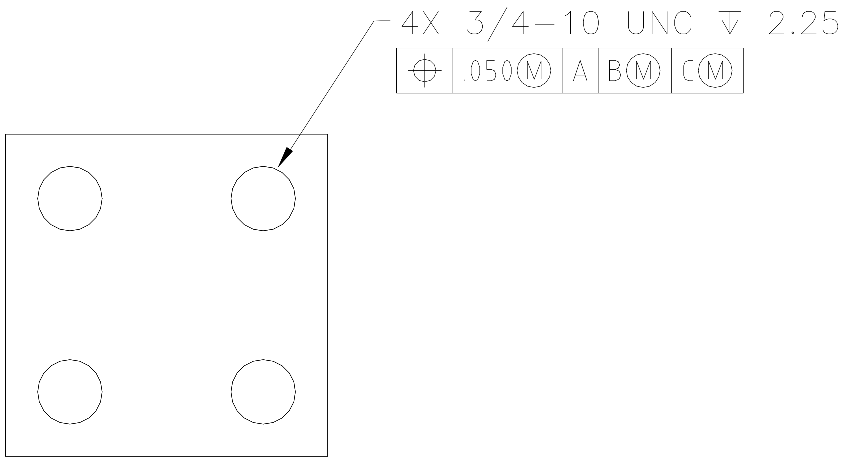


Figure 33 – Threaded features.

This is not the case with threaded features. A hole is not threaded to one specification and inspected to another. Also, unlike a simple diameter, the data associated with a thread specification is quite complex as shown in Figure 33. To allow the sharing of thread specification data the threaded feature defined by the ***ThreadedFeatureDefinitionType*** and the thread characteristic defined by the ***ThreadCharacteristicDefinitionType*** both have an *element* **ThreadSpecificationId** which references an instance of ***ThreadSpecificationType*** contained in a list of ***ThreadSpecificationsType***.

6.18.1 Thread specification types

The ***ThreadSpecificationType*** provides a choice between three *elements*: **SingleLeadSpecification** of ***SingleLeadSpecificationType***, **MultiLeadSpecification** of ***MultiLeadSpecificationType***, and **TextThreadSpecification** of ***TextThreadSpecificationType***. The single lead and multi-lead types allow for the unambiguous capturing of detailed information in their elements, some of which resolve to enumerations based on industry standards. If these pre-defined data containers are not sufficient to capture a thread specification then that specification may be entered as a simple text string via the **TextSpecification** *element* of ***TextThreadSpecificationType***.

6.19 Feature actual determination

The feature nominal is a representation of the design intent. The feature actual is determined from an actual physical part. The QIF information model allows for different levels of detail for the information related to how a feature actual is determined.

6.19.1 Checked and set features

At the most elementary level, the QIF information model distinguishes between a feature actual which is checked and one which is set. A feature called out on a print may not be measurable.

For example, a circle representing the region on the surface of a part which is inaccessible because of interference with a fixture net pad is not a measureable circle. But because the feature appears on a printed part drawing there may be the requirement for that feature to be reported. To accomplish this, the feature actual is set to its nominal value. More commonly, a feature will be checked by a measurement device or otherwise be evaluated using measurement data.

For a set feature, the **Set** *element* of the **DeterminationMode** *element* on the feature item is chosen. If the feature is checked then the **Checked** *element* of the **DeterminationMode** *element* on the feature item is chosen. The differentiation between set and checked features is the only required information from the **DeterminationMode** *element*.

6.19.2 Measurement and construction

More information about how a feature is checked can be added optionally with the **CheckDetails** *element*. A checked feature can be either measured or constructed. A measured feature is evaluated using data collected directly for that feature by a measurement device. A constructed feature is evaluated using data previously collected for other features referred to as base features.

For a measured feature, the **Measured** *element* of the **CheckDetails** *element* is chosen. For a constructed feature, the **Constructed** *element* of the **CheckDetails** *element* is chosen.

6.19.3 Measurement points

Details on the points used for measuring a feature can be added optionally with the **PointList** *element*. This *element* contains an ordered list of nominal target points with optional information about the measurement device and sensor used to collect each point.

6.19.4 Construction methods

Further detail about a feature construction can be optionally added by choosing between the construction methods for a given feature type.

Common to all construction methods is the optional **NominalsCalculated** *element*. This Boolean *element* is used and set to true when feature nominals are calculated from the nominals of the set of base features rather than being specified directly. This may come about if a generic macro or subroutine is used to perform a feature construction where both the feature nominal and feature actual are determined from the input base features.

A base feature used in a construction is a reference to another feature item by its id in the **FeatureItemId** *element*. In addition the referenced component of the base feature is identified by the **ReferencedComponent** *element*. Usually the actual component, but sometimes the nominal component, of a base feature is used in the construction of the feature actual. This is not to be confused with the concept of calculated nominals, where the nominal components of all base features are used to calculate the feature nominal.

The recompensated construction method references the measurement points of a base feature rather than the base feature itself. The **BaseFeaturePointList** *element* identifies an ordered list of measurement points on referenced base features. These points are identified by a single point index, a range of point indexes, or all the points on the specified base feature. Point indexes are integers that begin at 1 for the first point in a measurement point list.

The individual construction methods are described in FeatureTypes.xsd. The description includes the number and types of base feature references required for the particular construction types.

The descriptions of the various construction methods can be found in *Part 2: QIF Library - Information Model and XML Schema Files*. Some construction method types are similar; these methods are clarified and differentiated in the following sections.

6.19.4.1 Best-fit and recompensated construction methods

Common to many feature types are the best-fit construction method indicated by choosing the **BestFit** *element* in the **Constructed** *element*, and the recompensated construction method indicated by choosing the **Recompensated** *element* in the **Constructed** *element*. Both methods use a substitute feature fitting algorithm to determine the feature actual from point data. In the case of best-fit construction the point data is that from point reducible base features. The point-reduced data is compensated for probe size and a substitute feature algorithm may have been applied, such as determining the center of a circle. For recompensated construction the point data is uncompensated raw measurement points from base features. The result is that a recompensated constructed feature will be the same as if it were measured with the same points.

6.19.4.2 Extract and from-scan construction methods

The extract construction method indicated by choosing the **Extract** *element* in the **Constructed** *element* and from-scan indicated by choosing the **FromScan** *element* in the **Constructed** *element* are both used to determine a feature actual from a set of measurement points. These two methods are similar to the recompensated construction method in that all three create the feature actual using a substitute feature algorithm on raw uncompensated point data from base features. But the extract and from-scan construction methods differ from the recompensated construction method in that even though the base feature can be identified, the point indexes on that feature cannot because the feature is typically measured with a scanning device resulting in a large point set variable in both size and density.

Both the extract and from-scan construction methods use a subset of points from the base feature. This subset is determined by the nominal extents of the feature being constructed. Therefore, the feature being constructed must be naturally bounded (e.g., an arc) or explicitly bounded (e.g., a plane with a polyline boundary).

The extract construction method is used when the dimensionality of the base feature and the feature being constructed are the same. Two-dimensional features can be extracted from a two-dimensional scan curve in the same plane; arcs and lines can be extracted from a planar curve scan inside a filleted pocket because the arcs and lines are coincident with the base feature. Similarly, three-dimensional features can be extracted from a three-dimensional scan surface. In both cases, the measurement points used in the construction are those inside the limits on the bounded feature.

But when a two-dimensional feature is derived from the measurement points of a three-dimensional scanned surface, the feature being constructed may not be coincident with the scanned data. As a result, search windows must be used to extend the two-dimensional feature out of its plane in order to capture point data sufficient to determine the feature actual. The from-scan construction methods have *element*s like **SearchRadius** which allow for the definition of the search windows. (Some three dimensional feature types also have a from-scan construction method to be compatible with the DMIS 5.2 CONST (Input format 15) statement.)

6.19.4.3 Copy, cast, and transform construction methods

The copy,cast and transform construction methods perform the similar operation of copying actual data from one feature actual to another feature actual. The **NominalIsCalculated** *element* is used to control the copying of nominal feature data.

In the case of the copy construction method the base feature and the feature being constructed must be of the same type and all the actual data elements from the base feature are copied unchanged to the feature being constructed. If the **NominalIsCalculated** *element* is present and set to true then the nominal data elements are similarly copied from the base feature to the feature being constructed.

In the case of the cast construction method the base feature and the feature being constructed are not of the same type and all the actual data elements from the base feature may or may not have corresponding data elements in the feature being constructed. Only those actual data elements shared between the two feature types are copied unchanged from the base feature to the feature being constructed. If the **NominalIsCalculated** *element* is present and set to true then any shared nominal data elements are similarly copied from the base feature to the feature being constructed.

In the case of the transform construction method the base feature and the feature being constructed must be of the same type and all the actual data elements from the base feature are copied to the feature being constructed. All location and orientation elements are transformed by the actual transform matrix of the specified coordinate system. All size, form, and other dimensional data remain unchanged. If the **NominalIsCalculated** *element* is present and set to true then the nominal data elements are similarly copied from the base feature to the feature being constructed with the nominal location and orientation data being transformed by the nominal transform matrix of the specified coordinate system.

6.20 Key characteristics - encoding "balloon" numbers in QIF

Figure 10 shows a plate with ballooned tolerances. Characteristics that are key to a manufacturing process or to a part’s usability are often indicated on a drawing with an identifier in a circle or another shape. These balloon numbers are accommodated in QIF with the concept of a key characteristic.

The key characteristic, defined by the ***KeyCharacteristicType***, performs two tasks. First, the required **Designator** *element* captures the balloon number. Second, the optional **Criticality** *element* is used to indicate the level of importance of the characteristic. This is a user-defined token and may have any value. Examples include MAJOR, MINOR and KEY.

An *element* **KeyCharacteristic** of ***KeyCharacteristicType*** can be found in three locations in the characteristics aspect hierarchy: on the definition, on the nominal and on the item. This allows for the sharing of a key characteristic designator among several instances of a characteristic.

In Figure 10 the balloon number “1” is shared by the four holes in much the same way that the nominal diameter is shared. This relationship can be shown by using the optional **KeyCharacteristic** *element* on the characteristic nominal. This is the normal location for indicating the key characteristic designator and criticality except in special circumstances.

When the key characteristic designator is shared among several items and it is necessary to assign augmented labels or balloon numbers to each item, this is accomplished with the **KeyCharacteristic** *element* on the characteristic item. A balloon number of “1” may result in individual key characteristic designators like 1_1, 1_2, etc.; 1A, 1B, etc.; or 1.1, 1.2, etc. depending on company standards.

When a box tolerance is used (like that shown in section 6.7.4) and such a tolerance is ballooned then, in practice, that designator may be shared among several different characteristics and even different characteristic types. In this case, the **KeyCharacteristic** *element* on the characteristic definition is used. If the designator spans several different characteristic types then the same key characteristic must be re-defined for each characteristic types; QIF does not allow for characteristic definitions to be shared among characteristics of different types.

6.21 Attributes and Part Notes

The QIF information model enables users to insert otherwise unmodeled information in many places in QIF instance files by using **Attributes** and/or **PartNotes**. Note the capital **A** and bold font on **Attributes**; this is entirely different from XSDL *attribute*s.

An **Attributes** *element* is a list of **AttributeXXX** *element*s where the **XXX** indicates the type of data. The full list of **AttributeXXX** *element*s is shown in Figure 34.

Element Name	Data Type
AttributeBool	Boolean
AttributeQPId	QPIdType
AttributeI1	integer
AttributeI2	*list* of two integers
AttributeD1	double
AttributeD3	*list* of three doubles
AttributeStr	string
AttributeUser	user defined data in a binary array or XML structure

Figure 34 – Attribute element names and types

Attributes are intended to be used to convey important information that is not representable elsewhere in the QIF model. Items that can be represented in the model (the nominal diameter of a circle, for example) should not be put into **Attributes**. The **AttributeStr** *element* can be used to convey any sort of information in natural language. The types that have an **Attributes** *element* in the QIF model are shown in Figure 35. Where there is an **Attributes** *element* in a base type, all derived types will have one, too.

Characteristics.xsd
CharacteristicBaseType

Features.xsd
FeatureBaseType

IntermediatesPMI.xsd
AngularToleranceDefinitionType
CoordinateSystemType
DatumDefinitionType
DatumReferenceFrameType
DatumTargetDefinitionBaseType
LinearToleranceDefinitionType
TransformInstanceType

PrimitivesPD.xsd
NodeWithIdBaseType

QIFDocument.xsd
QIFDocumentType

QIFMeasurementResources.xsd
QualificationType
TemperatureType

QIFProduct.xsd
ComponentType
ProductDefinitionBaseType

QIFResults.xsd
ActualComponentType

QIFStatistics.xsd
StatisticalStudyPlanBaseType
StatisticalStudyResultsBaseType

Statistics.xsd
CharacteristicStatsEvalType
AssignableCauseType
CorrectiveActionType

Traceability.xsd
EnvironmentType
ManufacturingProcessTraceabilityType

Figure 35 – Types with Attributes element

The ***PartNoteType*** models a note that needs to be displayed graphically and may contain only text. The ***ProductType*** has a **PartNoteSet** *element* of ***PartNoteSetType*** that is a list of **PartNote** *elements* of ***PartNoteType***. These are referenceable by id. The ***ProductDefinitionBaseType*** has a **PartNoteIds** *element* that is a list of ids of part notes. This allows part notes to be connected to specific products. The ***PartNoteType*** also has a **PartNoteIds** *element* so that part notes may be nested.

7 Detailed requirements

7.1 XML naming and design rules (NDR)

XML technology was chosen for QIF encoding because the basic XML specifications are supported as open, public domain, royalty-free standards, and because XML technology is very widely used.

The QIF information model is built using the XML Schema Definition Language (XSDL). That language was chosen because:

- XSDL has adequate expressive power for the basic structure of the model. It includes the ability to define complex types with *attributes* and *elements*.
- XSDL allows more specialized complex types to be derived from less specialized complex types, so that type hierarchies can be defined.
- XSDL has built-in data types and the ability to specialize them.
- XSDL permits the modular construction of models via an "include" capability.
- XSDL has a default instance file format (XML) with a set of rules for determining if an instance file conforms to a model. Moreover, XML instance files are human-readable as well as machine-readable.
- XSDL enables the model builder to define constraints that extend the rules for determining whether an instance file conforms to a model.
- XSDL is a widely accepted language, and the XML file format of instance files is even more widely accepted.
- Tools for determining whether a model is syntactically correct and consistent and whether a given instance file conforms to a given model are available free or for a moderate price.

Tools for generating computer code that may be incorporated in an application from a model built in XSDL are available free or for a moderate price, thus lowering the cost of implementation.

QIF version 2.0 uses all the capabilities of XSDL just mentioned:

- over 1400 complex types are defined.
- derivation hierarchies are built up to five levels deep.
- over 100 specialized data types are defined.
- the QIFDocument model is built from 22 modules.
- QIF includes hundreds of key and keyref constraints.

In this section, the prefix xs: is used to indicate terms that are part of the XML schema definition language (XSDL). The xs: prefix is also used in the schema files. The term xs:type used here is not part of XSDL but means either xs:complexType or xs:simpleType.

7.1.1 Naming conventions

Certain naming conventions have been used in the development of the QIF XML schemas.

