Extensible Name Language (xNL) Standard Description Document for W3C DTD/Schema

Version 2.0

(Approved Committee Specification)

A Standard from the Customer Information Quality Technical Committee

Copyright© OASIS. All Rights Reserved

CHANGE HISTORY

Status	Version	Date	Author	Summary of Changes
Draft	1.0	01 March 2001	CIQ-TC	Initial Draft
Draft	1.1	17 May 2001	CIQ-TC	In conjunction with V1.1 of xNL DTD
Draft	2.0	31 May 2002	CIQ TC	New release of xNL DTD/Schema that supports postal services elements, namespaces and new structures
Final	2.0	24 July 2002	CIQ TC	Final Version of 2.0 after review

Copyright© OASIS. All Rights Reserved

OASIS COPYRIGHT NOTICE

Copyright (C) The Organization for the Advancement of Structured Information Standards [OASIS] (1 March 2001). All Rights Reserved

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 paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to OASIS, except as needed for the purpose of developing OASIS specifications, in which case the procedures for copyrights defined in the OASIS Intellectual Property Rights document 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 RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE."

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS's procedures with respect to rights in OASIS specifications can be found at the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification, can be obtained from the OASIS Executive Director.

OASIS invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this specification. Please address the information to the OASIS Executive Director.

OASIS has been notified of intellectual property rights claimed in regard to some or all of the contents of this specification. For more information consult the online list of claimed rights.

TABLE OF CONTENTS

1.0 ACKNOWLEDGEMENTS 6

2.0 INTRODUCTION 8

2.1 EXTENSIBLE NAME AND ADDRESS LANGUAGE 8
2.2 THE GOAL OF XNAL 8

3.0 THE OBJECTIVE AND SCOPE 9

4.0 EXTENSIBLE NAME LANGUAGE (XNL) 9

4.1 XNL 9
4.2 THE GOAL 9
4.3 THE CHALLENGE 9
4.4 WHAT DOES XNL NOT REPRESENT 10

5.0 DEFINING NAMES IN XNL 10

5.1 RESTRICTIONS 10
5.2 USING FIRST NAME, MIDDLE NAME(S) AND LAST NAME 11
5.3 USING TITLES AND PRECEDING TITLES 13

6.0 USING THE XNL DTD/SCHEMA 14

6.1 PURPOSE OF THE XML DTD/SCHEMA FOR NAMES 14
6.2 FLEXIBILITY 14
6.2.1 Example 14
6.3 DON'T GET CONFUSED – KEEP IT SIMPLE 16
6.4 NAMESPACES AND VERSIONS 16
6.5 XML SCHEMA: EXTENSIBILITY 16
6.6 XML SCHEMA: DOCUMENT FRAGMENTS 17
6.7 DEEP NESTING VS. FLAT STRUCTURE 17
6.8 WHERE TO START 18
6.9 COMPATIBILITY BETWEEN DTD AND SCHEMA 19
6.10 DOCUMENT EXCHANGE BETWEEN DIFFERENT PARTIES 19

7.0 XNL GRAMMAR 19

7.1 XNL ELEMENT 20
7.2 NAMELINE ELEMENT 20
7.3 NAMEDETAILS ELEMENT 22
7.3.1 Example 24
7.4 PERSONNAME ELEMENT 25
7.4.1 Example 28
7.5 ORGANISATIONNAMEDETAILS ELEMENT 29
7.5.1 Example 30
7.6 FORMERNAME ELEMENT 31
7.7 KNOWNAS ELEMENT 35
7.8 JOINTPERSONNAME ELEMENT 38
7.8.1 Example 40
7.9 DEPENDENCYNAME ELEMENT 40
7.9.1 Example 41
7.10 ORGANISATIONFORMERNAME ELEMENT 42
7.10.1 Example 43
7.11 ORGANISATIONKNOWNAS ELEMENT 44

Copyright© OASIS. All Rights Reserved

7.11.1 Example *45*

8.0 REFERENCES **46**

Copyright© OASIS. All Rights Reserved

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

Copyright© OASIS. All Rights Reserved

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.

Copyright© OASIS. All Rights Reserved

2.0 Introduction

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.

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.

