

## Quality Markup Language (QML)

Quality Markup Language (QML) beschreibt Qualitätsdaten in strukturierter Form. QML basiert auf der XML-Technologie und ist konform mit dem durch das World Wide Web Consortium (W3C) definierten Standard. Q-DAS wird neben dem bekannten ASCII Transferformat mit QML ein Format zur XML-orientierten Darstellung von Daten unterstützen.

Das Format selbst ist definiert durch ein QML-Schema (XML-Schema-Datei), das die Elemente beschreibt, die in einer gültigen Datei mit Daten im QML-Format vorkommen dürfen. Durch einen Validierungslauf der Daten-Datei gegen die Schemadatei ist es daher möglich, Fehler in den Transferdaten sehr früh zu erkennen und zu lokalisieren.

QML Version	Dateien zum Download	
	Schema	Beispiel
Version 1.0.0	<a href="#">QML-V 1 0 0.xsd</a>	<a href="#">QML-V 1 0 0-qdastest 2-V 0 3.xml</a> <a href="#">QML-V 1 0 0-qdastest 2-V 0 4.xml</a>
Version 1.1.0	<a href="#">QML-V 1 1 0.xsd</a>	<a href="#">QML-V 1 1 0-TestAll SAX-V 1 0.qml</a> <a href="#">QML-V 1 1 0-PosTol SAX-V 1 0.qml</a> <a href="#">QML-V 1 1 0-Values SAX-V 1 0.qml</a>
Ab Version ME 7		
Version 1.2.1	<a href="#">QML-V 1 2 1.xsd</a> <a href="#">RepType-V 1 2 1.xsd</a> <a href="#">XML-Rep-V 1 2 1.xsd</a>  <a href="#">QueryFilter-V 1 2 1.xsd</a>	<a href="#">QML-V 1 2 1-TestAll SAX-V 0 1.qml</a> <a href="#">QML-V 1 2 1-PosFord SAX-V 0 1.qml</a>  <a href="#">QML-V 1 2 1-FSK SAX-V 0 1.qml</a>

## QML Quality Markup Language - Qualität lernt sprechen?

*Was verbirgt sich hinter der QualityMarkupLanguage (kurz QML)? Worin liegen die Unterschiede zum Q-DAS ASCII Transferformat? Der folgende Artikel will Einblicke in QML, die Weiterentwicklung des Datenformats aus dem Hause Q-DAS, geben.*

Bereits die erste Version von qs-STAT unterstützte ein Format zum Datenaustausch. Dieses wurde mit dem Programm kontinuierlich weiterentwickelt. Formate wie das DFT/DFI-Format oder das DFD/DFQ-Format, besser bekannt als Q-DAS ASCII Transferformat, wurden implementiert. Letzteres kann mittlerweile als

Industriestandard angesehen werden. Dass mehrere Großkonzerne von Lieferanten von Messsystemen und Herstellern von SPC-Systemen die Einhaltung des Q-DAS ASCII Transferformates fordern, ist Beleg für die große Akzeptanz des Formates. Beide sind ASCII-Formate und wurden von Q-DAS stets offen zugänglich angeboten. Und genau in diese Tradition reiht sich auch das QML-Format ein.

An dieser Stelle sei ausdrücklich betont, dass das QML-Format nicht als Ablösung für das Q-DAS ASCII Transferformat entwickelt wurde, sondern als sinnvolle Ergänzung, um eine größere Flexibilität bezüglich Datentransfer und Aufbereitung bieten zu können. QML ist die Umsetzung des Datenformates für den Einsatz im XML-Umfeld. Q-DAS bietet somit ein Format, das den Bereich der Internet-Entwicklung optimal unterstützt.

Die Vorteile von QML liegen in dem strukturierten Aufbau und der Möglichkeit, durch die XML-Technologie die Daten schon beim Einlesen auf Konsistenz zu prüfen.

Die Entwicklung eines Transferformates auf XML-Basis wurde unter anderem durch ein Forschungsprojekt der Firma DaimlerChrysler vorangetrieben, alle Qualitätsdaten in einem universellen Format auszutauschen.