With few exceptions names are descriptive and formed by concatenation without abbreviation. One exception to the rule against abbreviation is that "identifier" is shortened everywhere it appears to "id" or "Id". All concatenated words in a name except possibly the first start with an upper case letter. The rules for the case of the first letter of the first word are:

- All names of XML items except xs:attribute names start with an upper case letter. This includes names for xs:type, xs:element, xs:key, and xs:keyref. Example xs:type name: ***ArcFeatureNominalType***
- xs:attribute names start with a lower case letter. Example xs:attribute name: id. There is one exception to this: an upper case N attribute is used for the number of items in a list or array.
- All xs:type names end in "Type".
- All xs:key names end in "Key".
- All xs:keyref names end in "Keyref".
- All names of instantiable feature xs:types end in "FeatureItemType", "FeatureDefinitionType", "FeatureNominalType", or "FeatureActualType".
- All names of instantiable characteristic xs:types end in "CharacteristicItemType", "CharacteristicDefinitionType", "CharacteristicNominalType", or "CharacteristicActualType".
- All names of enumerated xs:types end in "EnumType".
- Almost all names of xs:types that are parent xs:types not intended to be instantiated end in "BaseType".
- To a great extent, the names of xs:elements are formed from the name of the xs:type of the xs:element by removing the "Type" at the end. For example, the name of the xs:element whose xs:type is ***PlaneFeatureNominalType*** is **PlaneFeatureNominal**.

The name of an xs:element that is a reference to a QIF id almost always ends in "Id" and is always of ***QIFReferenceType***. If the value of an xs:element is a list of ids, the xs:element name almost always ends in "Ids".

7.1.2 Design rules

A number of design rules have been followed in building the QIF schema files.

- All xs:type definitions are declared globally, i.e., as direct children of an xs:schema. In other words, no xs:type definition is embedded inside another xs:type definition or inside an xs:element. This convention is commonly called using the venetian blind pattern.
- Although the names of QIF xs:elements and xs:types are very descriptive, the precise meaning almost always requires explanation. The XML schema definition language includes an xs:documentation node type that may be used to put documentation into a

schema. Documentation nodes must be preserved by XML tools. Comments may also be inserted in schema files but are not necessarily preserved by XML tools. Further details of documentation are given in section 7.2.

- All xs:types not intended to be instantiated are made abstract so as to be explicitly non-instantiable.
- When an xs:element is (or could be) declared to be of an abstract xs:type, there are three ways under the W3C rules in which the xs:element can be declared and instances of it put into instance files.
  - First, in the schema file, the xs:element may be declared to be of the abstract xs:type. In an instance file, instances of it may use the xs:element name for the abstract xs:type followed by an xsi:type declaration identifying one of the derived types. The xsi prefix is used for the standard XML instance namespace, http://www.w3.org/2001/XMLSchema-instance.
  - Second, in the schema file, rather than having a single xs:element declaration, an xs:choice of xs:elements of the various instantiable derived types may be used instead. In an instance file, one of the xs:elements in the xs:choice is used.
  - Third, in the schema file, the xs:element may be declared globally to be of the abstract xs:type and made to be the head of a substitution group. The instantiable derived xs:types are used as the xs:types for the other xs:elements in that substitution group. The xs:element for the abstract xs:type is then used via "ref" elsewhere in the schema file. In an instance file, the xs:element for a member of the substitution group headed by the abstract type is used.

  The first method can make writing key/keyref constraints difficult since (1) those constraints are expressed using xs:element names, (2) the constraints need to distinguish among xs:types, and (3) the same xs:element name is used with different xs:types. It also requires the instance file to be more verbose (because the xs:type declaration is needed). Hence, the first method is not used in QIF.

  The second and third methods have a different xs:type for each xs:element name, so they support writing key/keyref constraints. The third method is used overwhelmingly but not exclusively in QIF. The second method is used in several places.

- QIF 2.0 uses the namespace "http://www.qifstandards.org/xsd/qif2". The schema files in version 2.0 of QIF, use both no prefix and the prefix “t” for this namespace. The “t” is to satisfy the XSDL rule that xpaths require a prefix when a namespace is used.

7.1.3 Other naming and design items

XML instance files that conform to QIF must follow the rules for the conformance of instance files to XML schemas as defined by the W3C.

7.2 Annotation conventions

A great number of annotations (xs:annotation) containing documentation (xs:documentation) have been included in the QIF schema files in order to explain the full meaning of the QIF models. The meanings of the documentation nodes are essential parts of the model, not merely suggestions that may be taken or not. The documentation nodes have been prepared according the following conventions:

- Every schema file has a documentation node near the top that provides a general description of the file.

- Every xs:type has a documentation node.

- Every xs:element outside of a substitution group has a documentation node.

- The xs:element at the head of every substitution group has a documentation node.

- Every different sort of xs:key and xs:keyref has a documentation node. In the case of features and characteristics, the xs:keys and xs:keyrefs occur in large (over 25) batches that are very similar. For these batches, only the first member is documented.

- Only one documentation node is used under an annotation node, unless there is an exceptional reason to include more than one.

- The text of each documentation node consists of one or more complete sentences. The text can stand alone without the context of the XML schema code. The name of the schema file, attribute, type, element, key, or keyref is repeated in the sentence.

- All abstract types are annotated as abstract. QIF BaseTypes are always abstract.

- If an element has multiple pieces of information, then "defines" or "gives" is used to describe the element. If the element is one piece of information (even if it has substructure), then "is" is used.

- When an element has maxOccurs=”0”, that indicates the element is optional. In this case, the word “optional” immediately precedes the name of the element in the documentation.

- When an element has maxOccurs="unbounded", or maxOccurs=”N” (where N is an integer that is 2 or greater) there may be more than one such element in an instance file. In these cases, the usage is "Each XXX element defines" rather than "The XXX element defines". Note: The default for elements is exactly one.

- If an attribute is optional, which is the default for attributes, the word “optional” immediately precedes the attribute name in the documentation.

- If an attribute has use=”required”, that means the attribute is required. In this case, the word “required” immediately precedes the attribute name in the documentation.

- If a type is a list of specific values, the word "enumerated" or "enumerates" is always in the annotation, e.g., "The optional, enumerated TimeDescription element describes the time relative to the inspection, at which the environment data is measured."

- Individual enumeration values in an enumerated type do not have documentation nodes, but if the meaning of the values is not obvious, either they are explained in the documentation node of the type, or a reference is given to a document giving the meanings.

- Documentation nodes do not describe structural aspects of the information model.

- When text notes can assist people in reading and understanding the schema file, text describing the file structure is placed in XML comments.
- The sentence, “This element is in an optional choice.” describes a situation in which the element itself is not optional, but the element is effectively optional because the choice it is in is optional.

Annex A – Location of QIFDocument.xsd file

(informative)

The QIF Document information model is expressed in XML schema definition language in the file “QIFDocument.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.

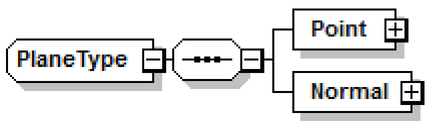


Figure B.1 – Notation for a type definition, *PlaneType*.

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***.

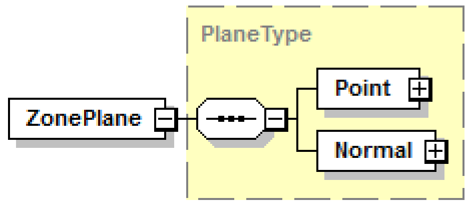


Figure B.2 – **Reuse of the type definition *PlaneType* to generate element ZonePlane.**

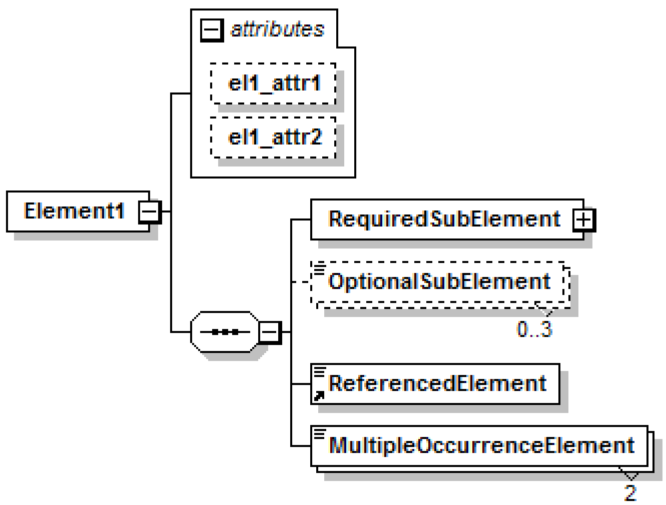
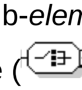


Figure B.3 – **Notation for *elements*, sub-*elements*, and *attributes*.**

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, e.g., "0..3", that indicates the number of occurrences as an inclusive range. The **OptionalSubElement** may occur zero,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 ()

indicates 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.

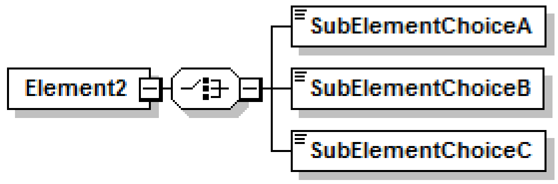


Figure B.4 – The *choice of* notation.

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 QIFDocument.xsd

(normative)

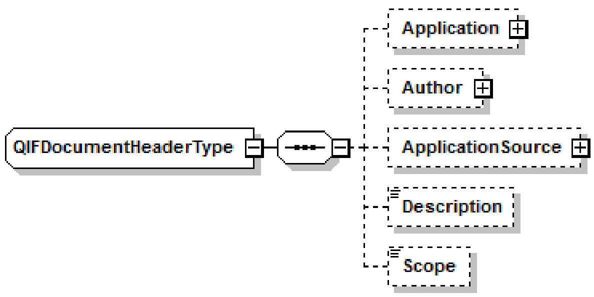
schema location:	**..\QIFApplications\QIFDocument.xsd**
attributeFormDefault:	**unqualified**
elementFormDefault:	**qualified**
targetNamespace:	**http://qifstandards.org/xsd/qif2**

Complex types

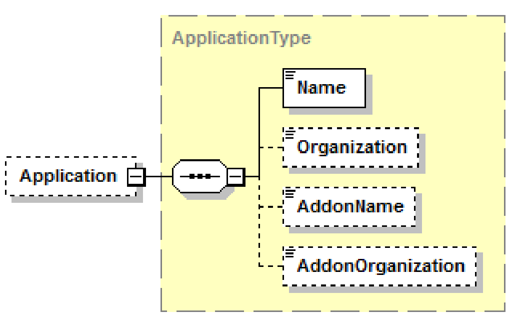
[QIFDocumentHeaderType]

[QIFDocumentType]

C.1 complexType QIFDocumentHeaderType

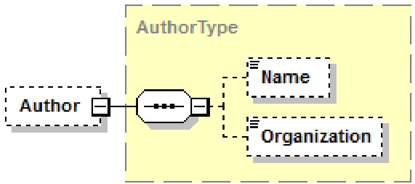
diagram	
children	[Application] [Author] [ApplicationSource] [Description] [Scope]
used by	element **Header**
annotation	documentation The HeaderType contains information about the generation of this QIF document.

C.2 element QIFDocumentHeaderType/Application

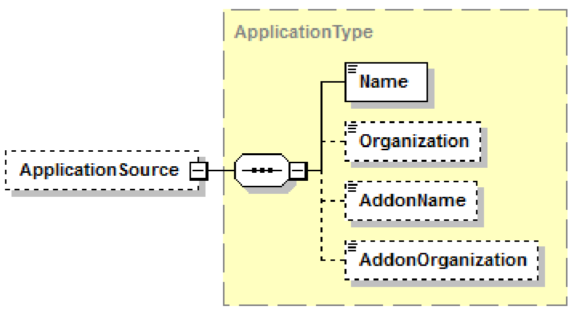
diagram	
type	**ApplicationType**
properties	minOcc 0 maxOcc 1 content complex
children	**Name Organization AddonName AddonOrganization**

annotation	documentation The optional Application element is the information about the software application wherein the QIF document was most recently edited.

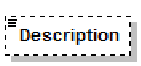
C.3 element QIFDocumentHeaderType/Author

diagram	
type	**AuthorType**
properties	minOcc 0 maxOcc 1 content complex
children	**Name Organization**
annotation	documentation The optional Author element is the author who generating this QIF document.

C.4 element QIFDocumentHeaderType/ApplicationSource

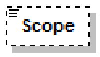
diagram	
type	**ApplicationType**
properties	minOcc 0 maxOcc 1 content complex
children	**Name Organization AddonName AddonOrganization**
annotation	documentation The optional ApplicationSource element is the information about the software application wherein the QIF document was created.

C.5 element QIFDocumentHeaderType/Description

diagram	
type	**xs:string**
properties	minOcc 0 maxOcc 1

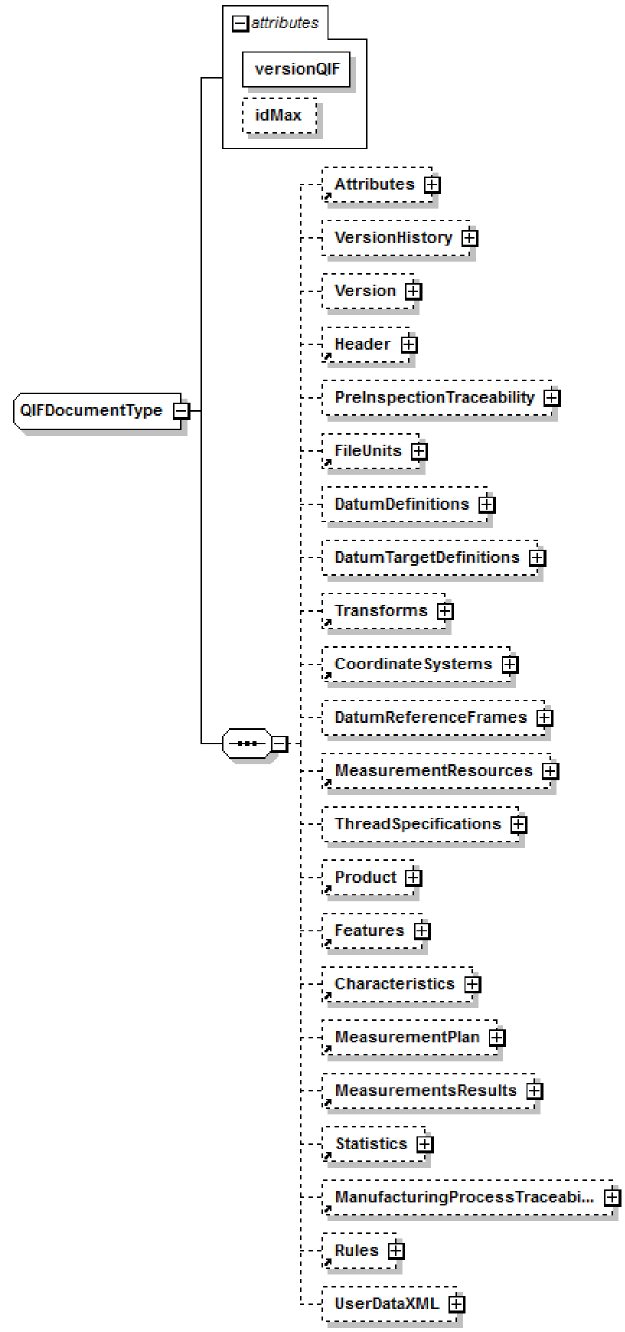
	content simple
annotation	documentation The optional Description element is a description of this QIF document.