2.1 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.

2.2 The Goal of xNAL

The goal of xNAL is:
- Open
- Vendor Neutral
- Application Independent, and
- Global, i.e., ability to represent names and addresses of any country irrespective of culture, religion, language and geographic location.

Copyright© OASIS. All Rights Reserved

3.0 The Objective and Scope

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

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

4.0 extensible Name Language (xNL)

4.1 xNL

The objective of xNL is to describe a common structure for Personal and Organization Names that would enable any applications that want to represent customer names 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 xNL grammar or parts of it.

It is important to read the following document as a pre-requisite to this document:

- xNAL Specifications Document Version 2.0 or W3C DTD/Schema

4.2 The Goal

The goal of xNL is:
- Open
- Vendor Neutral
- Application Independent
- Global, i.e., ability to represent names of any country irrespective of culture, religion, language and geographic location
- Flexible enough to handle simple representation of names (Example: Simple user registration system) to complex representation of names (Example: name parsing).

4.3 The Challenge

The challenge for xNL is to provide the ability to handle the following:
- About 36+ customer name formats
- Represented in 5,000+ languages (dialects), and at the same time,
- Should be application independent, open and vendor neutral.

Copyright© OASIS. All Rights Reserved

4.4 What does xNL not represent

xNL only defines the XML vocabulary to represent customer names.

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

5.0 Defining Names in xNL

Names in general, can be classified into two, namely:
- Personal Names
- Organisation Names, where Organisation name can be Club, Association, University, Company, Hospital, etc.

However, sometimes a name can be classified under general name when it is not clear whether it is a person name or a company name.

5.1 Restrictions

xNL grammar is set up to uniquely define a customer. It is not set up to define more than one customer at a time. For example,

Ram Kumar
Chief Technologist
MSI Business Solutions
PO Box: 773, Chatswood, NSW 2067

“Ram Kumar” is the person and his designation is “Chief Technologist”. This can be represented in xNL. But to represent “MSI Business Solutions” and its address, the xAL DTD/Schema must be used as MSI Business Solutions is considered as part of the address.

In the example,

MSI Business Solutions
PO Box: 773, Chatswood, NSW 2067

“MSI Business Solutions” is the name of the Organisation and it can be represented in xNL. However, the address part must be represented using xAL. You cannot have both name and address in xNL.

Copyright© OASIS. All Rights Reserved

In the example,

Ram Kumar
Chief Technologist
C/O MSI Business Solutions
PO Box: 773, Chatswood, NSW 2067

"Ram Kumar" is the person and his designation is "Chief Technologist". This can be represented in xNL. "MSI Business Solutions" is the dependency name used by "Ram Kumar" and therefore, this dependency name can also be represented using xNL. However, the address part must be represented using xAL. You cannot have both name and address in xNL.

In the example,

Attention: CEO
MSI Business Solutions
PO Box: 773, Chatswood, NSW 2067

Attention and CEO can be represented in xNL. But MSI Business Solutions and the address part must be represented in xAL.

5.2 Using First Name, Middle Name(s) and Last Name

Choosing generic XML nametags that is meaningful to represent the types of person names was a complex and difficult task. For example, there are different types of names such as:

- First Name
- Middle Name
- Last Name
- Family Name
- Surname
- Given Name
- Christian Name
- Maiden Name
- Mother's Name
- Father's Name
- Fore father's name
- etc

These name types varies from country to country, culture, ethnicity and religion. For example, in some countries there is no concept of family name or surname. In countries like India, a name say, Mr.Ramkumar Venkatachalam means, "Ramkumar" is the name of the person and "Venkatachalam" is the name of the person's father. The person is often referred to as "Ramkumar" or "Mr.Ramkumar" or "Mr.V.Ramkumar" and not "Mr.Venkatachalam". Ramkumar's child takes the name as say, Miss. Nivetha Ramkumar or Miss.R.Nivetha, where "Nivetha is the child's name and "Ramkumar" is Nivetha's father's name. Sometimes a person takes both his/her father and mother's name as part of his/her full name. For example, "Miss.Nivetha Shantha Ramkumar", where "Shantha" is the mother of "Nivetha" and "Ramkumar" is the father of "Nivetha". Some people take many names that identify the generation of fore/grand/great grand fathers. As you see from these examples, there is no concept of Given name/Christian name/Surname/Family name/First Name, etc.

