Extensible Name and Address Language (xNAL) Standard Description Document for W3C DTD/Schema

Version 2.0

(Approved Committee Specification)

A Standard from the Customer Information Quality Technical Committee

CHANGE HISTORY

Status	Version	Date	Author	Summary of Changes
Draft	1.0	08 May 2001	CIQ-TC	Initial Draft
Draft	1.1	01 December 2001	CIQ TC	Changes to the draft
Draft	2.0	31 May 2002	CIQ TC	New release of xNAL DTD/Schema that supports namespaces and new structures
Final	2.0	24 July 2002	CIQ TC	Final version of 2.0 specifications after changes and review

OASIS COPYRIGHT NOTICE

TABLE OF CONTENTS

1.0 ACKNOWLEDGEMENTS 5

2.0 INTODUCTION 7

2.1 PROBLEM AND OBJECTIVES 8
2.2 THE DESIGN OF A NAME AND ADDRESS STANDARD 8
2.3 XML IMPLEMENTATION 9
2.4 XML TAGGING CONVENTIONS 10
2.4.1 3.3.2. Guidelines for Tag Naming Conventions 10
2.5 EXTENSIBLE NAME AND ADDRESS LANGUAGE 11

3.0 THE OBJECTIVE AND SCOPE 12

3.1 THE CHALLENGE 12
3.2 WHAT DOES XNAL NOT REPRESENT 12

4.0 USING THE XNAL DTD/SCHEMA 13

4.1 PURPOSE OF THE XML DTD/SCHEMA FOR NAME AND ADDRESS 13
4.2 FLEXIBILITY 13
4.2.1 Simple Representation 13
4.2.2 Detailed Representation 14
4.3 DON'T GET CONFUSED – KEEP IT SIMPLE 15
4.4 NAMESPACES AND VERSIONS 15
4.5 XML SCHEMA: EXTENSIBILITY 15
4.6 XML SCHEMA: DOCUMENT FRAGMENTS 15
4.7 DEEP NESTING VS. FLAT STRUCTURE 15
4.8 WHERE TO START 16
4.9 COMPATIBILITY BETWEEN DTD AND SCHEMA 16
4.10 DOCUMENT EXCHANGE BETWEEN DIFFERENT PARTIES 16

5.0 OTHER NAME AND ADDRESS STANDARD INITIATIVES 17

5.1 HOW DOES XNAL DIFFER FROM THE OTHER STANDARDISATION EFFORTS 17

6.0 XNAL GRAMMAR 20

7.0 EXAMPLES 22

7.1 EXAMPLE 1 22
7.2 EXAMPLE 2 23
7.3 EXAMPLE 3 24
7.4 EXAMPLE 4 25
7.5 EXAMPLE 5 26
7.6 EXAMPLE 6 27
7.7 EXAMPLE 7 28
7.8 EXAMPLE 8 29

8.0 REFERENCES 31

1.0 Acknowledgements

OASIS and the CIQ Technical Committee (TC) wishes to acknowledge MSI Business Solutions Pty. Ltd, Australia (formerly known as MasterSoft International Pty. Ltd) for initiating this standards work to OASIS by submitting its XML standards for Customer Information Management called Customer Identity Markup Language (CIML) and the XML standards for name and address data management called Name and Address Markup Language (NAML) in March 2000. Ram Kumar (rkumar@msi.com.au), Chairman of the Customer Information Quality (CIQ) TC of OASIS and the Chief Technologist and Architect of MasterSoft played the key role in setting up the Technical Committee. Ram is the author of the two standards (NAML and CIML) developed by MSI. NAML was based on the Universal Name and Address (UNA) format of MSI.

OASIS and the CIQ TC wishes to acknowledge AND Solutions, Inc, Netherlands, for submitting its Global Address Standards to OASIS in October 2000 to be included as part of the address standards (xAL) effort. Mr. Vincent Buller, former Co-Chair of the Customer Information Quality TC of OASIS and former Senior Consultant of AND Solutions has played a significant role along with Ram Kumar in setting up the technical committee and has contributed to the development of xAL Standard.

The CIQ TC thanks Mr. Graham Rhind (graham@grcdi.nl), Consultant, International Address Databases, for his feedback/input on xNL and xAL standards and for his permission to use some of his materials on addresses.

The CIQ TC thanks Mr. Holger Wandt, chairman of the working group address databases within CEN/TC331/WG3 (CEN is the European Standardization body) for giving permission to use address examples from his committee's specifications.

OASIS wishes to acknowledge the contributions of the members of the CIQ TC to this standards work. The following individuals were members of the committee during the development of this specification:

Name	Organisation
Mr. Ram Kumar (Chairman)	MSI Business Solutions, Australia
Mr. David RR Webber	XML Global Technologies Inc, USA
Mr. John Bennett	Parlo.com, USA
Mr. Joe Lubenow	Individual member of OASIS, (Lubenow and Associates), USA
Mr. Nikolaj Nyholm	Ascio Technologies, Norway
Mr. Marcus Goncalves	Individual member of OASIS, (iCloud, Inc), USA
Mr. Mark Meadows	Microsoft, USA
Mr. Robert James	Individual member of OASIS, (Partner White Wulf Consultants), U.K
Mr. Max Voskob	MSI Business Solutions, New Zealand
Mr. John Glaubitz	Vertex, Inc, USA

The CIQ TC thanks all those who reviewed the specifications and provided feedback.

Last but not least, OASIS and the CIQ TC thanks all users of the CIQ TC standards in real world and for their continuous feedback and support.

