



CIQ TC Specifications

Customer Information Quality Technical Committee

Release Notes

Version 3.0 (draft)

Date Created: 14 July 2005

Last Updated: 15 December 2005

Editors

Ram Kumar, Individual Member and Chair, OASIS CIQ TC

Max Voskob, Individual Member, OASIS CIQ TC

Contributors

John Glaubitz	Vertex	Member, CIQ TC
Hido Hasimbegovic	Individual	Member, CIQT TC
Robert James	Individual	Member, CIQ TC
Joe Lubenow	Individual	Member, CIQ TC
Mark Meadows	Microsoft Corporation	Member, CIQ TC
John Putman	Individual	Prospective Member, CIQ TC
Michael Roytman	Vertex	Member, CIQ TC
Colin Wallis	New Zealand Government	Member, CIQ TC
David Webber	Individual	Member, CIQ TC

Abstract

This document provides an overview of the differences between version 2.0 and version 3.0 of the CIQ TC Specifications

Intellectual Property Rights, Patents, Licenses and Royalties

CIQ TC Specifications (includes documents, schemas and examples^{1 and 2}) are free of any Intellectual Property Rights, Patents, Licenses or Royalties. Public is free to download and implement the specifications free of charge. Please, read OASIS Copyright Notice in APPENDIX A.

¹xAL-Australia.XML

Address examples come from AS/NZ 4819:2003 standard of Standards Australia and are subject to copyright

²xAL-international.xml

Address examples come from a variety of sources including Universal Postal Union (UPU) website and the UPU address examples are subject to copyright.

Table of Contents

1	INTRODUCTION	4
1.1	STATUS	4
2	DIFFERENCES BETWEEN VERSION 2.0 AND 3.0.....	5
2.1	NEW DESIGN APPROACH	5
2.2	DTD SUPPORT	5
2.3	BACKWARD COMPATIBILITY	5
2.4	NAMES FOR SPECIFICATIONS.....	5
2.5	UML MODEL.....	6
2.6	SCHEMA STRUCTURE	6
2.7	DATA TYPES	7
2.8	NAMESPACE.....	7
2.9	ELEMENTS AND ATTRIBUTES	7
2.10	PRESERVATION OF THE ORIGINAL ORDER.....	8
2.11	XLINK TO DEFINE RELATIONSHIPS.....	8
2.12	ENUMERATIONS SUPPORT.....	8
2.13	DATA QUALITY METRICS	9
2.14	EXAMPLES	9
	APPENDIX A. NOTICES	10

1 Introduction

This document provides the difference between version 2.0 and version 3.0 of OASIS CIQ TC specifications family namely:

- **xNL** :extensible Name Language
- **xAL**:extensible Address Language
- **xNAL**:extensible Name and Address Language (combines xNL and xAL)
- **xPIL**:extensible Party Information Language (formerly known as extensible Customer Information language (**xCIL**))
- **xPRL**:extensible Party Relationships Language (formerly known as extensible Customer Relationships Language (**xPRL**)).

The purpose of this document also is to give readers a quick snapshot of the differences between the two versions of the CIQ TC specifications and help plan them to migrate from version 2.0 to version 3.0.

1.1 Status

This document is currently a draft version and will be updated periodically on no particular schedule. Send comments to the editor.

Committee members should send comments on this specification to the ciq@lists.oasis-open.org list. Others should subscribe to and send comments to the ciq-comment@lists.oasis-open.org list. To subscribe, send an email message to ciq-comment-request@lists.oasis-open.org with the word "subscribe" as the body of the message.

General public may also use "Send comment" option on OASIS CIQ TC home page to submit any feedback.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the CIQ TC web page (<http://www.oasis-open.org/committees/ciq/>).

The errata page for this specification is at <http://www.oasis-open.org/committees/xxx/yyy>.

2 Differences between version 2.0 and 3.0

In this section, we summarize the key differences between version 2.0 and version 3.0 of the CIQ TC specifications. The following documents should be read in conjunction with this document to get a detailed understanding of the differences:

- CIQ TC – Technical Overview V3.0
- CIQ TC – Name, Address and Party Specifications V3.0
- CIQ TC – Party Relationships Specification V3.0 – release date not set yet
- W3C xLink Recommendation (<http://www.w3.org/1999/xlink>)

2.1 New design approach

Version 3.0 has taken a new design approach to modeling CIQ Specifications. Please study the Technical Overview document or the CIQ XML Schemas for more information. The new approach is summarized as follows:

- Flat data structure design as opposed to the deeply nested hierarchical data structure in V2.0
- Generic element and attributes reused throughout the schema, but at the same time maintaining the semantic integrity
- Importance of semantics by enabling users to define the semantics of data for a particular context (e.g. cultural requirements) rather than forcing on the users
- Enumerations to support extensibility of the schema and to support customization of the schema to meet application requirements without impacting the structure of the schema
- Strong focus on namespace support for extensibility in a controlled manner
- Flexibility to meet cultural specific data semantic requirements
- Design approach based on implementation requirements and to ease implementation by programmers/developers
- Data Modeling approach
- Practical Requirements based approach.