C.6 element QIFDocumentHeaderType/Scope

diagram	
type	**xs:string**
properties	minOcc 0 maxOcc 1 content simple
annotation	documentation The optional Scope element defines the scope of this QIF document. Examples include: "Results","Plan", "Plan+Results", "MBD", "MBD+Plan+Results+Statistics", "Plan+External Product Definition". The scope of a QIF document can be determined from the optional elements that are present. This element allows for a concise, human-readable summary of the scope to appear near the beginning of a QIF document.

C.7 complexType QIFDocumentType

diagram



children	**Attributes VersionHistory Version Header PreInspectionTraceability FileUnits DatumDefinitions DatumTargetDefinitions Transforms CoordinateSystems DatumReferenceFrames MeasurementResources ThreadSpecifications Product Features Characteristics MeasurementPlan MeasurementsResults Statistics ManufacturingProcessTraceabilities Rules UserDataXML**
used by	element **QIFDocument**

attributes

Name	Type	Use	Default	Fixed	Annotation
versionQIF	**xs:NMTOKEN**	required		2.0.0	
idMax	**xs:unsignedInt**				documentation The idMax attribute specifies the largest ID used in the document.

annotation	documentation The QIFDocumentType defines a QIF document.

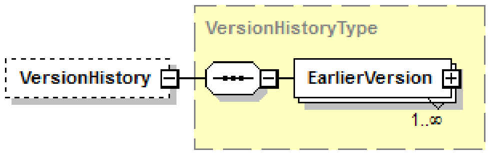
C.8 attribute QIFDocumentType/@versionQIF

type	**xs:NMTOKEN**
properties	use required fixed 2.0.0

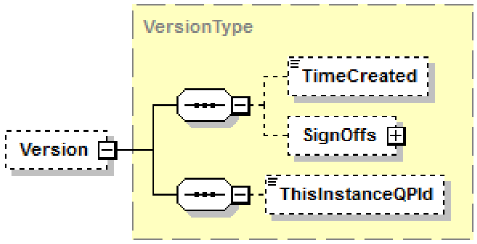
C.9 attribute QIFDocumentType/@idMax

type	**xs:unsignedInt**
annotation	documentation The idMax attribute specifies the largest ID used in the document.

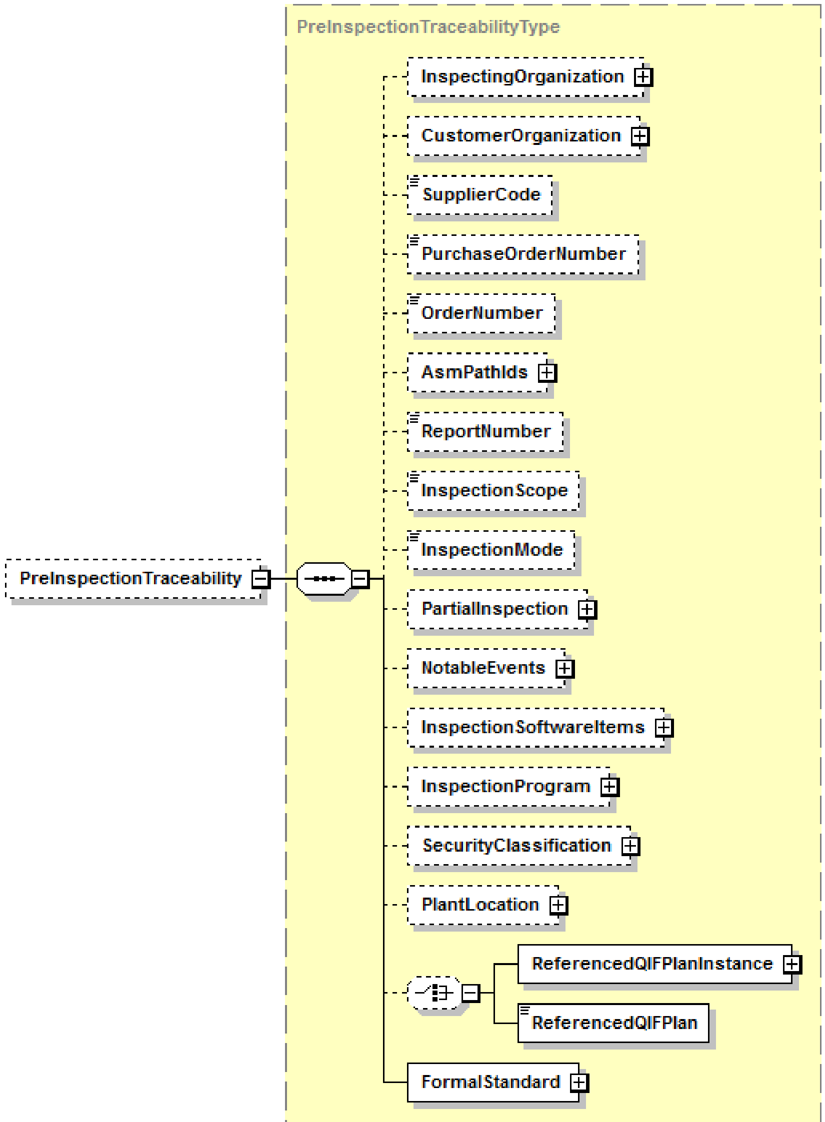
C.10 element QIFDocumentType/VersionHistory

diagram	
type	**VersionHistoryType**
properties	minOcc 0 maxOcc 1 content complex
children	**EarlierVersion**
annotation	documentation The optional VersionHistory element gives information about earlier versions of the QIF document.

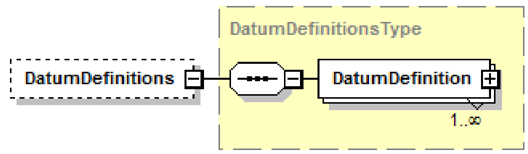
C.11 element QIFDocumentType/Version

diagram	
type	**VersionType**
properties	minOcc 0 maxOcc 1 content complex
children	**TimeCreated SignOffs ThisInstanceQPId**
annotation	documentation The optional Version element uniquely identifies the QIF document file.

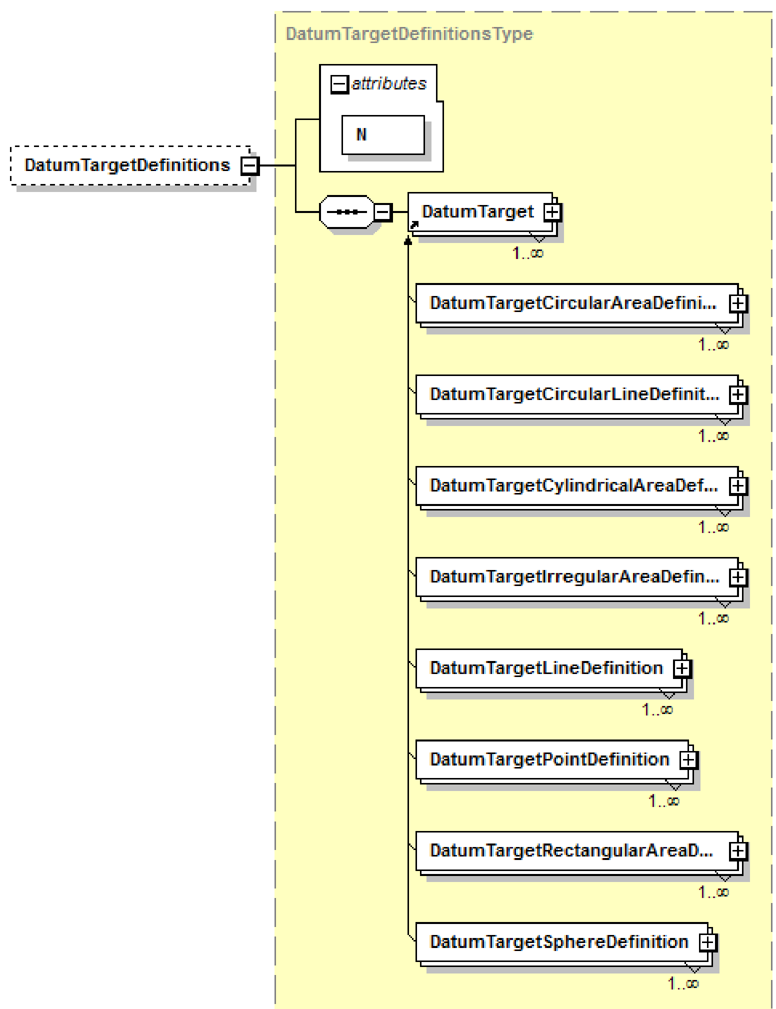
C.12 element QIFDocumentType/PreInspectionTraceability

diagram	
type	**PreInspectionTraceabilityType**
properties	minOcc 0 maxOcc 1 content complex
children	**InspectingOrganization CustomerOrganization SupplierCode PurchaseOrderNumber OrderNumber AsmPathIds ReportNumber InspectionScope InspectionMode PartialInspection NotableEvents InspectionSoftwareItems InspectionProgram SecurityClassification PlantLocation ReferencedQIFPlanInstance ReferencedQIFPlan FormalStandard**
annotation	documentation The optional PreInspectionTraceability element gives traceability information that applies to the entire QIF document.

C.13 element QIFDocumentType/DatumDefinitions

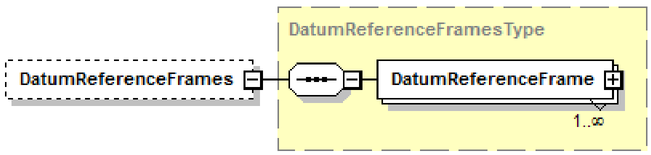
diagram	
type	**DatumDefinitionsType**
properties	minOcc 0 maxOcc 1 content complex
children	**DatumDefinition**
annotation	documentation The optional DatumDefinitions element gives information about the datum definitions to be used in the QIF document.

C.14 element QIFDocumentType/DatumTargetDefinitions

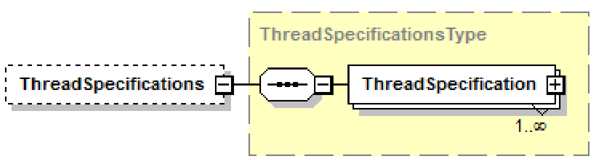
diagram	
type	**DatumTargetDefinitionsType**

properties	minOcc 0 maxOcc 1 content complex
children	**DatumTarget**
attributes	Name: N Type: **NaturalType** Use: required Default: Fixed: Annotation: documentation The required N attribute shows how many objects are present in the list.
annotation	documentation The optional DatumTargetDefinitions element gives information about the datum targets to be used in the QIF document.

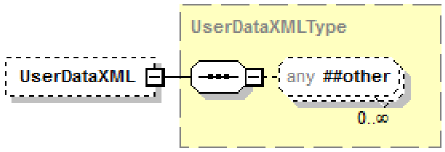
C.15 element QIFDocumentType/DatumReferenceFrames

diagram	
type	**DatumReferenceFramesType**
properties	minOcc 0 maxOcc 1 content complex
children	**DatumReferenceFrame**
annotation	documentation The optional DatumReferenceFrames element gives information about the datum reference frames to be used in the QIF document.

C.16 element QIFDocumentType/ThreadSpecifications

diagram	
type	**ThreadSpecificationsType**
properties	minOcc 0 maxOcc 1 content complex
children	**ThreadSpecification**
annotation	documentation The optional ThreadSpecifications element gives information about the thread specifications used in the QIF document.

C.17 element QIFDocumentType/UserDataXML

diagram	
type	**UserDataXMLType**
properties	minOcc 0 maxOcc 1 content complex
annotation	documentation The UserDataXMLType defines a user-defined XML structure from any namespace that is not the target namespace. The XML processor will validate elements if the corresponding schema will be presented. If the schema cannot be obtained, no errors will occur.

Annex D – Sample QIF instance files

(informative)

Two QIF instance files are included as inline text in this annex. They apply to the sample part drawing in Figure D.1.

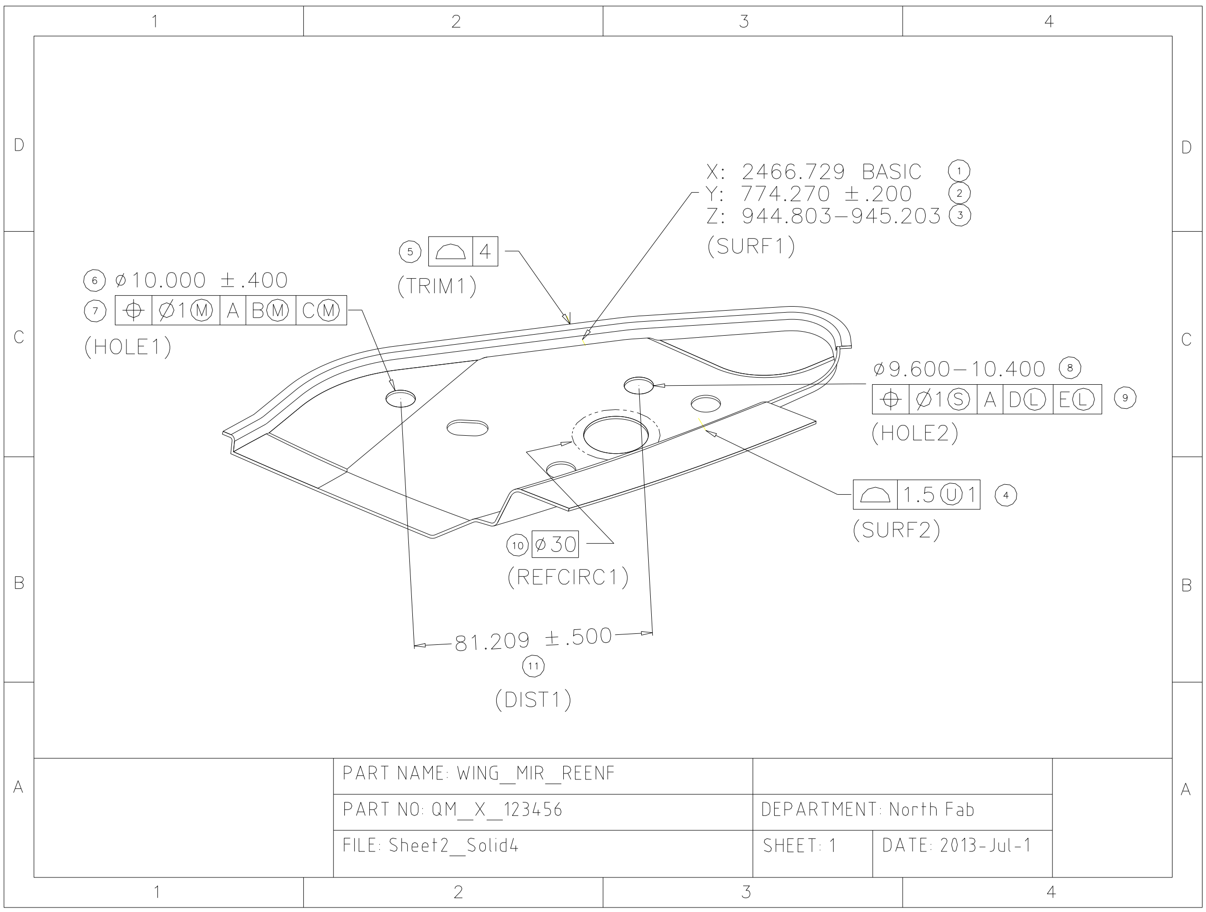


Figure D.1 – A sample part drawing

The features and ballooned characteristics called-out in Figure D.1 can be described fully and unambiguously in a QIF Plan instance file along with other traceability and measurement information.