2.0 Intoduction

Customer (Person or Organisation where, Organisation could be a company, association, club, University, etc) data consists of many components. However, a person or company's name and address is *the key* identifier of a "customer".

Name and address, as a data type, is very difficult to manage. This data is often volatile… customers come and go, addresses change, names change. This data is often cluttered when entered. Name and address fields on data entry screens are usually free format and ripe for users to enter comments without any edits. Name and address is subjective…it can be written in a number of different ways and still be the same. There is no application independent standard to represent name and address data and to measure its quality. This problem is further compounded by the different ethnic backgrounds of name and address data in a global market.

In the last millennium mailing and delivery to customers has become a vital link in the logistic chain between suppliers and customers. This link is troubled with an average of 15 % of individuals and businesses moving each year. Mailing and shipping to the wrong address is costing business a significant part of their revenue. Studies show that incorrect addresses can cost up to 8 % of a companies revenue, generated by double postage fees, extra printing and material cost, handling, and the related cost for organising and administering these erroneous deliveries.

Also call centres face challenges when registering correct addresses. Especially in an international environment where language differences can lead to misunderstandings and erroneous address input. In order to control and facilitate the data input tools for address data entry and address verification tools are traditionally used in environments where large databases with addresses exist. The addresses can generally be improved with standards in format, reference data and tools. Improvements are both in quality (correct address, customer friendly data intake) as in quantity (faster input or correction).

It is no surprise that vendors of ERP, CRM and Sales Force Automation packages incorporate the benefit of clean and structured address management tools and start integrating these tools in their own product propositions. Most recent on the Internet it is recognised that e- business is frustrated by the 5 to 9 % of the shipments that are being returned due to addressing errors. In a Forrester Research paper / survey of 1998 it is already described that the main barrier for implementing global e-Commerce is the shipping difficulties: determine whether the shipping and billing addresses are valid. For the on-line world the element of fraud has an even greater part in losing revenue.

Both on-line companies and the parcel delivery companies call for improvement of these parts or the chain. Problem that these www-companies face is a great variety in international addressing systems, and the lack of knowledge on the format, structure and data involved in a correct address.

Any larger international address database that has a certain dynamic (percentage of alterations) can benefit from address entry or address structuring and cleaning systems.

2.1 Problem and Objectives

Challenges in the treatment of name and address occur mostly during data entry. The order in which address elements are naturally provided varies from country to country. In some countries the house number is provided before the street name, in other countries the house number is given after the street name. For some countries the house number is essential to determine the postcode, for other countries a simple city input is sufficient. Correct entry of an address in an international environment becomes heavily dependent on the knowledge of the person performing the data entry, or the ability to interpret the appropriate address elements.

If an address database contains errors, for example the same address is entered in two different varieties (thus one could be erroneous) the retrieval of information becomes complicated. The fact that elements of a person are not unique any more can lead to unwanted duplication of information. Storing the same information in different ways makes de-duplication more difficult. Search and query functions into a database may give ambiguous or mixed up information, not presenting the desired matches and leading to long drop down lists of choices and a drop in the performance rate during retrieval. This does not improve the general communication or transfer of necessary data.

With a global world as the market a single system methodology is advisable for data entry, address data storage and address data transfer. Data and systems must support users in organising a company database that includes international addresses. What is wanted in these digital environments are support for the transfer of a (address) "record" from one database to another, and to support a transaction completely from begin to end.

There are, however, a number of name and address standards available throughout the world. To a large extent, these standards have been designed with a particular business requirement in mind, for example, the expedient delivery of a piece of mail. This has generally meant that while the particular standard is appropriate for the purpose for which it was designed, it is frequently not suitable for a variety of other purposes.

We strongly believe that this effort is only one of its kind in the world where the aim is to define a “global” name and address standard that is vendor neutral, open, and application independent. In this document, we will also list out the other name and address standardisation activities that are happening.

2.2 The design of a name and address standard

The design of a name and address standard using XML was carried out to address the problems mentioned in the previous sections and it is re-iterated here:

- Name and address is arguably the only type of business data that can be written in different ways and still be “correct’. Names can contain full given names or just initials. There are different ways of formatting the same address. Spelling errors are not uncommon.

- Without an external data format standard, database engineers have traditionally developed their own formats for name and address.

- It is difficult to move name and address data between systems, tools, products and platforms due to the many different formats available. The task of moving name and address data inevitably requires special programming for both the sender and receiver of that data.

The standard for name and address proposed in this document has been specifically designed to:

- Provide a uniform data model for name and address data
- Describe a standard method of storing and processing name and address data
- Provide a standard procedure for the interchange of name and address data between systems, tools, products and platforms
- Establish a standard method of describing relationships between name and address records in customer information files.

2.3 XML implementation

Data in an XML document is specified as either elements or as attributes of elements. What attributes an element has and which other elements it may contain is specified in a Document Type Definition (DTD) or Schema. This DTD/Schema is delivered as a separate file that has the filename extension ".dtd" or ".xsd". The data file that contains the addresses refers to this separate file.

The following rules were used to decide which data should be encoded as an XML element and which as an XML attribute:

1. All data (basic address elements, e.g., street number, premise name, post code, etc) that is displayed or printed (on an address label) should be encoded as an element (content) with the exception of separators/punctuation marks in the data (explained in point 3 below).
2. Data that will not be displayed, but instead tells what the meaning is of the element content should be encoded as element attributes. In practice, this means most of the "types", such as LocalityType, NameType, etc.
3. Data such as separators that may or may not be displayed shall be defined as attributes, e.g., "/" in 12/14, #in APT # 12, etc. This is important as the meaning of the separators such as punctuation marks in address data means different things for different countries. For example, a number range "12-14' in Australia this may indicate "12 to 14'. In The Netherlands, this is more likely to mean '12 Flat 14', and '12 to 14' is more likely to be written '12 t/m 14'. The punctuation mark is therefore essential to ensure deliverability. Because of these differences the punctuation marks are data (and cannot be made dependent on being output as a standard by an application) and they need to be stored.