Copyright© OASIS. All Rights Reserved

In Japan, people carry two names, one is official and the other is non-official.

The above examples are some of the many complex examples for names. Therefore, we had to cater for all different types of names used throughout the world, what they mean and how they are represented when we designed xNL. Moreover, coming up with a generic nametag is also extremely important as in some countries, people have never heard of the words such as Given name, Surname, etc.

We therefore, decided to adopt the following approach to define the names:

- We will use “First Name”, “Middle Name” and “Last Name” to cover the types of names.

- “First Name”, “Middle Name”, and “Last Name” will define the position of the name in a name string. For example, in some countries, the name appearing first in the name string is always a Given Name or Christian Name and the name appearing last in the name string is always a Surname or a Family Name. In some countries, this is the opposite. Moreover, names in a name string can be represented in any order in a database. Let us say, “Ram” is the first name and “Kumar” is the last name. This name can appear as follows in a file:
  - Ram Kumar
  - Ram, Kumar
  - Kumar, Ram
  - Kumar Ram
  - Kumar, Ram

  As long as the name type is used to ensure that the meaning of the type of name is preserved, this should be fine.

- The XML elements to define the above three types of names are:
  - First Name : <FirstName>
  - MiddleName : <MiddleName>
  - Last Name : <LastName>

  These are the best generic tag names that we could think of that would cover the different types of names.

- We will use an attribute “NameType” for the above XML elements to define the type of name. For example, a First Name or Middle Name or Last Name could be any of the above name types listed (see first paragraph of this section). This helps to avoid any conflicts as for some, First Name could be a family name or a surname and for some, first name could be a given name or Christian name or even father’s name. Therefore, this attribute defines the actual meaning of the name, as “FirstName”, “MiddleName”, and “LastName” are only positional indicators within a name string and no more than that.

Let us take an example say, Graham Rhind. It can be represented as:
- Graham Rhind or
- Rhind, Graham

In the former case, “Graham” is the First Name and is of NameType “Given Name” and “Rhind” is the Last Name and is of NameType “Surname”. In the later case, “Rhind” is First Name and is of NameType “Surname” and “Graham” is Last Name and is of NameType “Given Name”.

Copyright© OASIS. All Rights Reserved

5.3 Using Titles and Preceding Titles

Use “PrecedingTitle” element (can occur multiple times) to define special titles such as Honorary titles (eg. Honourable, His Excellency, Her Highness, etc). Use the “Type” attribute to define the type of preceding title.

Use “Title” element (can occur multiple times) that occurs after preceding title to define titles such as:

- Profession (Professor, Engineer, Doctor, etc)
- Academic titles (Dr., DR., Engr., Dipl.Ing., etc)
- Person general title (Mr., Mrs., Herr, Miss, Ms, etc)
- Title as a single string (e.g. Herr Professor Dipl.Ing)

Use The “Type” attribute of “Title” element to define the type of title.

Copyright© OASIS. All Rights Reserved

6.0 Using the xNL DTD/Schema

6.1 Purpose of the XML DTD/Schema for names

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 elements (Example: Title, First Name, Middle Name and Last Name) to a name validation system that needs all the elements of a name, can be defined using this name schema.

6.2 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.

6.2.1 Example

Let us consider the following example that can be represented in some of the different ways to show the flexibility provided by xNL:

Mr.Ram Laxhman B Kumar
in care of Mr. Venkat Krishnan

```
<xNL>
 <NameDetails PartyType="Person">
   <NameLine>
     Mr.Ram Laxhman B Kumar in care of Mr. Venkat Krishna
   </NameLine>
 </NameDetails>
</xNL>
```

Copyright© OASIS. All Rights Reserved

OR

```
<xNL>
  <NameDetails PartyType="Person">
   <NameLine>Mr.Ram Laxhman B Kumar</NameLine>
   <DependencyName DependencyType="in care of">
    <NameLine>Mr. Venkat Krishnan</NameLine>
   </DependencyName>
 </NameDetails>
</xNL>
```