2.2 DTD Support

Version 3.0 does not support any DTDs.

2.3 Backward compatibility

Version 3.0 is NOT backward compatible with Version 2.0. However, version 3 can represent any party related information that version 2 can, but in a simpler and more elegant way easing the implementation tasks.

2.4 Names for specifications

Term “PARTY” has a broader definition than term “CUSTOMER”. Customer is a subset of party. Therefore, the name “extensible Customer Information Language (xCIL)” is now replaced with the name “extensible Party Information Language (xPIL)” in version 3.0 and “extensible Customer Relationships Language (xCRL)” is now replaced with the name “extensible Party Relationships Language (xPRL)” in version 3.0.

2.5 UML Model

Version 3.0 provides high level UML models of the specifications that reflect the XML schemas that support the specifications.

2.6 Schema structure

There are different ways to model data, including hierarchical, relational and object-oriented. Address data for example, is hierarchical in nature (Example: a country has cities, a city has streets and streets have premises, premises have sub premises etc). So a hierarchical model was used to design the data model in Version 2.0.

However, due to the deep nested structure of the schemas, building object models from the schemas have been proven to be complex to implement and expensive. Therefore, version 3.0 design uses a relational approach by defining a flat schema structure. An example that differentiates the two approaches is shown below. Let us consider the following address example:

Example

Egis Building, Level 12, 67 Albert Avenue,
Chatswood, NSW 2067, Australia

Representation in xAL Version 2.0

```
<AddressDetails>
  <Country>
    <CountryName>Australia</CountryName>
    <AdministrativeArea>
      <AdministrativeAreaName>NSW</AdministrativeAreaName>
      <Locality>
        <LocalityName>Chatswood</LocalityName>
        <Thoroughfare Type="Street">
          <ThoroughfareNumber>67</ThoroughfareNumber>
          <ThoroughfareName>Archer Street</ThoroughfareName>
          <Premise Type="Building">
            <BuildingName>Egis</BuildingName>
            <SubPremise Type="LEVEL">
              <SubPremiseNumber>12</SubPremiseNumber>
            </SubPremise>
          </Premise>
        </Thoroughfare>
      <PostalCode>
        <PostalCodeNumber>2067</PostalCodeNumber>
      </PostalCode>
    </Locality>
  </AdministrativeArea>
</Country>
</AddressDetails>
```

Representation in xAL Version 3.0

```

<a:Address>
  <a:Country>
    <a:Name>Australia</a:Name>
  </a:Country>
  <a:AdministrativeArea>
    <a:Name>NSW</a:Name>
  </a:AdministrativeArea>
  <a:Locality>
    <a:Name>Chatswood</a:Name>
  </a:Locality>
  <a:Thoroughfare>
    <a:Name>Albert Avenue</a:Name>
    <a:Number>67</a:Number>
  </a:Thoroughfare>
  <a:Premesis>
    <a:NameElement a:NameType="BuildingName">Egis Building</a:NameElement>
    <a:NameElement a:NameType="Location">Level 12</a:NameElement>
  </a:Premesis>
  <a:PostalCode>
    <a:Identifier>2067</a:Identifier>
  </a:PostalCode>
</a:Address>

```

2.7 Data types

Version 2.0 does not specify strong data types for text nodes and attribute values. All elements and attributes in version 3.0 are strongly data typed.

2.8 Namespace

Version 3 allows for attributes from other namespaces to reside under any element, but disallows elements from other namespaces as in the following example.

```

<a:Contacts xmlns:a="urn:acme.org:corporate:contacts" xmlns:b="urn:acme.org:corporate:IDs"
xmlns:xnl="urn:oasis:names:tc:ciq:xnl:3">
  <xnl:PartyName b:CustomerID="123445" xnl:DataQuality="Valid">
    <xnl:PersonName>
      <xnl:NameElement>John Johnson</xnl:NameElement>
    </xnl:PersonName>
  </xnl:PartyName>
  <xnl:PartyName b:SupplierID="43589304" b:CustomerID="83453485">
    <xnl:OrganisationName>
      <xnl:NameElement>Universal Stuff Ltd.</xnl:NameElement>
    </xnl:OrganisationName>
  </xnl:PartyName>
</a:Contacts>

```

All elements in the CIQ Specifications are extensible allowing for any number of attributes from a non-target namespace to be added.

All elements share the same declaration:

```

<xs:anyAttribute namespace="##other" processContents="lax"/>

```

This specification mandates that an application should not fail if it encounters an attribute from a non-target namespace. The application may choose to ignore or remove the attribute.

2.9 Elements and attributes

With version 2.0, there was always ambiguity in placing elements and attributes because of the ability to use same elements and attributes in various places of the schema structure and xAL is a classical example of this. For example, "ThoroughfareName", could be used under "Country", "AdministrativeArea", "SunAdministrativeArea", "Locality", "SubLocality", "Thoroughfare", or "SubThoroughfare" structures in the schema due to the hierarchical nature of address structures.

This ambiguity has been avoided in version 3.0 by flattening the structure.

Release Notes

Locally declared elements and attributes in version 3.0 do not have parent's name as part of its own name as it was the case in version 2.0.