2.4 XML Tagging Conventions

We have extracted the XML tagging guidelines from the Open Travel Association Group (OTA) and from the ebXML as the basis for tagging xAL definitions with some changes to them.

2.4.1 3.3.2. Guidelines for Tag Naming Conventions

A key part of the XML grammar is consistent naming conventions for tags that represent the infrastructure and business-related elements. Tag name writers MUST follow these rules unless business requirements require other naming conventions.

- Use mixed case tag names, with the leading character of each word in upper case and the remainder in lower case.
  Example: <PostalCode>

- Acronyms are discouraged, but where needed, use all upper case.
  Example: <UserID>

- Illegal characters cannot be used (e.g.: forward slash, etc.). Recommended characters in a tag name are basically limited to letters and underscores.
  Example: (not allowed) <Date/Time>

- The use of periods to indicate the version and hierarchy is discouraged.

Tag writers SHOULD use these guidelines when constructing tag names.

- Use the same tag names with elements in a similar child structure
  Example: <ContactAddress>
  <HomeAddress>
  <WorkAddress>

- Use plural tag names only for collections.
  Example: <CreditCards>
  <CreditCard>

- Element and attribute name size have no limitation. The names must be meaningful.
  Example: <CustomerRelationshipInformation>

Element and attribute names should incorporate the proposed list of suffixes for tag names as recommended by ebXML. The ebXML Data Element Representation Classes are the following (includes ebXML definition):

Amount - A number of monetary units specified in a currency where the unit of currency is explicit or it may be implied.

Code - A character string that represents a member of a set of values.

Boolean - An enumerated list of two, and only two, values which indicates a Condition such as on/off; true/false etc. (It was the general consensus to use ‘Flag’ as a term to indicate a Boolean value.)

Date - A day within a particular calendar year. Note: Reference ISO 8601.

Time - The time within any day in public use locally, independent of a particular day. Reference ISO 8601:1988.

DateTime - A particular point in the progression of time. Note: This may incorporate dependent on the level of precision, the concept of date.

Identifier - (standard abbreviation Id, meaning a unique identifier) A character string used to identify and distinguish uniquely, one instance of an object within an identification scheme.

Name - A word or phrase that constitutes the distinctive designation of a person, object, place, event, concept etc.

Quantity - A number of non-monetary units. It is normally associated with a unit of measure.

Number - A numeric value that is often used to imply a sequence or a member of a series.

Rate - A ratio of two measures.

Text - A character string generally in the form of words.

Measure - A numeric value that is always associated with a unit of measure.

2.5 extensible Name and Address Language

With the advent of XML as a defacto standard for representing data, OASIS has developed an application independent XML standard for name and address data management extensible Name and Address Language (xNAL). xNAL does not include all the address components throughout the world. But that is where the power of XML comes into play. It is extensively scalable and extendable allowing xNAL to evolve as more additional components are identified.

xNAL is broken into two components namely,

- xNL : extensible Name Language to describe name components, and
- xAL : extensible Address Language to describe address components.

This has been done for maintainability of the DTDs/Schemas.

The objective of xNAL is to describe a common structure for Personal/Organization Names and Addresses that would enable any applications that want to represent customer names and addresses in a common standard format. The applications could be CRM/e-CRM, Customer Information Systems, Data Quality (Parsing, Matching, Validation, Verification, etc), Customer Data Warehouses, Postal services, etc.

However, any party for its own purposes and applications may use xNAL grammar or parts of it.

3.0 The Objective and Scope

The objective of this document is to describe the extensible Name and Address Language (xNAL) W3C DTD/Schema in detail with examples.

This document provides a set of simple guidelines to help using xNAL and exchange information between different parties with minimum misinterpretation and misuse of the structures.

It is important to read the following two documents in order to understand and use xNAL:

- xNL Specifications Document for W3C DTD/Schema
- xAL Specifications Document for W3C DTD/Schema.

3.1 The Challenge

The challenge for xNAL is to provide the ability to handle the following:
- About 36+ customer name formats
- Addresses of 241+ Countries
- With about 130+ Address Formats
- Represented in 5,000+ languages (dialects)
- Should be application independent, open and vendor neutral.

3.2 What does xNAL not represent

xNAL only defines the XML vocabulary to represent customer names.

xNAL does not:
- define vocabulary for security of the data represented in xNAL format
- define vocabulary for transportation of the data represented in xNAL format
- define vocabulary for messages associated with the data represented in xNAL format
- define vocabulary for privacy and permissioning of the data represented in xNAL format
- validate/verify the actual data represented in xNAL format
- format names.

4.0 Using the xNAL DTD/Schema

4.1 Purpose of the XML DTD/Schema for name and address

The XML DTD/Schema for names has been designed to be truly global and application independent and therefore, is designed to be flexible to handle name structures of different applications. For example, from a simple user registration system that uses simple name and address elements (Example: Title, First Name, Middle Name and Last Name, address lines) to a name and address validation system that needs all the elements of a name and address, can be defined using this name and address schema.