File QIF_PLAN_SAMPLE.QIF

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<QIFDocument xmlns="http://qifstandards.org/xsd/qif2" idMax="68"
versionQIF="2.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://qifstandards.org/xsd/qif2
../QIFApplications/QIFDocument.xsd">

  <Version>
    <ThisInstanceQPId>4e44b72d-ae46-4ca9-b248-3075dde8e05a</ThisInstanceQPId>
```

```
</Version>

<Header>
  <Application>
    <Name>Solidworks 2014</Name>
    <AddonName>CheckMate 14</AddonName>
    <AddonOrganization>Origin International Inc.</AddonOrganization>
  </Application>
</Header>

<PreInspectionTraceability>
  <InspectingOrganization>
    <Name>Origin International</Name>
    <Address>
      <StreetNumber>72</StreetNumber>
      <Street>Baynards Lane</Street>
      <Town>Richmond Hill</Town>
      <Region>Ontario</Region>
      <PostalCode>L4C 9B8</PostalCode>
      <FacsimileNumber>416 410 8313</FacsimileNumber>
      <TelephoneNumber>1-800-269-2509</TelephoneNumber>
      <ElectronicMailAddress>support@originintl.com</ElectronicMailAddress>
    </Address>
  </InspectingOrganization>
  <SupplierCode>North_Fab</SupplierCode>
  <PurchaseOrderNumber>PO123456</PurchaseOrderNumber>
  <AsmPathIds N="1">
    <Id>3</Id>
  </AsmPathIds>
  <ReportNumber>QIF 1</ReportNumber>
  <InspectionScope>DETAIL</InspectionScope>
  <InspectionMode>FAI_Full</InspectionMode>
  <FormalStandard>
    <FormalStandardEnum>ASME-Y14.5-1994</FormalStandardEnum>
  </FormalStandard>
</PreInspectionTraceability>

<FileUnits>
  <PrimaryUnits>
    <AngularUnit>
      <SIUnitName>radian</SIUnitName>
      <UnitName>degree</UnitName>
      <UnitConversion>
        <Factor>0.017453292519943</Factor>
      </UnitConversion>
    </AngularUnit>
    <LinearUnit>
      <SIUnitName>meter</SIUnitName>
      <UnitName>mm</UnitName>
```



```
      <UnitConversion>
        <Factor>0.001</Factor>
      </UnitConversion>
    </LinearUnit>
  </PrimaryUnits>
</FileUnits>

<DatumDefinitions>
  <DatumDefinition id="41">
    <DatumLabel>A</DatumLabel>
  </DatumDefinition>
  <DatumDefinition id="42">
    <DatumLabel>B</DatumLabel>
  </DatumDefinition>
  <DatumDefinition id="43">
    <DatumLabel>C</DatumLabel>
  </DatumDefinition>
  <DatumDefinition id="56">
    <DatumLabel>D</DatumLabel>
  </DatumDefinition>
  <DatumDefinition id="57">
    <DatumLabel>E</DatumLabel>
  </DatumDefinition>
</DatumDefinitions>

<DatumReferenceFrames>
  <DatumReferenceFrame id="11"/>
  <DatumReferenceFrame id="40">
    <Datum>
      <SimpleDatum>
        <DatumDefinitionId>41</DatumDefinitionId>
        <MaterialModifier>NONE</MaterialModifier>
        <ReferencedComponent>ACTUAL</ReferencedComponent>
      </SimpleDatum>
      <Precedence>
        <PrecedenceEnum>PRIMARY</PrecedenceEnum>
      </Precedence>
    </Datum>
    <Datum>
      <SimpleDatum>
        <DatumDefinitionId>42</DatumDefinitionId>
        <MaterialModifier>MAXIMUM</MaterialModifier>
        <ReferencedComponent>ACTUAL</ReferencedComponent>
      </SimpleDatum>
      <Precedence>
        <PrecedenceEnum>SECONDARY</PrecedenceEnum>
      </Precedence>
    </Datum>
    <Datum>
```

```
      <SimpleDatum>
        <DatumDefinitionId>43</DatumDefinitionId>
        <MaterialModifier>MAXIMUM</MaterialModifier>
        <ReferencedComponent>ACTUAL</ReferencedComponent>
      </SimpleDatum>
      <Precedence>
        <PrecedenceEnum>TERTIARY</PrecedenceEnum>
      </Precedence>
    </Datum>
  </DatumReferenceFrame>
  <DatumReferenceFrame id="55">
    <Datum>
      <SimpleDatum>
        <DatumDefinitionId>41</DatumDefinitionId>
        <MaterialModifier>NONE</MaterialModifier>
        <ReferencedComponent>ACTUAL</ReferencedComponent>
      </SimpleDatum>
      <Precedence>
        <PrecedenceEnum>PRIMARY</PrecedenceEnum>
      </Precedence>
    </Datum>
    <Datum>
      <SimpleDatum>
        <DatumDefinitionId>56</DatumDefinitionId>
        <MaterialModifier>LEAST</MaterialModifier>
        <ReferencedComponent>ACTUAL</ReferencedComponent>
      </SimpleDatum>
      <Precedence>
        <PrecedenceEnum>SECONDARY</PrecedenceEnum>
      </Precedence>
    </Datum>
    <Datum>
      <SimpleDatum>
        <DatumDefinitionId>57</DatumDefinitionId>
        <MaterialModifier>LEAST</MaterialModifier>
        <ReferencedComponent>ACTUAL</ReferencedComponent>
      </SimpleDatum>
      <Precedence>
        <PrecedenceEnum>TERTIARY</PrecedenceEnum>
      </Precedence>
    </Datum>
  </DatumReferenceFrame>
</DatumReferenceFrames>

<MeasurementResources>
  <MeasurementDevices>
    <MeasurementDevice id="14">
      <Name>CMM</Name>
    </MeasurementDevice>
```

```
    <MeasurementDevice id="46">
      <Name>GAGE PINS</Name>
    </MeasurementDevice>
    <MeasurementDevice id="53">
      <Name>CALIPERS</Name>
    </MeasurementDevice>
  </MeasurementDevices>
</MeasurementResources>

<Product>
  <PartSet N="1">
    <Part id="1">
      <Name>WING_MIR_REENF</Name>
      <QPId>2bbeb82a-96bf-4f1e-a327-4ba3500490e1</QPId>
      <ModelNumber>QM_X_123456</ModelNumber>
      <Description>Wing Mirror Re-enforcement</Description>
      <Version>1.02</Version>
      <DefinitionExternal id="6">
        <PrintedDrawing id="5">
          <Name>sheet2_solid4</Name>
          <Version>1.0.0</Version>
          <Description>Sample QIF elements</Description>
          <DrawingNumber>#1</DrawingNumber>
          <AdditionalChanges>none</AdditionalChanges>
          <Location>Cabinet 17, Drawer 3</Location>
        </PrintedDrawing>
      </DefinitionExternal>
    </Part>
  </PartSet>
  <ComponentSet N="1">
    <Component id="2">
      <QPId>af4d1612-8918-4a69-a716-709beee7a953</QPId>
      <Part>
        <Id>1</Id>
      </Part>
    </Component>
  </ComponentSet>
  <AsmPaths N="1">
    <AsmPath id="3">
      <ComponentIds N="1">
        <Id>2</Id>
      </ComponentIds>
    </AsmPath>
  </AsmPaths>
</Product>

<Features>
  <FeatureDefinitions>
    <EdgePointFeatureDefinition id="7">
```



```
        <InternalExternal>EXTERNAL</InternalExternal>
      </EdgePointFeatureDefinition>
      <PointFeatureDefinition id="15"/>
      <PointFeatureDefinition id="27"/>
      <CircleFeatureDefinition id="33">
        <InternalExternal>INTERNAL</InternalExternal>
        <Diameter>10</Diameter>
      </CircleFeatureDefinition>
      <CircleFeatureDefinition id="47">
        <InternalExternal>INTERNAL</InternalExternal>
        <Diameter>10</Diameter>
      </CircleFeatureDefinition>
      <CircleFeatureDefinition id="60">
        <InternalExternal>NOT_APPLICABLE</InternalExternal>
        <Diameter>30</Diameter>
      </CircleFeatureDefinition>
    </FeatureDefinitions>
    <FeatureNominals>
      <EdgePointFeatureNominal id="8">
        <FeatureDefinitionId>7</FeatureDefinitionId>
        <Location>2460.7099609375 770.604614257813
944.993591308594</Location>
        <Normal>-0.735465884156759 -0.307902932144912
0.603560864882807</Normal>
        <AdjacentNormal>0.0411800179070265 -0.909449806293278 -
0.413769568671133</AdjacentNormal>
      </EdgePointFeatureNominal>
      <PointFeatureNominal id="16">
        <FeatureDefinitionId>15</FeatureDefinitionId>
        <Location>2466.72924804688 774.269897460938
945.002746582031</Location>
        <Normal>0.731520704665031 0.146124942551 -0.665976696146977</Normal>
      </PointFeatureNominal>
      <PointFeatureNominal id="28">
        <FeatureDefinitionId>27</FeatureDefinitionId>
        <Location>2536.49487304688 782.806213378906
920.002197265625</Location>
        <Normal>-0.7609260410224 -0.648538030188332 -
0.0197479997320212</Normal>
      </PointFeatureNominal>
      <CircleFeatureNominal id="34">
        <FeatureDefinitionId>33</FeatureDefinitionId>
        <Location>0.0558150216639657 -0.907624351305543 -
0.416056150385791</Location>
        <Normal>2433.974609375 800.617431640625 890.049621582031</Normal>
      </CircleFeatureNominal>
      <CircleFeatureNominal id="48">
        <FeatureDefinitionId>47</FeatureDefinitionId>
```

```
        <Location>0.0418550117439733 -0.909169210310978 -
0.414318120547998</Location>
        <Normal>2496.25146484375 780.842590332031 938.272277832031</Normal>
      </CircleFeatureNominal>
      <CircleFeatureNominal id="61">
        <FeatureDefinitionId>60</FeatureDefinitionId>
        <Location>0.0418467877611875 -0.909189553580246 -
0.414274307693132</Location>
        <Normal>2506.63672251543 792.999124488284 912.644676969086</Normal>
      </CircleFeatureNominal>
    </FeatureNominals>
    <FeatureItems>
      <EdgePointFeatureItem id="9">
        <Attributes N="1">
          <AttributeStr name="Comment" value="An edge point with uniform
profile characteristic"/>
        </Attributes>
        <FeatureNominalId>8</FeatureNominalId>
        <FeatureName>TRIM1</FeatureName>
        <QPId>aeba4615-65db-4f7d-a2e8-a7928c80117f</QPId>
        <DeterminationMode>
          <Checked>
            <CheckDetails>
              <Measured/>
            </CheckDetails>
          </Checked>
        </DeterminationMode>
      </EdgePointFeatureItem>
      <PointFeatureItem id="17">
        <Attributes N="1">
          <AttributeStr name="Comment" value="A point on a surface with
coordinate characteristics"/>
        </Attributes>
        <FeatureNominalId>16</FeatureNominalId>
        <FeatureName>SURF1</FeatureName>
        <QPId>263c98a7-9786-4408-b62f-dda79a07daad</QPId>
        <DeterminationMode>
          <Checked>
            <CheckDetails>
              <Measured/>
            </CheckDetails>
          </Checked>
        </DeterminationMode>
      </PointFeatureItem>
      <PointFeatureItem id="29">
        <Attributes N="1">
          <AttributeStr name="Comment" value="A point on a surface with non-
uniform profile characteristic"/>
        </Attributes>
```

```
        <FeatureNominalId>28</FeatureNominalId>
        <FeatureName>SURF2</FeatureName>
        <QPId>63799169-1d43-41ee-b1ad-451f9d8a8f4e</QPId>
        <DeterminationMode>
          <Checked>
            <CheckDetails>
              <Measured/>
            </CheckDetails>
          </Checked>
        </DeterminationMode>
      </PointFeatureItem>
      <CircleFeatureItem id="35">
        <Attributes N="1">
          <AttributeStr name="Comment" value="A hole with position and gage-
pin diameter"/>
        </Attributes>
        <FeatureNominalId>34</FeatureNominalId>
        <FeatureName>HOLE1</FeatureName>
        <QPId>f29bb618-e50d-4dbb-b4b4-83b28755c741</QPId>
        <DeterminationMode>
          <Checked>
            <CheckDetails>
              <Measured/>
            </CheckDetails>
          </Checked>
        </DeterminationMode>
      </CircleFeatureItem>
      <CircleFeatureItem id="49">
        <Attributes N="1">
          <AttributeStr name="Comment" value="A hole with position and
caliper diameter"/>
        </Attributes>
        <FeatureNominalId>48</FeatureNominalId>
        <FeatureName>HOLE2</FeatureName>
        <QPId>5bb191b0-7a18-4393-8f24-1353ac8365c7</QPId>
        <DeterminationMode>
          <Checked>
            <CheckDetails>
              <Measured/>
            </CheckDetails>
          </Checked>
        </DeterminationMode>
      </CircleFeatureItem>
      <CircleFeatureItem id="62">
        <Attributes N="1">
          <AttributeStr name="Comment" value="A non-measureable reference
circle"/>
        </Attributes>
        <FeatureNominalId>61</FeatureNominalId>
```

```
      <FeatureName>REFCIRC1</FeatureName>
      <QPId>6443f863-c559-42f2-b62a-48c0172d3395</QPId>
      <DeterminationMode>
        <Set/>
      </DeterminationMode>
    </CircleFeatureItem>
  </FeatureItems>
</Features>

<Characteristics>
  <FormalStandard>
    <FormalStandardEnum>ASME-Y14.5-1994</FormalStandardEnum>
  </FormalStandard>
  <CharacteristicDefinitions>
    <PointProfileCharacteristicDefinition id="10">
      <ToleranceValue>4</ToleranceValue>
      <DatumReferenceFrameId>11</DatumReferenceFrameId>
    </PointProfileCharacteristicDefinition>
    <LinearCoordinateCharacteristicDefinition id="18">
      <NonTolerance>MEASURED</NonTolerance>
    </LinearCoordinateCharacteristicDefinition>
    <LinearCoordinateCharacteristicDefinition id="21">
      <Tolerance>
        <MaxValue>0.2</MaxValue>
        <MinValue>-0.2</MinValue>
        <DefinedAsLimit>false</DefinedAsLimit>
      </Tolerance>
    </LinearCoordinateCharacteristicDefinition>
    <LinearCoordinateCharacteristicDefinition id="24">
      <Tolerance>
        <MaxValue>945.20274658203095</MaxValue>
        <MinValue>944.80274658203086</MinValue>
        <DefinedAsLimit>true</DefinedAsLimit>
      </Tolerance>
    </LinearCoordinateCharacteristicDefinition>
    <PointProfileCharacteristicDefinition id="30">
      <ToleranceValue>1.5</ToleranceValue>
      <OuterDisposition>1</OuterDisposition>
      <DatumReferenceFrameId>11</DatumReferenceFrameId>
    </PointProfileCharacteristicDefinition>
    <DiameterCharacteristicDefinition id="36">
      <Tolerance>
        <MaxValue>0.4</MaxValue>
        <MinValue>-0.4</MinValue>
        <DefinedAsLimit>false</DefinedAsLimit>
      </Tolerance>
    </DiameterCharacteristicDefinition>
    <PositionCharacteristicDefinition id="39">
      <ToleranceValue>1</ToleranceValue>
```