OR

```
<xNL>
  <NameDetails PartyType="Person">
    <PersonName>
       <Title>Mr</Title>
       <FirstName Type="GivenName">Ram</FirstName>
       <MiddleName>Laxhman</MiddleName>
       <MiddleName Type="Initial">B</MiddleName>
       <LastName NameType="SurName">Kumar</LastName>
    </PersonName>
    <DependencyName DependencyType="in care of">
      <NameLine>Mr. Venkat Krishnan</NameLine>
    </DependencyName>
   </NameDetails>
</xNL>
```

OR

```
<xNL>
  <NameDetails  PartyType="Person">
    <PersonName>
       <Title>Mr</Title>
       <FirstName Type="GivenName">Ram</FirstName>
       <MiddleName>Laxhman</MiddleName>
       <MiddleName Type="Initial">B</MiddleName>
       <LastName NameType="SurName">Kumar</LastName>
       <Alias>Ram</Alias>
       <FormerName>
          <NameLine NameType="Full Name">Ramkumar</NameLine>
       </FormerName>
    </PersonName>
    <DependencyName    PartyType="Person"
                      DependencyType="C/O">
         <PersonName>
             <Title>Mr</Title>
             <FirstName NameType="GivenName" Type="Official">
                  Venkat
            </FirstName>
             <FirstName NameType="GivenName"
                        Type="Unofficial">Venki</FirstName>
             <LastName>Krishnan</LastName>
         </PersonName>
```

Copyright© OASIS. All Rights Reserved

```
    </DependencyName>
  </NameDetails>
</xNL>
```

6.3 Don't get confused – keep it simple

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

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.

6.4 Namespaces and Versions

xNL Schema's namespace is:

urn:oasis:names:tc:ciq:xsdschema:xNL:[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.

6.5 XML Schema: Extensibility

xNL Schema was designed to be extensible.

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

Copyright© OASIS. All Rights Reserved

6.6 XML Schema: Document Fragments

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

6.7 Deep Nesting vs. Flat Structure

xNL Schema/DTD allows dual way of reflecting relationships between entities: building a hierarchy or setting a reference.

Example 1: *DependencyName* structure is nested inside *NameDetails* element

```
<?xml version="1.0" encoding="UTF-8"?>
<xNL xmlns="urn:oasis:names:tc:ciq:xsdschema:xNL:2.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:oasis:names:tc:ciq:xsdschema:xNL:2.0 xNL.xsd">
   <NameDetails NameDetailsKey="123">
      <PersonName>
         <Title>Mr</Title>
         <FirstName>Max</FirstName>
         <LastName>Wax</LastName>
      </PersonName>
      <DependencyName Type="in care of">
         <PersonName>
            <Title>Mr</Title>
            <FirstName>John</FirstName>
            <LastName>Johnson</LastName>
         </PersonName>
      </DependencyName>
   </NameDetails>
</xNL>
```

Copyright© OASIS. All Rights Reserved

Example 2: the dependency name is represented as another *NameDetails* element referenced from *DependencyName* element using *NameDetailsKeyRef* attribute as a foreign key and *NameDetailsKey* attribute of *NameDetails* element as a primary key.

```
<?xml version="1.0" encoding="UTF-8"?>
<xNL xmlns="urn:oasis:names:tc:ciq:xsdschema:xNL:2.0-draft"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:oasis:names:tc:ciq:xsdschema:xNL:2.0-draft
xNL.xsd">
   <NameDetails NameDetailsKey="111">
      <PersonName>
         <Title>Mr</Title>
         <FirstName>Max</FirstName>
         <LastName>Wax</LastName>
      </PersonName>
      <DependencyName Type="in care of" NameDetailsKeyRef="222"/>
   </NameDetails>
   <NameDetails NameDetailsKey="222">
      <PersonName>
         <Title>Mr</Title>
         <FirstName>John</FirstName>
         <LastName>Johnson</LastName>
      </PersonName>
   </NameDetails>
</xNL>
```

These two examples show that it is up to the user which method to use.

The same rules apply for some other elements. See xNL Grammar for more details.

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

6.8 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 xNL
- Take a look at schema/DTD diagrams.
- Try to build the structures you need using the schema/DTD.