4.2 Flexibility

There is no necessity to define a name using all the possible tags and therefore, make the definition complex. Flexibility is provided to define a name with the tags that are necessary and are meaningful to the user.

For example, let us see how we can use xNAL to define the following details at simple level or a detailed level as xNAL provides this flexibility:

Mr.Ram V. Kumar
C/O Privacy Link
PO Box: 773, Chatswood, NSW 2057, Australia

4.2.1 Simple Representation

```
<xNAL>
   <Record>
      <xNL>
          <NameDetails PartyType="Person">
            <NameLine>Ram V Kumar</NameLine>
            <DependencyName PartyType="Organisation"
                            DependencyType="C/O">
         <NameLine>PrivacyLink</NameLine>
        </DependencyName>
         </NameDetails>
      </xNL>
      <xAL>
      <AddressDetails>
         <Address>POBox: 773, Chatswood,NSW 2057, Australia</Address>
      </AddressDetails>
     </xAL>
    </Record>
</xNAL>
```

4.2.2 Detailed Representation

```
<xNAL>
   <Record>
      <xNL>
         <NameDetails PartyType="Person">
            <PersonName>
               <Title>Mr</Title>
               <FirstName NameType="GivenName">Ram</FirstName>
               <MiddleName Type="Initial">V</MiddleName>
               <LastName NameType="SurName">Kumar</LastName>
            </PersonName>
            <DependencyName PartyType="Organisation"
                              DependencyType="C/O">
               <OrganisationNameDetails>
                  <NameLine>PrivacyLink</NameLine>
               </OrganisationNameDetails>
            </DependencyName>
         </NameDetails>
      </xNL>



      <xAL>
         <AddressDetails AddressType="Postal"
                           CurrentStatus="Investment"
                           ValidFromDate="1 Jan 2000"
                           ValidToDate="31 March 2000">
            <Country>
               <CountryName>Australia</CountryName>
               <AdministrativeArea Type="State">
                  <AdministrativeAreaName>NSW</AdministrativeAreaName>
                  <Locality>
                     <LocalityName>CHATSWOOD</LocalityName>
                     <PostBox Type="POBox">
                        <PostBoxNumber>773</PostBoxNumber>
                        <PostalCode>
                           <PostalCodeNumber>2057</PostalCodeNumber>
                        </PostalCode>
                     </PostBox>
                  </Locality>
               </AdministrativeArea>
            </Country>
         </AddressDetails>
      </xAL>
   </Record>
 </xNAL>
```

4.3 Don't get confused – keep it simple

Some users might feel that xNAL provides too much information to represent a simple name for their application. This is not true and the example in the previous section confirms this. xNAL can be used to define names or addresses in simple terms or in complex terms. It is up to the users to decide how they want to implement xNAL.

Important: Use only elements and attributes that make sense to you. Ignore the rest that are needless for you.

Enough flexibility is provided to make the name representation simple without using the detailed level of tags. Most of the elements and attributes are optional.

4.4 Namespaces and Versions

xNAL Schema's namespace is:

urn:oasis:names:tc:ciq:xsdschema:xNAL:[major version number]
where [major version number] is substituted with a number (e.g. 2.0, 2.5, etc.)

Schemas with different major version numbers are not compatible.

Attribute *version* of Schema's element *schema* indicates minor version number. Schemas with different minor version numbers are backward compatible.

DTD provides an attribute called "Version" that defines the version number of the DTD.

4.5 XML Schema: Extensibility

xNAL Schema was designed to be extensible.

1. some elements can have any child elements from ***##other*** namespaces (any that is not xNAL namespace).
2. all elements can have any attributes from ***##other*** namespaces (any that is not xNAL namespace).
3. key elements and types are declared globally to be reused by other schemas.

4.6 XML Schema: Document Fragments

xNAL Schema can be used to validate document fragments with globally declared elements as root elements.

4.7 Deep Nesting vs. Flat Structure

xNAL Schema/DTD allows dual way of reflecting relationships between entities by using xNL and xAL DTD/Schema: building a hierarchy or setting a reference. xNL and xAL provides Primary (*NameDetailsKey* for xNL and *AddressDetailsKey* for xAL) for and Foreign Keys (*NameDetailsKeyRef* for xNL) to enable setting reference to name details of a person/organisation

and address details of a person/organisation within and to an external XML document. This helps to simplify deep nesting of documents. This feature is an option and is not mandatory.

Note that *NameDetailsKey*, *NameDetailsKeyRef*, and *AddressDetailsKey* are not a constraint and existence of the referenced element is not checked at validation.

4.8 Where to start

Understanding this schema/DTD can be difficult for some users. To make it easier we would suggest you to undertake the following exercises:

- Read this document
- Take a look at the examples of XML documents for xNAL
- Take a look at schema/DTD diagrams.
- Try to build the structures you need using the schema/DTD.

The meaning of every element and attribute is described using *annotation*/*documentation* elements in XML schema.

For full schema description you can either go through the schema's/DTDs source code or use the detailed description of elements in this document or in the HTML document.

4.9 Compatibility between DTD and Schema

Instances of XML documents valid for xNAL W3C Schema may not always be valid for xNAL DTD and vise-versa, but the structures are almost identical.

4.10 Document Exchange between different parties

xNAL provides descriptions for every element and attribute, but it is up to the users how they implement it.

If you want to exchange information between different parties make sure that they are compatible:

1. all parties use the same namespace and version
2. all parties use the same interpretation of xNAL elements and attributes
3. all parties agree on enumerations and values used to describe types of data (for example element FirstName has attribute Type to indicate that the first name is full, formal, short form and etc., which is likely to be a predefined list of values for one party, but not compatible with a corresponding list of another party).