Im Laufe dieses Jahres wurde bei Q-DAS ein für alle offen zugänglicher QML-Standard definiert. Die erste Version des QML-Schemas wurde im Sommer 2003 auf einer eigens eingerichteten Homepage [1] veröffentlicht. Die Version qs-STAT ME 3.0 zur Control 2004 wird die erste Programmversion sein, die neben dem Q-DAS ASCII Transferformat auch das QML-Format lesen und speichern kann.

Für den an der verwendeten Technik interessierten Leser wird in den folgenden Abschnitten ein kleiner Einblick in XML (ExtensibleMarkupLanguage) gegeben.

**QualityMarkupLanguage (QML)** basiert auf XML, das ein Standard zum strukturierten Darstellen von Daten ist. Das Format ist universell und wurde von einem unabhängigen Konsortium, dem World Wide Web Consortium (kurz W3C [2]) in seiner Entwicklung vorangetrieben. XML hat äußerlich sehr große Ähnlichkeit mit HTML.

Mit XML können Informationen, die der Mensch rein aus der Struktur ableiten kann, maschinen-unabhängig eingelesen und verarbeitet werden. Als Beispiele seien hier die Gliederung eines Textes oder das Darstellen einer Hierarchie in einer Baumstruktur genannt. Was für den Menschen leicht ersichtlich ist, muss dem Computer auf eine andere Art beigebracht werden. Dazu ist XML das geeignete Werkzeug.

Die Struktur einer XML-Datei wird durch die strikte Verwendung von Start- und Endekennzeichnungen, den sogenannten Tags (engl. tag = Markierung) definiert.

Eine XML-Datei ist nur dann gültig und verarbeitbar, wenn zu jedem Start- auch ein Endetag existiert; die Datei wird dann als wohlgeformt bezeichnet. Innerhalb eines Elementes können beliebig viele weitere Elemente aufgelistet werden, wodurch die hierarchische Baumstruktur aufgebaut wird. Es gibt zwei Arten, die eigentliche Information in XML zu übermitteln:

- Als **Text** zwischen dem Start- und dem Endetag des Elementes (im Beispiel: Info1):

z.B.: <Info1>**hier wird die Information ausgetauscht**</Info1>

- Als Wert eines *Attributes* zu einem Element (im Beispiel: Info2):

z.B.: <Info2 *information*=**hier wird die Information ausgetauscht**/>

Das Attribut und sein Wert stehen dabei direkt im Starttag, der wie in obigem Beispiel (Info2) auch direkt abgeschlossen werden kann, wenn keine Textinformation vorhanden ist ("/" steht an Stelle des Endetags). Einem Element können beliebig viele Attribute zugeordnet werden. Beide Informationsquellen können gleichermaßen gut ausgewertet werden.

Da die Teile- und Merkmalsdaten aus qs-STAT mit einer Baumstruktur beschrieben werden können, können diese mit XML sehr gut abgebildet werden. Das folgende Beispiel von Positionstoleranzen zeigt, wie mit QML die verschachtelte Struktur von Teilen und Merkmalen abgebildet wird.