```
      <DatumReferenceFrameId>40</DatumReferenceFrameId>
      <MaterialCondition>MAXIMUM</MaterialCondition>
      <ZoneShape>
        <DiametricalZone/>
      </ZoneShape>
    </PositionCharacteristicDefinition>
    <DiameterCharacteristicDefinition id="50">
      <Tolerance>
        <MaxValue>10.4</MaxValue>
        <MinValue>9.6</MinValue>
        <DefinedAsLimit>true</DefinedAsLimit>
      </Tolerance>
    </DiameterCharacteristicDefinition>
    <PositionCharacteristicDefinition id="54">
      <ToleranceValue>1</ToleranceValue>
      <DatumReferenceFrameId>55</DatumReferenceFrameId>
      <MaterialCondition>REGARDLESS</MaterialCondition>
      <ZoneShape>
        <DiametricalZone/>
      </ZoneShape>
    </PositionCharacteristicDefinition>
    <DiameterCharacteristicDefinition id="63">
      <NonTolerance>SET</NonTolerance>
    </DiameterCharacteristicDefinition>
    <DistanceBetweenCharacteristicDefinition id="66">
      <Tolerance>
        <MaxValue>0.5</MaxValue>
        <MinValue>-0.5</MinValue>
        <DefinedAsLimit>false</DefinedAsLimit>
      </Tolerance>
    </DistanceBetweenCharacteristicDefinition>
  </CharacteristicDefinitions>
  <CharacteristicNominals>
    <PointProfileCharacteristicNominal id="12">
      <CharacteristicDefinitionId>10</CharacteristicDefinitionId>
    </PointProfileCharacteristicNominal>
    <LinearCoordinateCharacteristicNominal id="19">
      <CharacteristicDefinitionId>18</CharacteristicDefinitionId>
      <TargetValue>2466.729248046875</TargetValue>
      <Direction>XAXIS</Direction>
    </LinearCoordinateCharacteristicNominal>
    <LinearCoordinateCharacteristicNominal id="22">
      <CharacteristicDefinitionId>21</CharacteristicDefinitionId>
      <TargetValue>774.26989746093795</TargetValue>
      <Direction>YAXIS</Direction>
    </LinearCoordinateCharacteristicNominal>
    <LinearCoordinateCharacteristicNominal id="25">
      <CharacteristicDefinitionId>24</CharacteristicDefinitionId>
      <Direction>ZAXIS</Direction>
```

```
      </LinearCoordinateCharacteristicNominal>
      <PointProfileCharacteristicNominal id="31">
        <CharacteristicDefinitionId>30</CharacteristicDefinitionId>
      </PointProfileCharacteristicNominal>
      <DiameterCharacteristicNominal id="37">
        <CharacteristicDefinitionId>36</CharacteristicDefinitionId>
        <TargetValue>10</TargetValue>
      </DiameterCharacteristicNominal>
      <PositionCharacteristicNominal id="44">
        <CharacteristicDefinitionId>39</CharacteristicDefinitionId>
      </PositionCharacteristicNominal>
      <DiameterCharacteristicNominal id="51">
        <CharacteristicDefinitionId>50</CharacteristicDefinitionId>
      </DiameterCharacteristicNominal>
      <PositionCharacteristicNominal id="58">
        <CharacteristicDefinitionId>54</CharacteristicDefinitionId>
      </PositionCharacteristicNominal>
      <DiameterCharacteristicNominal id="64">
        <CharacteristicDefinitionId>63</CharacteristicDefinitionId>
        <TargetValue>30</TargetValue>
      </DiameterCharacteristicNominal>
      <DistanceBetweenCharacteristicNominal id="67">
        <CharacteristicDefinitionId>66</CharacteristicDefinitionId>
        <TargetValue>81.208839738425993</TargetValue>
        <AnalysisMode>THREEDIMENSIONAL</AnalysisMode>
      </DistanceBetweenCharacteristicNominal>
    </CharacteristicNominals>
    <CharacteristicItems>
      <PointProfileCharacteristicItem id="13">
        <Attributes N="1">
          <AttributeStr name="Comment" value="A uniform profile zone ±2.0
with empty DRF"/>
        </Attributes>
        <Name>5</Name>
        <QPId>6f2a6f83-3ff7-4327-b00c-98208361301a</QPId>
        <KeyCharacteristic>
          <Designator>5</Designator>
          <Criticality>MINOR</Criticality>
        </KeyCharacteristic>
        <FeatureItemIds N="1">
          <Id>9</Id>
        </FeatureItemIds>
        <MeasurementDeviceIds N="1">
          <Id>14</Id>
        </MeasurementDeviceIds>
        <CharacteristicNominalId>12</CharacteristicNominalId>
        <LocationOnDrawing>
          <DrawingId>5</DrawingId>
          <SheetNumber>SHEET1</SheetNumber>
```

```
          <DrawingZone>C2</DrawingZone>
        </LocationOnDrawing>
      </PointProfileCharacteristicItem>
      <LinearCoordinateCharacteristicItem id="20">
        <Attributes N="1">
          <AttributeStr name="Comment" value="A basic coordinate dimension"/>
        </Attributes>
        <Name>1</Name>
        <QPId>fdb07604-ac7e-4f7c-acc8-0860de517fd4</QPId>
        <KeyCharacteristic>
          <Designator>1</Designator>
          <Criticality>REF</Criticality>
        </KeyCharacteristic>
        <FeatureItemIds N="1">
          <Id>17</Id>
        </FeatureItemIds>
        <MeasurementDeviceIds N="1">
          <Id>14</Id>
        </MeasurementDeviceIds>
        <CharacteristicNominalId>19</CharacteristicNominalId>
        <LocationOnDrawing>
          <DrawingId>5</DrawingId>
          <SheetNumber>SHEET1</SheetNumber>
          <DrawingZone>D3</DrawingZone>
        </LocationOnDrawing>
      </LinearCoordinateCharacteristicItem>
      <LinearCoordinateCharacteristicItem id="23">
        <Attributes N="1">
          <AttributeStr name="Comment" value="A bi-directional coordinate
tolerance"/>
        </Attributes>
        <Name>2</Name>
        <QPId>445298bf-91a2-4ec6-8fa4-3b2c029eeb6e</QPId>
        <KeyCharacteristic>
          <Designator>2</Designator>
          <Criticality>MINOR</Criticality>
        </KeyCharacteristic>
        <FeatureItemIds N="1">
          <Id>17</Id>
        </FeatureItemIds>
        <MeasurementDeviceIds N="1">
          <Id>14</Id>
        </MeasurementDeviceIds>
        <CharacteristicNominalId>22</CharacteristicNominalId>
        <LocationOnDrawing>
          <DrawingId>5</DrawingId>
          <SheetNumber>SHEET1</SheetNumber>
          <DrawingZone>D3</DrawingZone>
        </LocationOnDrawing>
```



```
      </LinearCoordinateCharacteristicItem>
      <LinearCoordinateCharacteristicItem id="26">
        <Attributes N="1">
          <AttributeStr name="Comment" value="A limit coordinate tolerance"/>
        </Attributes>
        <Name>3</Name>
        <QPId>187ce9c9-7125-47b0-869b-d37a9722a35d</QPId>
        <KeyCharacteristic>
          <Designator>3</Designator>
          <Criticality>MAJOR</Criticality>
        </KeyCharacteristic>
        <FeatureItemIds N="1">
          <Id>17</Id>
        </FeatureItemIds>
        <MeasurementDeviceIds N="1">
          <Id>14</Id>
        </MeasurementDeviceIds>
        <CharacteristicNominalId>25</CharacteristicNominalId>
        <LocationOnDrawing>
          <DrawingId>5</DrawingId>
          <SheetNumber>SHEET1</SheetNumber>
          <DrawingZone>D3</DrawingZone>
        </LocationOnDrawing>
      </LinearCoordinateCharacteristicItem>
      <PointProfileCharacteristicItem id="32">
        <Attributes N="1">
          <AttributeStr name="Comment" value="An offset profile zone +1.0/-
0.5 to ABC"/>
        </Attributes>
        <Name>4</Name>
        <QPId>7ad19cce-458a-4d0a-9436-6a2d2683b018</QPId>
        <KeyCharacteristic>
          <Designator>4</Designator>
          <Criticality>CRITICAL</Criticality>
        </KeyCharacteristic>
        <FeatureItemIds N="1">
          <Id>29</Id>
        </FeatureItemIds>
        <MeasurementDeviceIds N="1">
          <Id>14</Id>
        </MeasurementDeviceIds>
        <CharacteristicNominalId>31</CharacteristicNominalId>
        <LocationOnDrawing>
          <DrawingId>5</DrawingId>
          <SheetNumber>SHEET1</SheetNumber>
          <DrawingZone>B3</DrawingZone>
        </LocationOnDrawing>
      </PointProfileCharacteristicItem>
      <DiameterCharacteristicItem id="38">
```

```
        <Attributes N="1">
          <AttributeStr name="Comment" value="A bi-directional diameter
tolerance"/>
        </Attributes>
        <Name>6</Name>
        <QPId>4e609ab8-7d1c-4317-9b16-5fea194f9d7e</QPId>
        <KeyCharacteristic>
          <Designator>6</Designator>
          <Criticality>MINOR</Criticality>
        </KeyCharacteristic>
        <FeatureItemIds N="1">
          <Id>35</Id>
        </FeatureItemIds>
        <MeasurementDeviceIds N="1">
          <Id>14</Id>
        </MeasurementDeviceIds>
        <CharacteristicNominalId>37</CharacteristicNominalId>
        <LocationOnDrawing>
          <DrawingId>5</DrawingId>
          <SheetNumber>SHEET1</SheetNumber>
          <DrawingZone>C1</DrawingZone>
        </LocationOnDrawing>
      </DiameterCharacteristicItem>
      <PositionCharacteristicItem id="45">
        <Attributes N="1">
          <AttributeStr name="Comment" value="MMC diametrical position to
AB(m)C(m)"/>
        </Attributes>
        <Name>7</Name>
        <QPId>4438e85a-441a-420f-93d8-c6ed52fae87e</QPId>
        <KeyCharacteristic>
          <Designator>7</Designator>
          <Criticality>CRITICAL</Criticality>
        </KeyCharacteristic>
        <FeatureItemIds N="1">
          <Id>35</Id>
        </FeatureItemIds>
        <MeasurementDeviceIds N="1">
          <Id>46</Id>
        </MeasurementDeviceIds>
        <CharacteristicNominalId>44</CharacteristicNominalId>
        <LocationOnDrawing>
          <DrawingId>5</DrawingId>
          <SheetNumber>SHEET1</SheetNumber>
          <DrawingZone>C1</DrawingZone>
        </LocationOnDrawing>
      </PositionCharacteristicItem>
      <DiameterCharacteristicItem id="52">
        <Attributes N="1">
```

```
          <AttributeStr name="Comment" value="Limit diameter tolerance with
caliper "/>
        </Attributes>
        <Name>8</Name>
        <QPId>a5a0d3a2-e6c4-4be9-93fe-18ba95d8372c</QPId>
        <KeyCharacteristic>
          <Designator>8</Designator>
        </KeyCharacteristic>
        <FeatureItemIds N="1">
          <Id>49</Id>
        </FeatureItemIds>
        <MeasurementDeviceIds N="1">
          <Id>53</Id>
        </MeasurementDeviceIds>
        <CharacteristicNominalId>51</CharacteristicNominalId>
        <LocationOnDrawing>
          <DrawingId>5</DrawingId>
          <SheetNumber>SHEET1</SheetNumber>
          <DrawingZone>C3</DrawingZone>
        </LocationOnDrawing>
      </DiameterCharacteristicItem>
      <PositionCharacteristicItem id="59">
        <Attributes N="1">
          <AttributeStr name="Comment" value="RFS position to AD(l)E(l)"/>
        </Attributes>
        <Name>9</Name>
        <QPId>c79e4f25-f235-4886-88a3-b8a2eeaba096</QPId>
        <KeyCharacteristic>
          <Designator>9</Designator>
          <Criticality>MINOR</Criticality>
        </KeyCharacteristic>
        <FeatureItemIds N="1">
          <Id>49</Id>
        </FeatureItemIds>
        <MeasurementDeviceIds N="1">
          <Id>14</Id>
        </MeasurementDeviceIds>
        <CharacteristicNominalId>58</CharacteristicNominalId>
        <LocationOnDrawing>
          <DrawingId>5</DrawingId>
          <SheetNumber>SHEET1</SheetNumber>
          <DrawingZone>C3</DrawingZone>
        </LocationOnDrawing>
      </PositionCharacteristicItem>
      <DiameterCharacteristicItem id="65">
        <Name>10</Name>
        <QPId>ee9bb7c3-31fb-40a4-841d-4f1b9fc885bd</QPId>
        <KeyCharacteristic>
          <Designator>10</Designator>
```

```
        </KeyCharacteristic>
        <FeatureItemIds N="1">
          <Id>62</Id>
        </FeatureItemIds>
        <CharacteristicNominalId>64</CharacteristicNominalId>
      </DiameterCharacteristicItem>
      <DistanceBetweenCharacteristicItem id="68">
        <Attributes N="1">
          <AttributeStr name="Comment" value="Distance between HOLE1 and
HOLE2"/>
        </Attributes>
        <Name>DIST1</Name>
        <QPId>3e6797f5-b443-4b15-9108-055190223837</QPId>
        <KeyCharacteristic>
          <Designator>11</Designator>
        </KeyCharacteristic>
        <FeatureItemIds N="2">
          <Id>49</Id>
          <Id>35</Id>
        </FeatureItemIds>
        <MeasurementDeviceIds N="1">
          <Id>14</Id>
        </MeasurementDeviceIds>
        <CharacteristicNominalId>67</CharacteristicNominalId>
        <LocationOnDrawing>
          <DrawingId>5</DrawingId>
          <SheetNumber>SHEET1</SheetNumber>
          <DrawingZone>B2</DrawingZone>
        </LocationOnDrawing>
      </DistanceBetweenCharacteristicItem>
    </CharacteristicItems>
  </Characteristics>

  <MeasurementPlan>
    <Version>
      <ThisInstanceQPId>9fcebcb6-99da-44f9-b997-
2ba76fd438dd</ThisInstanceQPId>
    </Version>
    <UnorderedPlanRoot>
      <Steps>
        <MeasureEvaluateAll/>
      </Steps>
    </UnorderedPlanRoot>
  </MeasurementPlan>

</QIFDocument>
```


*** end Plans sample file ***

The execution of the sample plan with the specified measurement equipment might result in a measurement report file like:

SURF1 POINT (SURFACE)

		ACTUAL	NOMINAL	UPR-TOL	LWR-TOL	BONUS	DEV	ERROR
1	X	2466.900	2466.729	BASIC			0.171	
2	Y	774.310	774.270	0.200	-0.200		0.040	----|*---
3	Z	944.840	945.003	0.200	-0.200		-0.163	*---|----

SURF2 POINT (SURFACE)

		ACTUAL	NOMINAL	UPR-TOL	LWR-TOL	BONUS	DEV	ERROR	**
	X	2537.170	2536.495				0.675		
	Y	783.380	782.806				0.574		
	Z	920.020	920.002				0.018		
4	*V	-0.886	0.000	1.000	-0.500		-0.886	-0.386	

TRIM1 EDGE POINT (TRIM)

		ACTUAL	NOMINAL	UPR-TOL	LWR-TOL	BONUS	DEV	ERROR
	X	2460.720	2460.710				0.010	
	Y	770.620	770.605				0.015	
	Z	944.980	944.994				-0.014	
5	V	-0.020	0.000	2.000	-2.000		-0.020	---*|----

HOLE1 CIRCLE

		ACTUAL	NOMINAL	UPR-TOL	LWR-TOL	BONUS	DEV	ERROR	**
	X	2434.010	2433.975				0.035		
	Z	889.980	890.050				-0.070		
6	*D	9.499	10.000	0.400	-0.400	0.000	-0.501	-0.101	