5.0 Other Name and Address Standard Initiatives

A number of name and address XML standards efforts are underway throughout the world. To a large extent, these standards have been designed with a particular business requirement in mind, for example, the expedient delivery of a piece of mail. This has generally meant that while the particular standard is appropriate for the purpose for which it was designed, it is frequently not suitable for a variety of other purposes. **This is the key differentiator between xNAL and the other name and address initiatives throughout world.**

5.1 How does xNAL differ from the other standardisation efforts

xNAL is the only Name and Address XML Standard in industry to-date that is open, vendor neutral, application independent (i.e. independent of postal services, CRM, name and address parsing, matching and validation, etc.) and importantly, global (designed to handle names and addresses of any country in a detailed level [detailed elementisation of name and address data] or in an abstract level).

Most of the other standards are based around the postal and address database businesses (e.g. mail delivery). But, it is important to note that name and address data are the most commonly used entities by businesses for many different purposes (e.g. CRM, address validation, marketing, data quality, data validation, customer profiling, census, segmentation, personal information, etc.). Sticking in specific postal rules and postal specific elements in name and address standards for mail delivery purposes could be overkill to other non-postal businesses. The ideal solution to build a name and address standard for businesses like postal services is to take an application independent name and address standard like xNAL and then extend it. We are in the 21st century and surprisingly, less than 40 countries of the 240+ countries have formal postal strategies (postal rules) that have been implemented and are followed. Only a handful of developed Western/European countries in the world have a Postal/National Address Reference File to perform address validation and verification. It is therefore, a long haul and a long hard journey to go through before postal authorities of countries around the world would agree and start to use a single global name and address standard designed for postal services.

The CIQ TC therefore, made sure that the standards that it develops could cope with any type of name and address that needs an abstract level or detailed level of representation of name and address data. It took more than two years of serious hard work from the CIQ TC to develop xNAL involving experts with many years of expertise in the following:

- International (Global) Address Database Management Services
- Use and management of International Name and Address data in small to large complex enterprise applications and systems (e.g. CRM, DW/DM, e-business, Customer Information Systems, etc)
- International Name and Address Data Integrity and Quality Management
- International Name and Address Parsing, Matching, Validation and Verification
- International Name and Address Data Management Tools
- Postal Address Certification

The above skills and expertise led to the successful development of xNAL and ensured that it is open, application independent, vendor neutral, and truly global (international). xNAL also provides options to represent postal service elements. It also provides options to use external namespace references in its schema structure.

Following are the other initiatives in progress to develop a name and address standard:

Project Team	Consortium/ Organisation	Comments
ASC/X12 Transaction Set 101 for Domestic Name and Address List	United States Postal Services, Industry leaders and DISA	Specific to postal service business. Domestic (USA) name and address for transmission of strung address lines or parsed address elements. No XML involved. USA specific.
UN/PROLST	UN/EDIFACT for international name and address lists based on Transaction Set 101	Specific to postal service business. Specifically designed for Mail delivery and uses code tables and templates Working closely with UPU on address templates Some XML work underway. More USA specific.
British Standard 7666	BSI IST/36	XML schema developed. Specific to names and addresses of U.K
CEN TC 331/WG3	European Standardisation Body and Universal Postal Union (UPU)	Specific to postal service business. Specifically designed for Mail delivery and postal services. No schemas available yet. Still at a requirements/specifications stage. Aimed at international name and address. A long way to go.
ECCMA International Address Element Code	Electronic Commerce Code Management Association	Specific to postal service business. Developed the International Address Element Code (IAEC) that is a schema that identifies the component data elements of a name and address. This is to improve the distribution of name and address information and the formatting of international addresses for mailing and postal services purpose. Address templates are developed to specify the order in which name and address information will be displayed line by line in an envelope/letter. Though aimed as international standard, only very few developed countries are involved in this project. A long way to go. IAEC is very USA centric when you look into the table. Nothing to do with XML.
GCA Address Data Interchange Specification (ADIS)	GCA/IDEAlliance	Specific to postal service business. An industry name and address standard for domestic (USA) and International Address Management and Mail production using Address elements. Compatible with USA EDI TS-101 standard. Supports PROLST. This standard also transmits printer-oriented information for printing addresses on envelopes.
HR-XML	HR-XML Consortium	Does not concentrate on name and address standards, but has developed its own name and address standard as part of its specifications. Claims to do international addressing, but uses simple address lines to tag complex international addresses. Therefore, the standard does not help to interpret international addresses and hence, cannot be used effectively in

Project Team	Consortium/ Organisation	Comments
		applications (Example: Parsing, matching, validation, verification, etc) where detailed elementisation of name and address data is important. Lack of understanding of the complexity of international names and addresses. For example, uses elements like Given Name and Family Name (without even an option to define what type of name it is) that has no meaning for many non-western/non European Countries, as there is no such concept.
UK GovTalk Address	UK Government	XML schemas that extends on BS 7666 Standards for addresses and specific to UK addresses only.
Universal Postal Union (UPU)	Universal Postal Union (UPU)	Specific to postal service business. Aligned with the USPS, UN/PROLST and works with these groups closely.
Address Data Content Standard	US Federal Geographic Data Committee	Aims at building a standard for sharing address information. Applicable to addresses having a spatial component and primarily for geographic data. No XML standards development.
Australian Standard for Exchange of Client Information AS-4590	Standards Australia	No XML standard development. But a general standard has been developed for managing Australian names and addresses and other client information.
FINAL DRAFT AUSTRALIAN/NEW ZEALAND STANDARD: Geographic Information -Rural and Urban Addressing	Standards Australia Standards New Zealand	No XML standards implementation. But a general standard for geographical addressing for the Australian and New Zealand countries.
HL7	HL7 Consortium	Does not concentrate on name and address standards. But has developed a simple name and address XML standard as part of its overall efforts. Claims to have developed an international name and address standard as part of health care standard. But uses simple address lines to tag complex international addresses. Therefore, the standard does not help to interpret international name and addresses and hence, cannot be used effectively in applications where names and addresses are important. Lack of understanding of the complexity of international names and addresses.