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.

Copyright© OASIS. All Rights Reserved

6.9 Compatibility between DTD and Schema

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

6.10 Document Exchange between different parties

xNL 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 xNL 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).

7.0 xNL Grammar

This section describes the xNL Grammar in detail. We have used the DTD version of xNL to generate the diagrams and to explain the grammar. However, note that the structures of DTD and Schema are compatible except for the ##o*ther* 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 xNL, users are recommended to download the HTML documentation of xNL 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.

For ease of reading, under XML Elements column in the tables in the following sections, XML Tag names in **bold** are XML Containers (consisting of sub-XML elements), XML Tags in regular text are XML Elements and Tag names in *italics* in the Description column of the tables are Attributes of XML elements. Let us consider the following example:

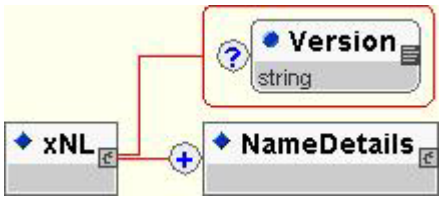
Copyright© OASIS. All Rights Reserved

```
<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.

7.1 xNL Element

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



Example:

```
<xNL Version=”2.0”>
  <NameDetails>
      ……..
      ……..
   </NameDetails>
   <NameDetails>
      ……..
      ……..
   </NameDetails>
 </xNL>
```

7.2 NameLine Element

NameLine element is used to represent name as a free format text. NameLine will be helpful when one intends to create a flat structure for their data instead of nested structure.

“Type” attribute: Defines the type of data as a free format text. Example: Former name, Nick name, Known as, etc. or anything else to help identify the line as part of the name.

“NameType” attribute: Defines the meaning of the element. Example: First Name can be Christian name, Given name, first name, etc.

Copyright© OASIS. All Rights Reserved

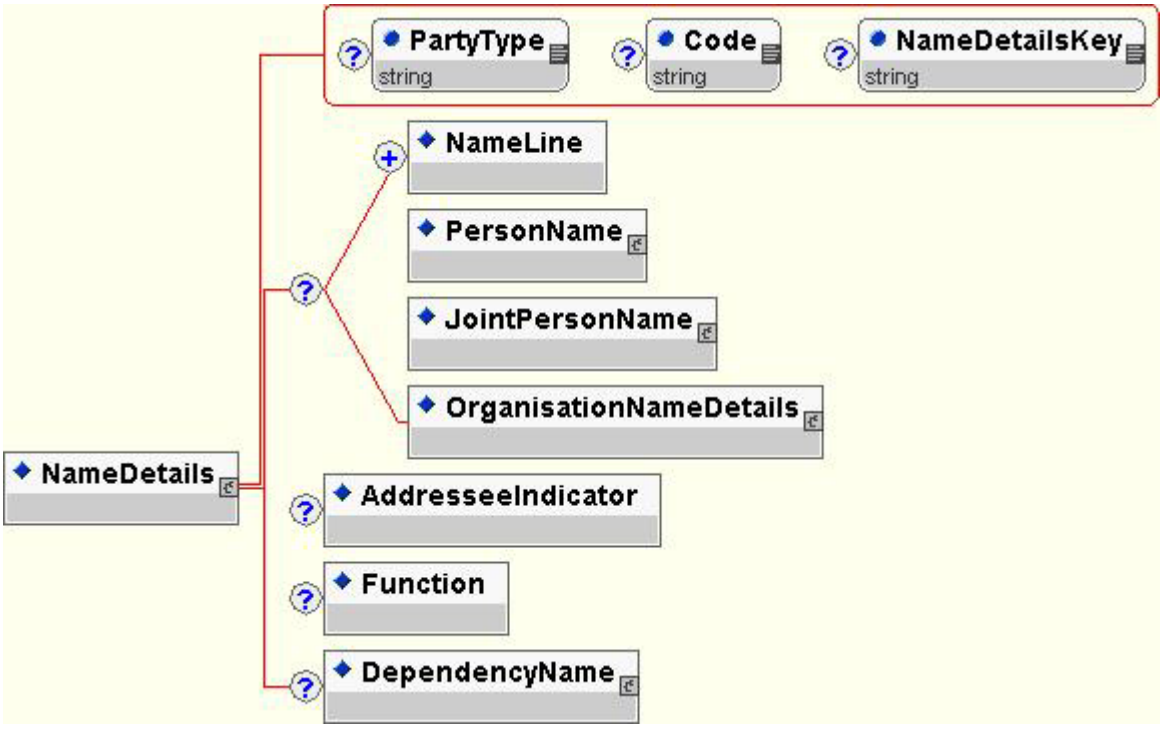
“Code” attribute: Helps to Indicate the name element code defined by postal standard groups like ECCMA, ADIS, UN/PROLIST specifically for postal service applications.