```
7         P     0.897     0.000    1.000             1.000    0.897      |---*


HOLE2     CIRCLE
              ACTUAL    NOMINAL   UPR-TOL   LWR-TOL    BONUS      DEV     ERROR   **
          X   2496.390  2496.251                        0.139
          Z    938.090   938.272                       -0.182
8         D    10.200    10.000    0.400    -0.400            0.200   ----|-*--
9        *P     1.138     0.000    1.000                    1.138    0.138


REFCIRC1   SET DIMENSION
              ACTUAL    NOMINAL   UPR-TOL   LWR-TOL    BONUS      DEV     ERROR
          X   2506.637  2506.637                        0.000
          Y    792.999   792.999                        0.000
          Z    912.645   912.645                        0.000
10         D    30.000    30.000    SET                 0.000


DIST1     DISTANCE BETWEEN
              ACTUAL    NOMINAL   UPR-TOL   LWR-TOL    BONUS      DEV     ERROR
11         S    81.221    81.209     0.500      -0.500       0.012   ----|*---
```

The actual information contained in such a report can be merged with information in the original inspection plan to produce a QIF Results instance file.

File QIF_RESULTS_SAMPLE.QIF


```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<QIFDocument xmlns="http://qifstandards.org/xsd/qif2" idMax="86"
versionQIF="2.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://qifstandards.org/xsd/qif2
../QIFApplications/QIFDocument.xsd">

  <Version>
    <ThisInstanceQPId>b61e7786-891f-4883-8d55-ef2b9abacfad</ThisInstanceQPId>
  </Version>

  <Header>
    <Application>
      <Name>Solidworks 2014</Name>
      <AddonName>CheckMate 14</AddonName>
      <AddonOrganization>Origin International Inc.</AddonOrganization>
    </Application>
  </Header>

  <PreInspectionTraceability>
    <InspectingOrganization>
      <Name>Origin International</Name>
      <Address>
        <StreetNumber>72</StreetNumber>
        <Street>Baynards Lane</Street>
        <Town>Richmond Hill</Town>
        <Region>Ontario</Region>
        <PostalCode>L4C 9B8</PostalCode>
        <FacsimileNumber>416 410 8313</FacsimileNumber>
        <TelephoneNumber>1-800-269-2509</TelephoneNumber>
        <ElectronicMailAddress>support@originintl.com</ElectronicMailAddress>
      </Address>
    </InspectingOrganization>
    <SupplierCode>North_Fab</SupplierCode>
    <PurchaseOrderNumber>PO123456</PurchaseOrderNumber>
    <AsmPathIds N="1">
      <Id>3</Id>
    </AsmPathIds>
    <ReportNumber>QIF 1</ReportNumber>
    <InspectionScope>DETAIL</InspectionScope>
    <InspectionMode>FAI_Full</InspectionMode>
    <ReferencedQIFPlanInstance>
      <ItemQPId>9fcebcb6-99da-44f9-b997-2ba76fd438dd</ItemQPId>
    </ReferencedQIFPlanInstance>
    <FormalStandard>
      <FormalStandardEnum>ASME-Y14.5-1994</FormalStandardEnum>
    </FormalStandard>
  </PreInspectionTraceability>
```



```
<FileUnits>
  <PrimaryUnits>
    <AngularUnit>
      <SIUnitName>radian</SIUnitName>
      <UnitName>degree</UnitName>
      <UnitConversion>
        <Factor>0.017453292519943</Factor>
      </UnitConversion>
    </AngularUnit>
    <LinearUnit>
      <SIUnitName>meter</SIUnitName>
      <UnitName>mm</UnitName>
      <UnitConversion>
        <Factor>0.001</Factor>
      </UnitConversion>
    </LinearUnit>
  </PrimaryUnits>
</FileUnits>

<DatumDefinitions>
  <DatumDefinition id="51">
    <DatumLabel>A</DatumLabel>
  </DatumDefinition>
  <DatumDefinition id="52">
    <DatumLabel>B</DatumLabel>
  </DatumDefinition>
  <DatumDefinition id="53">
    <DatumLabel>C</DatumLabel>
  </DatumDefinition>
  <DatumDefinition id="69">
    <DatumLabel>D</DatumLabel>
  </DatumDefinition>
  <DatumDefinition id="70">
    <DatumLabel>E</DatumLabel>
  </DatumDefinition>
</DatumDefinitions>

<DatumReferenceFrames>
  <DatumReferenceFrame id="12"/>
  <DatumReferenceFrame id="50">
    <Datum>
      <SimpleDatum>
        <DatumDefinitionId>51</DatumDefinitionId>
        <MaterialModifier>NONE</MaterialModifier>
        <ReferencedComponent>ACTUAL</ReferencedComponent>
      </SimpleDatum>
      <Precedence>
        <PrecedenceEnum>PRIMARY</PrecedenceEnum>
```



```
      </Precedence>
    </Datum>
    <Datum>
      <SimpleDatum>
        <DatumDefinitionId>52</DatumDefinitionId>
        <MaterialModifier>MAXIMUM</MaterialModifier>
        <ReferencedComponent>ACTUAL</ReferencedComponent>
      </SimpleDatum>
      <Precedence>
        <PrecedenceEnum>SECONDARY</PrecedenceEnum>
      </Precedence>
    </Datum>
    <Datum>
      <SimpleDatum>
        <DatumDefinitionId>53</DatumDefinitionId>
        <MaterialModifier>MAXIMUM</MaterialModifier>
        <ReferencedComponent>ACTUAL</ReferencedComponent>
      </SimpleDatum>
      <Precedence>
        <PrecedenceEnum>TERTIARY</PrecedenceEnum>
      </Precedence>
    </Datum>
  </DatumReferenceFrame>
  <DatumReferenceFrame id="68">
    <Datum>
      <SimpleDatum>
        <DatumDefinitionId>51</DatumDefinitionId>
        <MaterialModifier>NONE</MaterialModifier>
        <ReferencedComponent>ACTUAL</ReferencedComponent>
      </SimpleDatum>
      <Precedence>
        <PrecedenceEnum>PRIMARY</PrecedenceEnum>
      </Precedence>
    </Datum>
    <Datum>
      <SimpleDatum>
        <DatumDefinitionId>69</DatumDefinitionId>
        <MaterialModifier>LEAST</MaterialModifier>
        <ReferencedComponent>ACTUAL</ReferencedComponent>
      </SimpleDatum>
      <Precedence>
        <PrecedenceEnum>SECONDARY</PrecedenceEnum>
      </Precedence>
    </Datum>
    <Datum>
      <SimpleDatum>
        <DatumDefinitionId>70</DatumDefinitionId>
        <MaterialModifier>LEAST</MaterialModifier>
        <ReferencedComponent>ACTUAL</ReferencedComponent>
```

```
        </SimpleDatum>
        <Precedence>
          <PrecedenceEnum>TERTIARY</PrecedenceEnum>
        </Precedence>
      </Datum>
    </DatumReferenceFrame>
  </DatumReferenceFrames>

  <MeasurementResources>
    <MeasurementDevices>
      <MeasurementDevice id="15">
        <Name>CMM</Name>
      </MeasurementDevice>
      <MeasurementDevice id="56">
        <Name>GAGE PINS</Name>
      </MeasurementDevice>
      <MeasurementDevice id="65">
        <Name>CALIPERS</Name>
      </MeasurementDevice>
    </MeasurementDevices>
  </MeasurementResources>

  <Product>
    <PartSet N="1">
      <Part id="1">
        <Name>WING_MIR_REENF</Name>
        <QPId>2bbeb82a-96bf-4f1e-a327-4ba3500490e1</QPId>
        <ModelNumber>QM_X_123456</ModelNumber>
        <Description>Wing Mirror Re-enforcement</Description>
        <Version>1.02</Version>
        <DefinitionExternal id="6">
          <PrintedDrawing id="5">
            <Name>sheet2_solid4</Name>
            <Version>1.0.0</Version>
            <Description>Sample QIF elements</Description>
            <DrawingNumber>#1</DrawingNumber>
            <AdditionalChanges>none</AdditionalChanges>
            <Location>Cabinet 17, Drawer 3</Location>
          </PrintedDrawing>
        </DefinitionExternal>
      </Part>
    </PartSet>
    <ComponentSet N="1">
      <Component id="2">
        <QPId>af4d1612-8918-4a69-a716-709beee7a953</QPId>
        <Part>
          <Id>1</Id>
        </Part>
      </Component>
```

```
    </ComponentSet>
    <AsmPaths N="1">
      <AsmPath id="3">
        <ComponentIds N="1">
          <Id>2</Id>
        </ComponentIds>
      </AsmPath>
    </AsmPaths>
  </Product>

  <Features>
    <FeatureDefinitions>
      <EdgePointFeatureDefinition id="7">
        <InternalExternal>EXTERNAL</InternalExternal>
      </EdgePointFeatureDefinition>
      <PointFeatureDefinition id="17"/>
      <PointFeatureDefinition id="33"/>
      <CircleFeatureDefinition id="41">
        <InternalExternal>INTERNAL</InternalExternal>
        <Diameter>10</Diameter>
      </CircleFeatureDefinition>
      <CircleFeatureDefinition id="58">
        <InternalExternal>INTERNAL</InternalExternal>
        <Diameter>10</Diameter>
      </CircleFeatureDefinition>
      <CircleFeatureDefinition id="74">
        <InternalExternal>NOT_APPLICABLE</InternalExternal>
        <Diameter>30</Diameter>
      </CircleFeatureDefinition>
    </FeatureDefinitions>
    <FeatureNominals>
      <EdgePointFeatureNominal id="8">
        <FeatureDefinitionId>7</FeatureDefinitionId>
        <Location>2460.7099609375 770.604614257813
944.993591308594</Location>
        <Normal>-0.735465884156759 -0.307902932144912
0.603560864882807</Normal>
        <AdjacentNormal>0.0411800179070265 -0.909449806293278 -
0.413769568671133</AdjacentNormal>
      </EdgePointFeatureNominal>
      <PointFeatureNominal id="18">
        <FeatureDefinitionId>17</FeatureDefinitionId>
        <Location>2466.72924804688 774.269897460938
945.002746582031</Location>
        <Normal>0.731520704665031 0.146124942551 -0.665976696146977</Normal>
      </PointFeatureNominal>
      <PointFeatureNominal id="34">
        <FeatureDefinitionId>33</FeatureDefinitionId>
```

```
        <Location>2536.49487304688 782.806213378906
920.002197265625</Location>
        <Normal>-0.7609260410224 -0.648538030188332 -
0.0197479997320212</Normal>
      </PointFeatureNominal>
      <CircleFeatureNominal id="42">
        <FeatureDefinitionId>41</FeatureDefinitionId>
        <Location>0.0558150216639657 -0.907624351305543 -
0.416056150385791</Location>
        <Normal>2433.974609375 800.617431640625 890.049621582031</Normal>
      </CircleFeatureNominal>
      <CircleFeatureNominal id="59">
        <FeatureDefinitionId>58</FeatureDefinitionId>
        <Location>0.0418550117439733 -0.909169210310978 -
0.414318120547998</Location>
        <Normal>2496.25146484375 780.842590332031 938.272277832031</Normal>
      </CircleFeatureNominal>
      <CircleFeatureNominal id="75">
        <FeatureDefinitionId>74</FeatureDefinitionId>
        <Location>0.0418467877611875 -0.909189553580246 -
0.414274307693132</Location>
        <Normal>2506.63672251543 792.999124488284 912.644676969086</Normal>
      </CircleFeatureNominal>
    </FeatureNominals>
    <FeatureItems>
      <EdgePointFeatureItem id="9">
        <Attributes N="1">
          <AttributeStr name="Comment" value="An edge point with uniform
profile characteristic"/>
        </Attributes>
        <FeatureNominalId>8</FeatureNominalId>
        <FeatureName>TRIM1</FeatureName>
        <QPId>aeba4615-65db-4f7d-a2e8-a7928c80117f</QPId>
        <DeterminationMode>
          <Checked>
            <CheckDetails>
              <Measured/>
            </CheckDetails>
          </Checked>
        </DeterminationMode>
      </EdgePointFeatureItem>
      <PointFeatureItem id="19">
        <Attributes N="1">
          <AttributeStr name="Comment" value="A point on a surface with
coordinate characteristics"/>
        </Attributes>
        <FeatureNominalId>18</FeatureNominalId>
        <FeatureName>SURF1</FeatureName>
        <QPId>263c98a7-9786-4408-b62f-dda79a07daad</QPId>
```

```
        <DeterminationMode>
          <Checked>
            <CheckDetails>
              <Measured/>
            </CheckDetails>
          </Checked>
        </DeterminationMode>
      </PointFeatureItem>
      <PointFeatureItem id="35">
        <Attributes N="1">
          <AttributeStr name="Comment" value="A point on a surface with non-
uniform profile characteristic"/>
        </Attributes>
        <FeatureNominalId>34</FeatureNominalId>
        <FeatureName>SURF2</FeatureName>
        <QPId>63799169-1d43-41ee-b1ad-451f9d8a8f4e</QPId>
        <DeterminationMode>
          <Checked>
            <CheckDetails>
              <Measured/>
            </CheckDetails>
          </Checked>
        </DeterminationMode>
      </PointFeatureItem>
      <CircleFeatureItem id="43">
        <Attributes N="1">
          <AttributeStr name="Comment" value="A hole with position and gage-
pin diameter"/>
        </Attributes>
        <FeatureNominalId>42</FeatureNominalId>
        <FeatureName>HOLE1</FeatureName>
        <QPId>f29bb618-e50d-4dbb-b4b4-83b28755c741</QPId>
        <DeterminationMode>
          <Checked>
            <CheckDetails>
              <Measured/>
            </CheckDetails>
          </Checked>
        </DeterminationMode>
      </CircleFeatureItem>
      <CircleFeatureItem id="60">
        <Attributes N="1">
          <AttributeStr name="Comment" value="A hole with position and
caliper diameter"/>
        </Attributes>
        <FeatureNominalId>59</FeatureNominalId>
        <FeatureName>HOLE2</FeatureName>
        <QPId>5bb191b0-7a18-4393-8f24-1353ac8365c7</QPId>
        <DeterminationMode>
```



```
          <Checked>
            <CheckDetails>
              <Measured/>
            </CheckDetails>
          </Checked>
        </DeterminationMode>
      </CircleFeatureItem>
      <CircleFeatureItem id="76">
        <Attributes N="1">
          <AttributeStr name="Comment" value="A non-measureable reference
circle"/>
        </Attributes>
        <FeatureNominalId>75</FeatureNominalId>
        <FeatureName>REFCIRC1</FeatureName>
        <QPId>6443f863-c559-42f2-b62a-48c0172d3395</QPId>
        <DeterminationMode>
          <Set/>
        </DeterminationMode>
      </CircleFeatureItem>
    </FeatureItems>
  </Features>