For more details about some of the above standards, go to:
http://xml.coverpages.org/namesAndAddresses.html

6.0 xNAL Grammar

This section describes the xNAL Grammar in detail. We have used the DTD version of xNAL to generate the diagrams and to explain the grammar. However, note that the structures of DTD and Schema are compatible except for the ##*other* element used in the Schema. Moreover, in Schema, structures are defined as elements (local and global), simple type, and complex type or of a particular type.

For detailed documentation of the XML Schema version of xNAL, users are recommended to download the HTML documentation of xAL from http://www.oasis-open.org/committees/ciq.

How to read the diagrams in the following sections:

1	**:**	**Either Or**
?	**:**	**Optional** (0 or more occurrences)
+	**:**	**At least 1** (1 or more occurrences)
♦	**:**	**An Element**
●	**:**	**An Attribute**
	**:**	**Has sub elements**

XML Containers consist of sub-XML elements and are not used to tag a piece of data directly. They use their sub-elements to tag the data. XML Elements are used to tag a piece of data directly. Let us consider the following example:

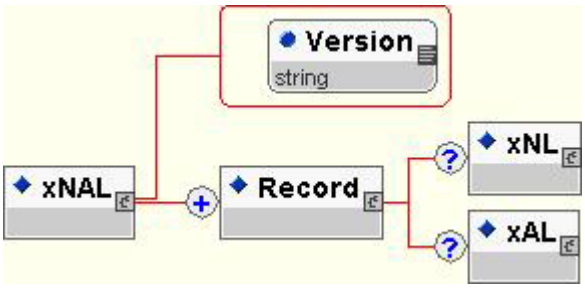
<**Name**>
  <FirstName *Type*=”Given Name”>Ram</FirstName>
  <LastName>Kumar</LastName>
</Name>

<**Name**> is the Container, <FirstName> and <LastName> are the XML Elements and *Type* is the Attribute.

In the following sections, we deliberately use examples of names and addresses that are represented using xNAL at a detailed representation. It is emphasised here again that name and addresses need not be represented at a detailed level. It depends upon the application requirements to define the level representation.

The xNAL DTD/Schema is straightforward as it uses xNL and xAL DTD/Schemas as references.

The figure below shows the xNAL grammar:



"xNAL" is a container and is the root element consisting of a sub-element called "Record" that can occur multiple times, but at least once. The attribute "Version" defines the version of xNAL used (specific to DTD only) and has a fixed value. For example, the value is "2.0" for version number 2.0.

Example:

```
<xNAL Version="2.0">
  <Record>
      ……..
      ……..
  </Record>
  <Record>
      ……..
      ……..
  </Record>
</xNAL>
```

"xNL" is a container and is a sub-element of "Record" element. xNL defines the name details and is optional (can occur 0 or once). See the xNL Specifications document for details about xNL.

"xAL" is a container and is a sub-element of "Record" element. xAL defines the address details and is optional (can occur 0 or once). See the xAL Specifications document for details about xAL.

7.0 Examples

Following are some of the xNAL examples.

7.1 Example 1

Mr.Ram V. Kumar
C/O Privacy Link
PO Box: 773, Chatswood, NSW 2057, Australia

```
<xNAL>
   <Record>
      <xNL>
          <NameDetails PartyType="Person">
            <NameLine>Ram V Kumar</NameLine>
            <DependencyName PartyType="Organisation"
                        DependencyType="C/O">
        <NameLine>PrivacyLink</NameLine>
       </DependencyName>
         </NameDetails>
      </xNL>
      <xAL>
     <AddressDetails>
        <Address>POBox: 773, Chatswood,NSW 2057, Australia</Address>
     </AddressDetails>
    </xAL>
   </Record>
</xNAL>
```

OR

```
<xNAL>
   <Record>
      <xNL>
         <NameDetails PartyType="Person">
            <PersonName>
               <Title>Mr</Title>
               <FirstName NameType="GivenName">Ram</FirstName>
               <MiddleName Type="Initial">V</MiddleName>
               <LastName NameType="SurName">Kumar</LastName>
            </PersonName>
            <DependencyName PartyType="Organisation"
                            DependencyType="C/O">
               <OrganisationNameDetails>
                  <NameLine>PrivacyLink</NameLine>
               </OrganisationNameDetails>
            </DependencyName>
         </NameDetails>
      </xNL>
```

```
    <xAL>
      <AddressDetails AddressType="Postal"
                      CurrentStatus="Investment"
                      ValidFromDate="1 Jan 2000"
                      ValidToDate="31 March 2000">
        <Country>
          <CountryName>Australia</CountryName>
          <AdministrativeArea Type="State">
            <AdministrativeAreaName>NSW</AdministrativeAreaName>
            <Locality>
              <LocalityName>CHATSWOOD</LocalityName>
              <PostBox Type="POBox">
                <PostBoxNumber>773</PostBoxNumber>
                <PostalCode>
                  <PostalCodeNumber>2057</PostalCodeNumber>
                </PostalCode>
              </PostBox>
            </Locality>
          </AdministrativeArea>
        </Country>
      </AddressDetails>
    </xAL>
  </Record>
</xNAL>
```