Copyright© OASIS. All Rights Reserved

7.3 NameDetails Element

NameDetails is the element that defines a name (person/organisation) in detail by breaking it down into sub-elements.

A NameDetails Element has the following elements and their relationship is shown in the figure below:



For ease of reading, under XML Elements column in the tables below, XML Tag names in **bold** are XML Containers (consisting of sub-XML elements), XML Tags in regular text are XML Elements and Tag names in *italics* in the Description column of the tables are Attributes of XML elements. 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.

Copyright© OASIS. All Rights Reserved

Name Elements	xNL Elements (XML Tags)	Description
Name Details	**NameDetails**	This element is the sub-element of root element "xNL". This element can occur multiple times and it is mandatory that it occurs at least once (1 or more). This element is a container. This element provides the following attributes: *PartyType:* Defines the type of customer/party and is optional. Example, Club, Organisation, Person, etc. *Code:* Some postal services use a special code to define the element. Example: ECCMA Code Tables for postal services. *NameDetailsKey*: Defines the primary key and is optional. Key identifier for the element for not reinforced references from other elements. Not required to be unique for the document to be valid, but application may get confused if not unique. Extend this schema adding unique constraint if needed Example: <xNL> <NameDetails PartyType="Person"> <PersonName> ……. ……. </PersonName> </NameDetails> </xNL>
Description of the Addressee	AddresseeIndicator	This element is a sub-element of "NameDetails" element and is used to define the description of the addressee and this is purely for mailing purposes and is optional. This element can occur once and is optional (0 or 1). Example: ATTENTION To, ter Attentie van (in The Netherlands), etc. This element provides the following attribute: *Code:* Some postal services use a special code to define the element. Example: ECCMA Code Tables for postal services.