  <Characteristics>
    <FormalStandard>
      <FormalStandardEnum>ASME-Y14.5-1994</FormalStandardEnum>
    </FormalStandard>
    <CharacteristicDefinitions>
      <PointProfileCharacteristicDefinition id="11">
        <ToleranceValue>4</ToleranceValue>
        <DatumReferenceFrameId>12</DatumReferenceFrameId>
      </PointProfileCharacteristicDefinition>
      <LinearCoordinateCharacteristicDefinition id="21">
        <NonTolerance>MEASURED</NonTolerance>
      </LinearCoordinateCharacteristicDefinition>
      <LinearCoordinateCharacteristicDefinition id="25">
        <Tolerance>
          <MaxValue>0.2</MaxValue>
          <MinValue>-0.2</MinValue>
          <DefinedAsLimit>false</DefinedAsLimit>
        </Tolerance>
      </LinearCoordinateCharacteristicDefinition>
      <LinearCoordinateCharacteristicDefinition id="29">
        <Tolerance>
          <MaxValue>945.20274658203095</MaxValue>
          <MinValue>944.80274658203086</MinValue>
          <DefinedAsLimit>true</DefinedAsLimit>
        </Tolerance>
      </LinearCoordinateCharacteristicDefinition>
      <PointProfileCharacteristicDefinition id="37">
```

```
      <ToleranceValue>1.5</ToleranceValue>
      <OuterDisposition>1</OuterDisposition>
      <DatumReferenceFrameId>12</DatumReferenceFrameId>
    </PointProfileCharacteristicDefinition>
    <DiameterCharacteristicDefinition id="45">
      <Tolerance>
        <MaxValue>0.4</MaxValue>
        <MinValue>-0.4</MinValue>
        <DefinedAsLimit>false</DefinedAsLimit>
      </Tolerance>
    </DiameterCharacteristicDefinition>
    <PositionCharacteristicDefinition id="49">
      <ToleranceValue>1</ToleranceValue>
      <DatumReferenceFrameId>50</DatumReferenceFrameId>
      <MaterialCondition>MAXIMUM</MaterialCondition>
      <ZoneShape>
        <DiametricalZone/>
      </ZoneShape>
    </PositionCharacteristicDefinition>
    <DiameterCharacteristicDefinition id="62">
      <Tolerance>
        <MaxValue>10.4</MaxValue>
        <MinValue>9.6</MinValue>
        <DefinedAsLimit>true</DefinedAsLimit>
      </Tolerance>
    </DiameterCharacteristicDefinition>
    <PositionCharacteristicDefinition id="67">
      <ToleranceValue>1</ToleranceValue>
      <DatumReferenceFrameId>68</DatumReferenceFrameId>
      <MaterialCondition>REGARDLESS</MaterialCondition>
      <ZoneShape>
        <DiametricalZone/>
      </ZoneShape>
    </PositionCharacteristicDefinition>
    <DiameterCharacteristicDefinition id="78">
      <NonTolerance>SET</NonTolerance>
    </DiameterCharacteristicDefinition>
    <DistanceBetweenCharacteristicDefinition id="82">
      <Tolerance>
        <MaxValue>0.5</MaxValue>
        <MinValue>-0.5</MinValue>
        <DefinedAsLimit>false</DefinedAsLimit>
      </Tolerance>
    </DistanceBetweenCharacteristicDefinition>
  </CharacteristicDefinitions>
  <CharacteristicNominals>
    <PointProfileCharacteristicNominal id="13">
      <CharacteristicDefinitionId>11</CharacteristicDefinitionId>
    </PointProfileCharacteristicNominal>
```



```
      <LinearCoordinateCharacteristicNominal id="22">
        <CharacteristicDefinitionId>21</CharacteristicDefinitionId>
        <TargetValue>2466.729248046875</TargetValue>
        <Direction>XAXIS</Direction>
      </LinearCoordinateCharacteristicNominal>
      <LinearCoordinateCharacteristicNominal id="26">
        <CharacteristicDefinitionId>25</CharacteristicDefinitionId>
        <TargetValue>774.26989746093795</TargetValue>
        <Direction>YAXIS</Direction>
      </LinearCoordinateCharacteristicNominal>
      <LinearCoordinateCharacteristicNominal id="30">
        <CharacteristicDefinitionId>29</CharacteristicDefinitionId>
        <Direction>ZAXIS</Direction>
      </LinearCoordinateCharacteristicNominal>
      <PointProfileCharacteristicNominal id="38">
        <CharacteristicDefinitionId>37</CharacteristicDefinitionId>
      </PointProfileCharacteristicNominal>
      <DiameterCharacteristicNominal id="46">
        <CharacteristicDefinitionId>45</CharacteristicDefinitionId>
        <TargetValue>10</TargetValue>
      </DiameterCharacteristicNominal>
      <PositionCharacteristicNominal id="54">
        <CharacteristicDefinitionId>49</CharacteristicDefinitionId>
      </PositionCharacteristicNominal>
      <DiameterCharacteristicNominal id="63">
        <CharacteristicDefinitionId>62</CharacteristicDefinitionId>
      </DiameterCharacteristicNominal>
      <PositionCharacteristicNominal id="71">
        <CharacteristicDefinitionId>67</CharacteristicDefinitionId>
      </PositionCharacteristicNominal>
      <DiameterCharacteristicNominal id="79">
        <CharacteristicDefinitionId>78</CharacteristicDefinitionId>
        <TargetValue>30</TargetValue>
      </DiameterCharacteristicNominal>
      <DistanceBetweenCharacteristicNominal id="83">
        <CharacteristicDefinitionId>82</CharacteristicDefinitionId>
        <TargetValue>81.208839738425993</TargetValue>
        <AnalysisMode>THREEDIMENSIONAL</AnalysisMode>
      </DistanceBetweenCharacteristicNominal>
    </CharacteristicNominals>
    <CharacteristicItems>
      <PointProfileCharacteristicItem id="14">
        <Attributes N="1">
          <AttributeStr name="Comment" value="A uniform profile zone ±2.0
with empty DRF"/>
        </Attributes>
        <Name>5</Name>
        <QPId>6f2a6f83-3ff7-4327-b00c-98208361301a</QPId>
        <KeyCharacteristic>
```

```
          <Designator>5</Designator>
          <Criticality>MINOR</Criticality>
        </KeyCharacteristic>
        <FeatureItemIds N="1">
          <Id>9</Id>
        </FeatureItemIds>
        <MeasurementDeviceIds N="1">
          <Id>15</Id>
        </MeasurementDeviceIds>
        <CharacteristicNominalId>13</CharacteristicNominalId>
        <LocationOnDrawing>
          <DrawingId>5</DrawingId>
          <SheetNumber>SHEET1</SheetNumber>
          <DrawingZone>C2</DrawingZone>
        </LocationOnDrawing>
      </PointProfileCharacteristicItem>
      <LinearCoordinateCharacteristicItem id="23">
        <Attributes N="1">
          <AttributeStr name="Comment" value="A basic coordinate dimension"/>
        </Attributes>
        <Name>1</Name>
        <QPId>fdb07604-ac7e-4f7c-acc8-0860de517fd4</QPId>
        <KeyCharacteristic>
          <Designator>1</Designator>
          <Criticality>REF</Criticality>
        </KeyCharacteristic>
        <FeatureItemIds N="1">
          <Id>19</Id>
        </FeatureItemIds>
        <MeasurementDeviceIds N="1">
          <Id>15</Id>
        </MeasurementDeviceIds>
        <CharacteristicNominalId>22</CharacteristicNominalId>
        <LocationOnDrawing>
          <DrawingId>5</DrawingId>
          <SheetNumber>SHEET1</SheetNumber>
          <DrawingZone>D3</DrawingZone>
        </LocationOnDrawing>
      </LinearCoordinateCharacteristicItem>
      <LinearCoordinateCharacteristicItem id="27">
        <Attributes N="1">
          <AttributeStr name="Comment" value="A bi-directional coordinate
tolerance"/>
        </Attributes>
        <Name>2</Name>
        <QPId>445298bf-91a2-4ec6-8fa4-3b2c029eeb6e</QPId>
        <KeyCharacteristic>
          <Designator>2</Designator>
          <Criticality>MINOR</Criticality>
```



```
        </KeyCharacteristic>
        <FeatureItemIds N="1">
          <Id>19</Id>
        </FeatureItemIds>
        <MeasurementDeviceIds N="1">
          <Id>15</Id>
        </MeasurementDeviceIds>
        <CharacteristicNominalId>26</CharacteristicNominalId>
        <LocationOnDrawing>
          <DrawingId>5</DrawingId>
          <SheetNumber>SHEET1</SheetNumber>
          <DrawingZone>D3</DrawingZone>
        </LocationOnDrawing>
      </LinearCoordinateCharacteristicItem>
      <LinearCoordinateCharacteristicItem id="31">
        <Attributes N="1">
          <AttributeStr name="Comment" value="A limit coordinate tolerance"/>
        </Attributes>
        <Name>3</Name>
        <QPId>187ce9c9-7125-47b0-869b-d37a9722a35d</QPId>
        <KeyCharacteristic>
          <Designator>3</Designator>
          <Criticality>MAJOR</Criticality>
        </KeyCharacteristic>
        <FeatureItemIds N="1">
          <Id>19</Id>
        </FeatureItemIds>
        <MeasurementDeviceIds N="1">
          <Id>15</Id>
        </MeasurementDeviceIds>
        <CharacteristicNominalId>30</CharacteristicNominalId>
        <LocationOnDrawing>
          <DrawingId>5</DrawingId>
          <SheetNumber>SHEET1</SheetNumber>
          <DrawingZone>D3</DrawingZone>
        </LocationOnDrawing>
      </LinearCoordinateCharacteristicItem>
      <PointProfileCharacteristicItem id="39">
        <Attributes N="1">
          <AttributeStr name="Comment" value="An offset profile zone +1.0/-
0.5 to ABC"/>
        </Attributes>
        <Name>4</Name>
        <QPId>7ad19cce-458a-4d0a-9436-6a2d2683b018</QPId>
        <KeyCharacteristic>
          <Designator>4</Designator>
          <Criticality>CRITICAL</Criticality>
        </KeyCharacteristic>
        <FeatureItemIds N="1">
```



```
          <Id>35</Id>
        </FeatureItemIds>
        <MeasurementDeviceIds N="1">
          <Id>15</Id>
        </MeasurementDeviceIds>
        <CharacteristicNominalId>38</CharacteristicNominalId>
        <LocationOnDrawing>
          <DrawingId>5</DrawingId>
          <SheetNumber>SHEET1</SheetNumber>
          <DrawingZone>B3</DrawingZone>
        </LocationOnDrawing>
      </PointProfileCharacteristicItem>
      <DiameterCharacteristicItem id="47">
        <Attributes N="1">
          <AttributeStr name="Comment" value="A bi-directional diameter
tolerance"/>
        </Attributes>
        <Name>6</Name>
        <QPId>4e609ab8-7d1c-4317-9b16-5fea194f9d7e</QPId>
        <KeyCharacteristic>
          <Designator>6</Designator>
          <Criticality>MINOR</Criticality>
        </KeyCharacteristic>
        <FeatureItemIds N="1">
          <Id>43</Id>
        </FeatureItemIds>
        <MeasurementDeviceIds N="1">
          <Id>15</Id>
        </MeasurementDeviceIds>
        <CharacteristicNominalId>46</CharacteristicNominalId>
        <LocationOnDrawing>
          <DrawingId>5</DrawingId>
          <SheetNumber>SHEET1</SheetNumber>
          <DrawingZone>C1</DrawingZone>
        </LocationOnDrawing>
      </DiameterCharacteristicItem>
      <PositionCharacteristicItem id="55">
        <Attributes N="1">
          <AttributeStr name="Comment" value="MMC diametrical position to
AB(m)C(m)"/>
        </Attributes>
        <Name>7</Name>
        <QPId>4438e85a-441a-420f-93d8-c6ed52fae87e</QPId>
        <KeyCharacteristic>
          <Designator>7</Designator>
          <Criticality>CRITICAL</Criticality>
        </KeyCharacteristic>
        <FeatureItemIds N="1">
          <Id>43</Id>
```

```
        </FeatureItemIds>
        <MeasurementDeviceIds N="1">
          <Id>56</Id>
        </MeasurementDeviceIds>
        <CharacteristicNominalId>54</CharacteristicNominalId>
        <LocationOnDrawing>
          <DrawingId>5</DrawingId>
          <SheetNumber>SHEET1</SheetNumber>
          <DrawingZone>C1</DrawingZone>
        </LocationOnDrawing>
      </PositionCharacteristicItem>
      <DiameterCharacteristicItem id="64">
        <Attributes N="1">
          <AttributeStr name="Comment" value="Limit diameter tolerance with
caliper "/>
        </Attributes>
        <Name>8</Name>
        <QPId>a5a0d3a2-e6c4-4be9-93fe-18ba95d8372c</QPId>
        <KeyCharacteristic>
          <Designator>8</Designator>
        </KeyCharacteristic>
        <FeatureItemIds N="1">
          <Id>60</Id>
        </FeatureItemIds>
        <MeasurementDeviceIds N="1">
          <Id>65</Id>
        </MeasurementDeviceIds>
        <CharacteristicNominalId>63</CharacteristicNominalId>
        <LocationOnDrawing>
          <DrawingId>5</DrawingId>
          <SheetNumber>SHEET1</SheetNumber>
          <DrawingZone>C3</DrawingZone>
        </LocationOnDrawing>
      </DiameterCharacteristicItem>
      <PositionCharacteristicItem id="72">
        <Attributes N="1">
          <AttributeStr name="Comment" value="RFS position to AD(l)E(l)"/>
        </Attributes>
        <Name>9</Name>
        <QPId>c79e4f25-f235-4886-88a3-b8a2eeaba096</QPId>
        <KeyCharacteristic>
          <Designator>9</Designator>
          <Criticality>MINOR</Criticality>
        </KeyCharacteristic>
        <FeatureItemIds N="1">
          <Id>60</Id>
        </FeatureItemIds>
        <MeasurementDeviceIds N="1">
          <Id>15</Id>
```

```
        </MeasurementDeviceIds>
        <CharacteristicNominalId>71</CharacteristicNominalId>
        <LocationOnDrawing>
          <DrawingId>5</DrawingId>
          <SheetNumber>SHEET1</SheetNumber>
          <DrawingZone>C3</DrawingZone>
        </LocationOnDrawing>
      </PositionCharacteristicItem>
      <DiameterCharacteristicItem id="80">
        <Name>10</Name>
        <QPId>ee9bb7c3-31fb-40a4-841d-4f1b9fc885bd</QPId>
        <KeyCharacteristic>
          <Designator>10</Designator>
        </KeyCharacteristic>
        <FeatureItemIds N="1">
          <Id>76</Id>
        </FeatureItemIds>
        <CharacteristicNominalId>79</CharacteristicNominalId>
      </DiameterCharacteristicItem>
      <DistanceBetweenCharacteristicItem id="84">
        <Attributes N="1">
          <AttributeStr name="Comment" value="Distance between HOLE1 and
HOLE2"/>
        </Attributes>
        <Name>DIST1</Name>
        <QPId>3e6797f5-b443-4b15-9108-055190223837</QPId>
        <KeyCharacteristic>
          <Designator>11</Designator>
        </KeyCharacteristic>
        <FeatureItemIds N="2">
          <Id>60</Id>
          <Id>43</Id>
        </FeatureItemIds>
        <MeasurementDeviceIds N="1">
          <Id>15</Id>
        </MeasurementDeviceIds>
        <CharacteristicNominalId>83</CharacteristicNominalId>
        <LocationOnDrawing>
          <DrawingId>5</DrawingId>
          <SheetNumber>SHEET1</SheetNumber>
          <DrawingZone>B2</DrawingZone>
        </LocationOnDrawing>
      </DistanceBetweenCharacteristicItem>
    </CharacteristicItems>
  </Characteristics>