7.2 Example 2

Captain James Ruddock, C/O Australian Armed Forces in East Timor

```
<xNAL>
  <Record>
    <xNL>
      <NameDetails PartyType="Person">
        <PersonName>
          <Title>Captain</Title>
          <FirstName>James</FirstName>
          <LastName>Ruddock</LastName>
        </PersonName>
        <DependencyName PartyType="Organisation"
                        DependencyType="C/O">
          <OrganisationNameDetails Type="Militiary">
            <OrganisationName>
              Australian Armed Forces
            </OrganisationName>
          </OrganisationNameDetails>
        </DependencyName>
      </NameDetails>
    </xNL>
    <xAL>
      <AddressDetails>
        <Country>
          <CountryName>East Timor</CountryName>
        </Country>
      </AddressDetails>
    </xAL>
```

```
   </Record>
</xNAL>
```

7.3 Example 3

Professor and Chairman, School of Computer Science and Engineering,
Asian Institute of Technology, G.P.O. Box 4, Klong Luang,
Pathumthani 12120, Thailand

```
<xNAL>
  <Record>
     <xNL>
        <NameDetails PartyType="Person">
           <Function>Professor and Chairman</Function>
        </NameDetails>
     </xNL>
     <xAL>
        <AddressDetails>
           <Country>
              <CountryName>Thailand</CountryName>
              <AdministrativeArea Type="Province">
                <AdministrativeAreaName>
                   Pathumthani
                </AdministrativeAreaName>
                 <Locality Type="District">
                   <LocalityName>Klong Luang</LocalityName>
                   <PostBox Type="G.P.O">
                      <PostBoxNumber>4</PostBoxNumber>
                      <Firm Type="University">
                         <FirmName>
                            Asian Institute Of Technology
                         </FirmName>
                         <Department>
                            <DepartmentName>
                               School of Computer Science and Engineering
                            </DepartmentName>
                         </Department>
                      </Firm>
                   </PostBox>
                   <PostalCode>
                      <PostalCodeNumber>12120</PostalCodeNumber>
                   </PostalCode>
                 </Locality>
              </AdministrativeArea>
           </Country>
        </AddressDetails>
     </xAL>
  </Record>
</xNAL>
```

7.4 Example 4

ATTN THE MANAGER
FLORIDA SHOPPING MALL
287 VICTORIA STREET
MIAMI FLORIDA 33136

```
<xNAL>
   <Record>
     <xNL>
       <NameDetails PartyType="Person">
         <AddresseeIndicator>ATTN</AddresseeIndicator>
         <Function>THE MANAGER</Function>
       </NameDetails>
     </xNL>
     <xAL>
       <AddressDetails>
         <AdministrativeArea>
           <AdministrativeAreaName>FLORIDA</AdministrativeAreaName>
           <Locality>
             <LocalityName>MIAMI</LocalityName>
             <Thoroughfare>
               <ThoroughfareNumber>287</ThoroughfareNumber>
               <ThoroughfareName>VICTORIA</ThoroughfareName>
               <ThoroughfareTrailingType>
                    STREET
               </ThoroughfareTrailingType>
               <Premise Type="SHOPPING MALL">
                 <PremiseName TypeOccurrence="After">FLORIDA
                 </PremiseName>
               </Premise>
             </Thoroughfare>
             <PostalCode>
               <PostalCodeNumber>33136</PostalCodeNumber>
             </PostalCode>
           </Locality>
         </AdministrativeArea>
       </AddressDetails>
     </xAL>
   </Record>
</xNAL>
```

7.5 Example 5

K.S.Palanisamy Gounder, Balu Illam,
Attukkaaran Thottam, Karattoor, Kuppandapalayam (P.O)
Via-Athani, Kovai District, 638012, Tamilnadu, India

```
<xNAL>
  <Record>
     <xNL>
        <NameDetails PartyType="Person">
           <PersonName>
              <FirstName Type="Initial">K</FirstName>
              <MiddleName Type="Initial">S</MiddleName>
              <MiddleName>Palanisamy</MiddleName>
              <LastName>Gounder</LastName>
           </PersonName>
        </NameDetails>
     </xNL>
     <xAL>
        <AddressDetails>
           <Country>
              <CountryName>India</CountryName>
              <AdministrativeArea Type="State">
                 <AdministrativeAreaName>Tamilnadu
                 </AdministrativeAreaName>
                 <SubAdministrativeArea Type="District"
                                        Indicator="(Dist)">
                    <SubAdministrativeAreaName>Kovai
                    </SubAdministrativeAreaName>
                 </SubAdministrativeArea>
                 <Locality>
                    <LocalityName>Athani</LocalityName>
                    <PostOffice Indicator="(P.O)">
                       <PostOfficeName>Kuppaandapalayam</PostOfficeName>
                       <PostalCode>
                          <PostalCodeNumber>638012</PostalCodeNumber>
                       </PostalCode>
                    </PostOffice>
                    <DependentLocality Type="Town" Connector="Via">
                       <DependentLocalityName>Karattoor
                       </DependentLocalityName>
                       <Premise Type="Farm">
                          <PremiseName>Attukkaaran Thottam</PremiseName>
                          <SubPremise Type="House">
                             <SubPremiseName>Balu Illam</SubPremiseName>
                          </SubPremise>
                       </Premise>
                    </DependentLocality>
                 </Locality>
              </AdministrativeArea>
           </Country>
        </AddressDetails>
     </xAL>
```

```
   </Record>
</xNAL>
```