2.10 Preservation of the original order

Order of name or address elements occurring in the original data should be preserved for correct presentation.

If an application needs to present the name to a user it may not always be aware about the correct order of the elements if the semantics of the name elements are not available. Version 3.0 supports the order of presentation in xNL and xAL.

Example – normal order

```
Mr Jeremy Apatuta Johnson PhD
```

could be presented as follows in version 3.0

```
<n:PartyName>
  <n:PersonName>
    <n:NameElement>Mr</n:NameElement>
    <n:NameElement>Jeremy</n:NameElement>
    <n:NameElement>Apatuta</n:NameElement>
    <n:NameElement>Johnson</n:NameElement>
    <n:NameElement>PhD</n:NameElement>
  </n:PersonName>
</n:PartyName>
```

and restored back to *Mr Jeremy Apatuta Johnson PhD* during data formatting exercise.

Any other order of *NameElement* tags in the XML fragment could lead to an incorrect presentation of the name.

2.11 xLink to define relationships

xLink provides a set of attributes that can be reused within other namespaces. The meaning and usage of those attributes are well defined in xLink specification from W3C. Version 3.0 of CIQ TC specifications uses xLink to define relationships between two parties. By incorporating xLink, the CIQ TC specifications have been significantly simplified for defining relationships.

2.12 Enumerations Support

Version 3.0 supports for extensive use of enumerations for flexibility in the way the schemas can be used. This enables adopters of the schemas to adjust the schemas for their specific needs without affecting the actual structure. Let us consider the following example:

Example

```
Professor Dr. Paruvachi Ammasai Venkatachalam PhD
```

The above name is of Indian origin. This name in Anglo Saxon culture is represented as follows:

```
<n:PartyName>
  <n:PersonName>
    <n:NameElement Type="PrecedingTitle">Professor</n:NameElement>
    <n:NameElement Type="Title">Dr</n:NameElement>
    <n:NameElement Type="GivenName">Paruvachi</n:NameElement>
    <n:NameElement Type="MiddleName">Ammasai</n:NameElement>
    <n:NameElement Type="FamilyName">Venkatachalam</n:NameElement>
    <n:NameElement Type="Suffix">PhD</n:NameElement>
  </n:PersonName>
</n:PartyName>
```

Release Notes

Version 3.0 provides the ability for the implementers to define the type of names as enumerations and in the above example, the enumeration values for *NameElement* are: *PrecedingTitle*, *Title*, *GivenName*, *MiddleName* and *FamilyName*. These values are provided with the schema.

If the above name had to be represented in its native Indian culture (applicable to southern part of India only), it would like the following example:

```
<n:PartyName>
  <n:PersonName>
    <n:NameElement Type="PositionTitle">Professor</n:NameElement>
    <n:NameElement Type="EducationTitle">Dr</n:NameElement>
    <n:NameElement Type="NativePlaceName">Paruvachi</n:NameElement>
    <n:NameElement Type="FatherName">Ammasai</n:NameElement>
    <n:NameElement Type="ActualName">Venkatachalam</n:NameElement>
    <n:NameElement Type="Degree">PhD</n:NameElement>
  </n:PersonName>
</n:PartyName>
```

The implementers can add the enumeration list (in this case, *PositionTitle*, *EducationTitle*, *NativePlaceName*, *FatherName*, *ActualName*, and *Degree*) without impacting the structure of the schema.

However, it is important that the enumeration list has to be agreed by the parties involved for achieve true interoperability.

2.13 Data Quality Metrics

One of the key aims of the CIQ (Customer **Information Quality**) TC is to enable representation and exchange of quality party information. The CIQ TC is of the strong view that data quality plays a significant role in interoperability.

We at OASIS CIQ TC strongly believe in the following formula:

Data Interoperability = *Data Integration* + *Data Quality* + *Data Standards* + *Data Semantics*

All components on the right hand side of the above formula are important for successful data interoperability.

For the first time since the CIQ TC's inception in 2000, version 3.0 of the CIQ TC specifications have concentrated on introducing simple data quality metrics to the data it represents. The specifications allows for data quality information to be provided as part of the entity using attribute *DataQuality* that can be set to either "Valid" or "Invalid", if such status is known. If *DataQuality* attribute is omitted it is presumed that the validity of the data is unknown.

DataQuality attribute refers to the content of a container, e.g. *PersonName*, asserting that all the values are known to be true and correct. This specification has no provision for partial data quality where some parts of the content are correct and some are not or unknown.

Example – data quality

```
<n:PersonName n:DataQuality="Valid">
  <n:NameElement>John Anthony Jackson</n:NameElement>
</n:PersonName>
```

In this example *John Anthony Jackson* is known to be the true and correct value asserted by the sender of this data.

This feature allows the recipient of data to get an understanding of the quality of data they are receiving and assists them to take appropriate measures to handle the data according to its quality.

2.14 Examples

Version 3.0 provides numerous international address examples (covering most of the countries) than version 2.0 in "xal-international.xml" file.

Appendix A. Notices

Copyright © OASIS Open 2005. All Rights Reserved.

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full Policy may be found at the OASIS website.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.