  <MeasurementsResults>
    <MeasurementResults id="86">
```

```
      <ThisResultsInstanceQPId>8521ff0f-4c05-4f13-a2be-
1386190f75a6</ThisResultsInstanceQPId>
      <ExternalFileReference>
        <URI>C:\R19_parts\5802804.res</URI>
        <FileSpec>
          <OtherFileSpec>res</OtherFileSpec>
        </FileSpec>
        <Description>Zeiss Umess CMM Results</Description>
      </ExternalFileReference>
      <MeasuredFeatures>
        <FeatureActuals>
          <EdgePointFeatureActual id="10">
            <FeatureItemId>9</FeatureItemId>
            <Location>2460.72 770.62 944.98</Location>
            <Normal>-0.735465884156759 -0.307902932144912
0.603560864882807</Normal>
          </EdgePointFeatureActual>
          <PointFeatureActual id="20">
            <FeatureItemId>19</FeatureItemId>
            <Location>2466.9 774.31 944.84</Location>
          </PointFeatureActual>
          <PointFeatureActual id="36">
            <FeatureItemId>35</FeatureItemId>
            <Location>2537.17 783.38 920.02</Location>
          </PointFeatureActual>
          <CircleFeatureActual id="44">
            <FeatureItemId>43</FeatureItemId>
            <Location>2434.01 801.52505599193 889.98</Location>
            <Normal>0.0558150216639657 -0.907624351305543 -
0.416056150385791</Normal>
            <Diameter>9.499476</Diameter>
          </CircleFeatureActual>
          <CircleFeatureActual id="61">
            <FeatureItemId>60</FeatureItemId>
            <Location>2496.39 781.751759542342 938.09</Location>
            <Normal>0.0418550117439733 -0.909169210310978 -
0.414318120547998</Normal>
            <Diameter>10.199987999999999</Diameter>
          </CircleFeatureActual>
          <CircleFeatureActual id="77">
            <FeatureItemId>76</FeatureItemId>
            <Location>2506.63672251543 792.999124488284
912.644676969086</Location>
            <Normal>0.0418467877611875 -0.909189553580246 -
0.414274307693132</Normal>
            <Diameter>30</Diameter>
          </CircleFeatureActual>
        </FeatureActuals>
      </MeasuredFeatures>
```

```
    <MeasuredCharacteristics>
      <CharacteristicActuals>
        <PointProfileCharacteristicActual id="16">
          <Status>
            <CharacteristicStatusEnum>PASS</CharacteristicStatusEnum>
          </Status>
          <CharacteristicItemId>14</CharacteristicItemId>
          <FeatureActualIds N="1">
            <Id>10</Id>
          </FeatureActualIds>
          <NonConformanceDesignator>NA</NonConformanceDesignator>
          <Value>-0.020323885080472</Value>
        </PointProfileCharacteristicActual>
        <LinearCoordinateCharacteristicActual id="24">
          <Status>
            <CharacteristicStatusEnum>BASIC</CharacteristicStatusEnum>
          </Status>
          <CharacteristicItemId>23</CharacteristicItemId>
          <FeatureActualIds N="1">
            <Id>20</Id>
          </FeatureActualIds>
          <NonConformanceDesignator>NA</NonConformanceDesignator>
          <TypeOfCoordinates>
            <CoordinateEnum>CARTESIAN_3D</CoordinateEnum>
          </TypeOfCoordinates>
          <Value>2466.9000000000001</Value>
        </LinearCoordinateCharacteristicActual>
        <LinearCoordinateCharacteristicActual id="28">
          <Status>
            <CharacteristicStatusEnum>PASS</CharacteristicStatusEnum>
          </Status>
          <CharacteristicItemId>27</CharacteristicItemId>
          <FeatureActualIds N="1">
            <Id>20</Id>
          </FeatureActualIds>
          <NonConformanceDesignator>NA</NonConformanceDesignator>
          <TypeOfCoordinates>
            <CoordinateEnum>CARTESIAN_3D</CoordinateEnum>
          </TypeOfCoordinates>
          <Value>774.30999999999995</Value>
        </LinearCoordinateCharacteristicActual>
        <LinearCoordinateCharacteristicActual id="32">
          <Status>
            <CharacteristicStatusEnum>PASS</CharacteristicStatusEnum>
          </Status>
          <CharacteristicItemId>31</CharacteristicItemId>
          <FeatureActualIds N="1">
            <Id>20</Id>
          </FeatureActualIds>
```

```
    <NonConformanceDesignator>NA</NonConformanceDesignator>
    <TypeOfCoordinates>
      <CoordinateEnum>CARTESIAN_3D</CoordinateEnum>
    </TypeOfCoordinates>
    <Value>944.84000000000003</Value>
  </LinearCoordinateCharacteristicActual>
  <PointProfileCharacteristicActual id="40">
    <Status>
      <CharacteristicStatusEnum>FAIL</CharacteristicStatusEnum>
    </Status>
    <CharacteristicItemId>39</CharacteristicItemId>
    <FeatureActualIds N="1">
      <Id>36</Id>
    </FeatureActualIds>
    <NonConformanceDesignator>1234</NonConformanceDesignator>
    <Value>-0.886195693015566</Value>
  </PointProfileCharacteristicActual>
  <DiameterCharacteristicActual id="48">
    <Status>
      <CharacteristicStatusEnum>FAIL</CharacteristicStatusEnum>
    </Status>
    <CharacteristicItemId>47</CharacteristicItemId>
    <FeatureActualIds N="1">
      <Id>44</Id>
    </FeatureActualIds>
    <NonConformanceDesignator>1234</NonConformanceDesignator>
    <Value>9.499476</Value>
  </DiameterCharacteristicActual>
  <PositionCharacteristicActual id="57">
    <Status>
      <CharacteristicStatusEnum>PASS</CharacteristicStatusEnum>
    </Status>
    <CharacteristicItemId>55</CharacteristicItemId>
    <FeatureActualIds N="1">
      <Id>44</Id>
    </FeatureActualIds>
    <NonConformanceDesignator>NA</NonConformanceDesignator>
    <Value>0.897298445619386</Value>
  </PositionCharacteristicActual>
  <DiameterCharacteristicActual id="66">
    <Status>
      <CharacteristicStatusEnum>PASS</CharacteristicStatusEnum>
    </Status>
    <CharacteristicItemId>64</CharacteristicItemId>
    <FeatureActualIds N="1">
      <Id>61</Id>
    </FeatureActualIds>
    <NonConformanceDesignator>NA</NonConformanceDesignator>
    <Value>10.199987999999999</Value>
```

```
        </DiameterCharacteristicActual>
        <PositionCharacteristicActual id="73">
          <Status>
            <CharacteristicStatusEnum>FAIL</CharacteristicStatusEnum>
          </Status>
          <CharacteristicItemId>72</CharacteristicItemId>
          <FeatureActualIds N="1">
            <Id>61</Id>
          </FeatureActualIds>
          <NonConformanceDesignator>1234</NonConformanceDesignator>
          <Value>1.137681133150095</Value>
        </PositionCharacteristicActual>
        <DiameterCharacteristicActual id="81">
          <Status>
            <CharacteristicStatusEnum>BASIC</CharacteristicStatusEnum>
          </Status>
          <CharacteristicItemId>80</CharacteristicItemId>
          <FeatureActualIds N="1">
            <Id>77</Id>
          </FeatureActualIds>
          <Value>30</Value>
        </DiameterCharacteristicActual>
        <DistanceBetweenCharacteristicActual id="85">
          <Status>
            <CharacteristicStatusEnum>PASS</CharacteristicStatusEnum>
          </Status>
          <CharacteristicItemId>84</CharacteristicItemId>
          <FeatureActualIds N="2">
            <Id>61</Id>
            <Id>44</Id>
          </FeatureActualIds>
          <Value>81.220808617516994</Value>
        </DistanceBetweenCharacteristicActual>
      </CharacteristicActuals>
    </MeasuredCharacteristics>
    <InspectionStatus>
      <InspectionStatusEnum>FAIL</InspectionStatusEnum>
    </InspectionStatus>
    <ActualComponentIds N="1">
      <Id>4</Id>
    </ActualComponentIds>
  </MeasurementResults>
  <ActualComponentSet N="1">
    <ActualComponent id="4">
      <SerialNumber>Run 3, Bin 17</SerialNumber>
      <Status>
        <InspectionStatusEnum>FAIL</InspectionStatusEnum>
      </Status>
      <AsmPathId>3</AsmPathId>
```

```
      </ActualComponent>
    </ActualComponentSet>
    <InspectionTraceability>
      <ReportPreparer>
        <Name>John Doe</Name>
        <EmployeeId>123-456</EmployeeId>
      </ReportPreparer>
      <ReportPreparationDate>2014-07-30T11:14:06</ReportPreparationDate>
    </InspectionTraceability>
  </MeasurementsResults>

</QIFDocument>
```


*** end of Results sample file ***

Annex E – ISO GPS (1101) support in QIF 2.0
(informative)

Preface
Most dimensional and geometrical tolerancing standards used in industry are based on either American Society of Mechanical Engineers (ASME) or International Organization for Standardization (ISO) standards. Both ISO Geometrical Product Specifications (GPS) and ASME Y14.5 help establish uniform practices for stating and interpreting product design intent. With QIF 2.0 being an ANSI standard, it focuses on ASME Y14.5, however the QIF contains significant support of ISO GPS documented practices which are noteworthy.

ISO GPS and ASME Y14.5 commonality
ISO GPS and ASME Y14.5 share many symbols and semantic concepts at a high level although they may differ at a more detailed level. It is by virtue of this commonality that there is a de facto support in QIF 2.0 of much of ISO GPS concepts.

QIF designation of formal tolerance standards
The primary mechanism for differentiating the detailed meaning of QIF types and *element*s across different standards is the required use of the **FormalStandardEnum** *element* of ***FormalStandardEnumType*** when characteristics are defined. Currently, QIF can identify three different versions of the ISO GPS main standard:

ISO1101:1983
ISO1101:2004
ISO1101:2012

ISO GPS specific items in QIF 2.0
ISO GPS includes several symbols; datum, tolerance, and feature control frame modifiers; and semantic concepts that do not have ASME Y14.5 counterparts. QIF implements support of these with types and *element*s specific to ISO GPS as listed below.

The following is a list of QIF 2.0 types, *element*s and enumerations specific to ISO GPS support:

1. Median feature Ⓐ
   CharacteristicDefinitionBaseType/**MedianFeature** optional Boolean *element*

2. Envelope requirement □
   CharacteristicDefinitionBaseType/**EnvelopeRequirement** optionalBoolean *element*

3. Common zone modifier C Z
   CharacteristicDefinitionBaseType/**CommonZone** optional Boolean *element*

4. United feature modifier U F
   CharacteristicDefinitionBaseType/**UnitedFeature** optional Boolean *element*

5. Separate zone modifier S Z
   CharacteristicDefinitionBaseType/**SeparateZone** optional Boolean *element*

6. Direction feature modifier ◀ // | B
   GeometricCharacteristicDefinitionBaseType/**DirectionFeature** optional *element*, the type of which is ***DirectionFeatureType*** (a complex type)

7. Collection plane modifier ○ // | B
   GeometricCharacteristicDefinitionBaseType/**CollectionPlane** optional *element*, the type of which is ***CollectionPlaneType*** (a complex type)

8. Intersection plane modifier ◁ // | B
   GeometricCharacteristicDefinitionBaseType/**IntersectionPlane** optional *element*, the type of which is ***IntersectionPlaneType*** (a complex type)

9. Orientation plane modifier ◁ // | B ▷
   GeometricCharacteristicDefinitionBaseType/**OrientationPlane** optional *element*, the type of which is ***OrientationPlaneType*** (a complex type)

10. Linear element modifier LE (as Y14.5M equivalent EACH ELEMENT)
    OrientationCharacteristicDefinitionBaseType/**EachElement** optional Boolean *element*

11. Not convex modifier NC
    FlatnessCharacteristicDefinitionType/**NotConvex** optional Boolean *element*

12. Orientation only modifier ><
    PositionCharacteristicDefinitionType/**OrientationOnly** optional Boolean *element*, and ***ProfileCharacteristicDefinitionBaseType***/**OrientationOnly** optional Boolean *element*

13. Unequally disposed zone modifier UZ (optional choice between ISO unequally disposed zone UZ and ANSI outer disposition Ⓤ methods)
    ProfileCharacteristicDefinitionBaseType/**UnequallyDisposedZone** linear value optional *element*

14. ALLOVERTHISSIDE and ALLAROUNDTHISSIDE additions to ANSI ALLOVER and ALLAROUND
    ExtentEnumType

15. Offset zone modifier OZ
    ProfileCharacteristicDefinitionBaseType/**OffsetZone** optional Boolean *element*

16. Projected datum modifier Ⓟ
    DatumType/**ProjectedDatum** linear value optional *element*

17. Threaded datum diameter modifiers PD, MD, and LD
    DatumType/**DiameterModifier** optional *element*, the type of which is ***DiameterModifierEnumType*** (a simple type)

18. Datum section modifiers ACS and ALS
    DatumType/**SectionModifier** optional *element*, the type of which is

SectionModifierEnumType (a simple type)

19. Contacting feature datum modifier CF
DatumType/**ContactingFeature** optional Boolean *element*

20. Distance variable modifier DV
DatumType/**DistanceVariable** optional Boolean *element*

21. Datum fixed modifier DF
DatumType/**DatumFixed** optional Boolean *element*

22. Datum reduction modifiers PT, SL and PL
DatumType/**ReducedDatum** optional *element*, the type of which is ***ReducedDatumEnumType*** (a simple type)

23. Constrain orientation >< datum modifier
DatumType/**ConstrainOrientation** optional Boolean *element*

24. Constrain subsequent <> datum modifier
DatumType/**ConstrainSubsequent** optional Boolean *element*

25. Degrees of freedom modifiers (as choice between ISO Tx,Ty,Tz,Rx,Ry,Rz and ANSI u,v,w,x,y,z)
DegreesOfFeedomType/**ISODegreesOfFreedom** *element*(s), the type of which is ***ISODegreeOfFreedomEnumType*** (a simple type)

26. ISO Thread specifications
ThreadSeriesEnumType various
ThreadClassEnumType various

27. Zone direction for circular runout (via DMIS support of ISO)
CircularRunoutCharacteristicNominalType/**ZoneDirection** optional *element*, the type of which is ***UnitVectorType*** (a complex type)

Annex F – DMSC Volunteer Agreement

(informative)

DMSC volunteer agreement: “You hereby agree, by your participation in any activity of this standards committee (including committee meeting attendance, email exchanges, phone conversations, or document generation), that you will not disclose any corporate confidential information or corporate trade secrets either verbally or in writing. Furthermore, any information disclosed to you in any activity of this standards committee, or disclosed to you in documents produced by this committee, will be provided to you for the sole purpose of establishing an industry-wide standard pursuant to the procedures prescribed by ANSI and ISO. You therefore agree not to use this information, or to collaborate in its use, in any manner that might suggest you have any proprietary rights to such information, such as rights to a patent, trademark, or copyright.”

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.

[3] ASME B89.7.2 – 1999, *Dimensional Measurement Planning*