7.6 Example 6

Jessica Wood
Standard Chartered Bank
30th Floor, Standard Chartered Tower
388 Kwun Tong Rd, Kwun Tong
Hong Kong

```
<xNAL>
  <Record>
     <xNL>
        <NameDetails PartyType="Person">
           <PersonName>
              <FirstName>Jessica</FirstName>
              <LastName>Wood</LastName>
           </PersonName>
        </NameDetails>
     </xNL>
     <xAL>
        <AddressDetails>
           <Country>
              <CountryName>Hong Kong</CountryName>
              <Locality>
                 <LocalityName>Kwun Tong</LocalityName>
                 <Thoroughfare>
                    <ThoroughfareNumber>388</ThoroughfareNumber>
                    <ThoroughfareName>Kwun Tong</ThoroughfareName>
                    <ThoroughfareTrailingType>Rd
                    </ThoroughfareTrailingType>
                    <Premise Type="Building">
                       <PremiseName>Standard Chartered Tower</PremiseName>
                       <SubPremise Type="Floor">
                          <SubPremiseNumber>30</SubPremiseNumber>
                          <Firm Type="Bank">
                             <FirmName>Standard Chartered Bank</FirmName>
                          </Firm>
                       </SubPremise>
                    </Premise>
                 </Thoroughfare>
              </Locality>
           </Country>
        </AddressDetails>
     </xAL>
  </Record>
</xNAL>
```

7.7 Example 7

Juci & Duso Arnon
Gaaton A Kibbutz
DN Ashrat 25130
ISRAEL

```
<xNAL>
  <Record>
     <xNL>
        <NameDetails PartyType="Person">
           <NameLine NameType="Joint Name">Juch and Duso Arnon
           </NameLine>
        </NameDetails>
     </xNL>
     <xAL>
        <AddressDetails>
           <Country>
              <CountryName>ISRAEL</CountryName>
              <Locality Type="Collective Farming Community">
                 <LocalityName>Gaaton A Kibbutz</LocalityName>
                 <PostOffice Type="Mobile Post">
                    <PostalRoute>
                       <PostalRouteName>DN Ashrat</PostalRouteName>
                    </PostalRoute>
                 </PostOffice>
                 <PostalCode>
                    <PostalCodeNumber>25130</PostalCodeNumber>
                 </PostalCode>
              </Locality>
           </Country>
        </AddressDetails>
     </xAL>
   </Record>
</xNAL>
```

7.8 Example 8

C/ W A GORRY AND CO STE 140
14TH FL MLC CENTRE
CNR GEORGE & ADELAIDE STS
BRISBANE QLD 4000

```
<xNAL>
  <Record>
    <xNL>
      <NameDetails PartyType="Organisation">
        <DependencyName DependencyType="C/">
          <OrganisationNameDetails>
            <NameLine>W A GORRY AND CO</NameLine>
          </OrganisationNameDetails>
        </DependencyName>
      </NameDetails>
    </xNL>
    <xAL>
      <AddressDetails>
        <AdministrativeArea>
          <AdministrativeAreaName>QLD</AdministrativeAreaName>
          <Locality>
            <LocalityName>BRISBANE</LocalityName>
            <Thoroughfare DependentThoroughfares="Yes"
                          DependentThoroughfaresIndicator="CORNER OF"
                          DependentThoroughfaresConnector="AND"
                           DependentThoroughfaresType="STS">
              <ThoroughfareName>GEORGE</ThoroughfareName>
              <DependentThoroughfare>
                <ThoroughfareName>ADELAIDE</ThoroughfareName>
              </DependentThoroughfare>
              <Premise Type="Building">
                <PremiseName>MLC CENTRE</PremiseName>
                <SubPremise Type="FL">
                  <SubPremiseNumber Indicator="TH"
                                    IndicatorOccurrence="After"
                                     NumberTypeOccurrence="Before">14
                  </SubPremiseNumber>
                  <SubPremise Type="STE">
                    <SubPremiseNumber NumberTypeOccurrence="After">140
                    </SubPremiseNumber>
                  </SubPremise>
                </SubPremise>
              </Premise>
            </Thoroughfare>
            <PostalCode>
              <PostalCodeNumber>4000</PostalCodeNumber>
            </PostalCode>
          </Locality>
        </AdministrativeArea>
      </AddressDetails>
    </xAL>
```

```
    </Record>
</xNAL>
```

8.0 References

- Name and Address Markup Language (NAML) Specifications document (Version 1-1.3), MasterSoft International, April 2000

- Global Address Specifications document (Version 1-1.2), December 2000

- xNL Specifications Document for W3C DTD/Schema, OASIS CIQ TC, http://www.oasis-open.org/committees/ciq, May 2002

- xAL Specifications Document for W3C DTD/Schema, OASIS CIQ TC, http://www.oasis-open.org/committees/ciq, May 2002

- Ram Kumar, XML Standards for Customer Information Quality Management, XML Journal, Vol.1, No.2, July 2000, pp.41-45.

- Using the UN/PROLST Version 1.1, May 2001

- GCA-ADIS Address Management Specifications Document, March 2001

- Australian Standard (AS 4590-1999) for Interchange of Client Information

- Postal Services – Address data bases, CEN TC 331 Document, December 2000

- Ram Kumar, XML Standards for Global Customer Information Management, DMReview, Vol.12, No.5, May 2002