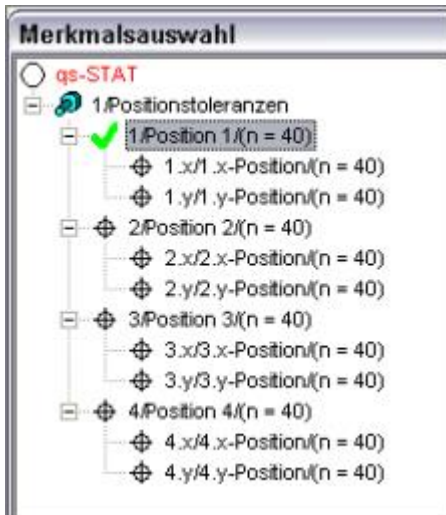


Abbildung 2: Teile-/Merkmalsliste aus qs-STAT®

Das Beispiel zeigt den hierarchischen Aufbau von einem Teil und seinen untergeordneten Merkmalen. Das Element Parts bildet sozusagen die äußere Klammer um die Teile-/Merkmalsliste. Darunter hängt ein Teil (Part; in diesem Fall als "Positionstoleranzen" bezeichnet), dem eine Positionstoleranz (PositionalTolerance; "Position 1") mit zwei Positionen (Position) als Merkmale untergeordnet sind. Diese Struktur wird alleine durch die Verschachtelung der QML-Elemente (Parts, Part, Positional Tolerance und Position) vorgegeben.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--QML Data to QML-Schema Version 1.0.2 developed by Q-DAS-->
<QML xmlns="QML" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="QML QML-V_1_0_2.xsd" qmlVersion="1.0.2" k0100="11"
qmlSchemaLocation="QML-V_1_0_2.xsd">
  <Parts>
    <Part k1001="1" k1002="Positionstoleranzen">
      <PositionalTolerance k2001="1" k2002="Position 1">
        <Position k2001="1.x" k2002="1.x-Position"/>
        <Position k2001="1.y" k2002="1.y-Position"/>
      </PositionalTolerance>
      <PositionalTolerance k2001="2" k2002="Position 2">
        <Position k2001="2.x" k2002="2.x-Position"/>
        <Position k2001="2.y" k2002="2.y-Position"/>
      </PositionalTolerance>
      <PositionalTolerance k2001="3" k2002="Position 3">
        <Position k2001="3.x" k2002="3.x-Position"/>
        <Position k2001="3.y" k2002="3.y-Position"/>
      </PositionalTolerance>
      <PositionalTolerance k2001="4" k2002="Position 4">
        <Position k2001="4.x" k2002="4.x-Position"/>
        <Position k2001="4.y" k2002="4.y-Position"/>
      </PositionalTolerance>
    </Part>
  </Parts>
</QML>
```

Abbildung 3: Auszug aus einer QML-Datei

Wie aus dem Beispiel zu erkennen, sind die Namen der einzelnen QML-Elemente so gewählt, dass der Anwender aus ihnen schon Rückschlüsse auf den Informationsinhalt machen kann. Die Information wird in QML meistens in Attribute und deren Werte (siehe z.B. k2001 und k2002 in Abbildung 3) verpackt. Die Attributnamen entsprechen den K-Feldern aus dem Q-DAS® ASCII Transferformat.

XML bietet die Möglichkeit, die innerhalb einer Datei erlaubten Elemente und Attribute vorzugeben. Dies geschieht in einer sogenannten Schema-Datei (Dateien mit der Endung \*.xsd).

### **XML-Schema**

Mit einem XML-Schema wird die Menge von XML-Elementen definiert, die in einer XML-Datei verwendet werden dürfen. Die Schemadatei ist selbst eine XML Datei, deren Schema vom W3C vorgegeben wurde.

Einer XML-Datei muss das verwendete Schema sozusagen als "Bauvorschrift" zugewiesen werden. Dies geschieht im Kopf der XML-Datei.

Das Verwenden von XML-Schemata ermöglicht den nächsten Vorteil der XML-Technologie: Das Validieren. Darunter versteht man die Prüfung der Inhalte und der Struktur einer XML-Datei beim Einlesen bezüglich den in der Schemadatei definierten Vorgaben. So können Fehler in den Transferdaten sehr schnell und komfortabel erkannt werden.

Zum Erstellen einer Schemadatei stehen viele Tools zur Verfügung. Folgende Abbildung zeigt die grafische Darstellung des in dem QML-Schema definierten Elementes Part und seiner möglichen Kindelemente.

In Abbildung 4 sieht man passend zu dem obigen Beispiel die Definition des Teiles und der Positionstoleranz. Das Diagramm ist so zu lesen, dass alle Elemente, die in der Grafik rechts von Part aufgelistet sind, auch als Elemente innerhalb des QML-Elementes vorkommen dürfen. Dies sind die in qs-STAT vorkommenden Merkmalstypen (VariableCharacteristic, Positional Tolerance usw.) sowie das Element Part selbst, um eine Teilehierarchie zu ermöglichen. Unterhalb der Positionstoleranz werden die einzelnen Positionen und/oder Messwerte (Value) des Elementes Positionstoleranz aufgeführt. Die Messwerte der einzelnen Positionen findet man entsprechend einer Ebene unterhalb der Position (in der Abbildung nicht mehr dargestellt).

## **IHR KONTAKT**

### **SUPPORT HOTLINE**

hotline.qdas.mi@hexagon.com

T: +49 6201 3941-214