Name as a general free format text field	NameLine	This element is a sub-element of "NameDetails" element and is used to define the name of a customer (person/company) as a free format text field without breaking it into sub-elements. This element can occur multiple times and is optional (0 or more). See Section "NameLine Element" for more details.
Name of the person	**PersonName**	This element is a sub-element of "NameDetails" element and is used to define a person's name in detail. This element can occur once and is optional (0 or 1). See section "PersonName Element" for further details. Can occur once and is optional.
Name of the organisation	**OrganisationNameDetails**	This element is a sub-element of "NameDetails" element and is used to define an organisation's name in detail. This element can occur once and is optional (0 or 1). See section "PersonName Element" for further details. Can occur once and is optional.
Function/Position/Role	Function	This element is a sub-element of "NameDetails" element that defines the position of the person and is optional. This is purely for mailing purposes. Example: Managing Director, CEO, etc. This element has the following attribute: *Code:* Some postal services use a special code to define the element and is

Copyright© OASIS. All Rights Reserved

Name Elements	xNL Elements (XML Tags)	Description
		optional. Example: ECCMA Code Tables for postal services.
More than one person	**JointPersonName**	This element is a sub-element of "NameDetails" element and is used to define a joint person's name in detail. This element can occur once and is optional (0 or 1). See section "JointPersonName Element" for further details. Can occur once and is optional.
Dependent name on another name details	**DependencyName**	This element is a sub-element of "NameDetails" element and is used to define a dependent name in detail. This element can occur once and is optional (0 or 1). See section "DependencyName Element" for further details. Can occur once and is optional.

7.3.1 Example

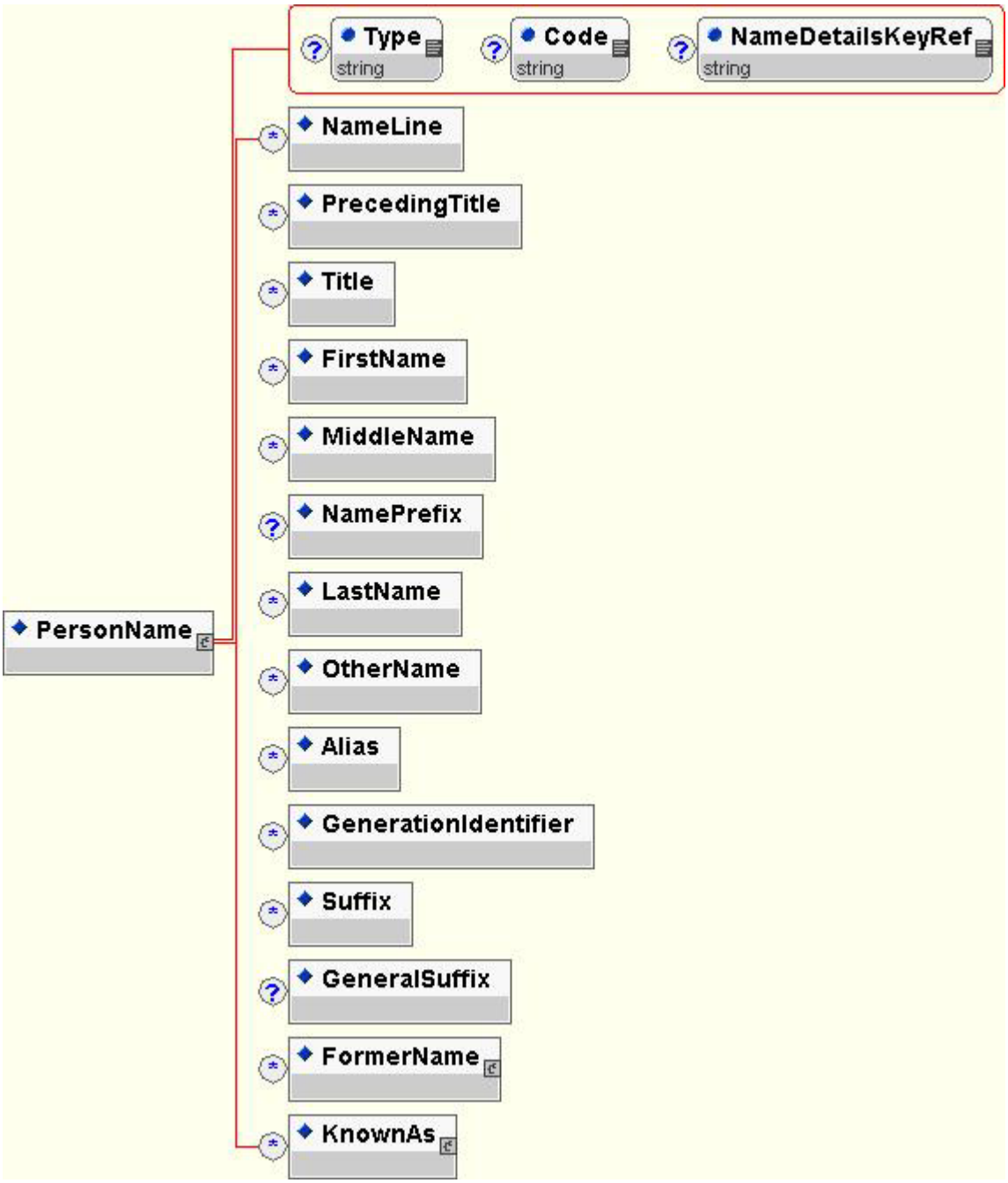
ATTENTION : Mr. Ram Kumar
CEO

```
<xNL>
   <NameDetails PartyType="Person">
      <NameLine NameType="FullName">Mr.Ram Kumar</NameLine>
      <AddresseeIndicator>ATTENTION:</AddresseeIndicator>
      <Function>CEO</Function>
   </NameDetails>
</xNL>
```

Copyright© OASIS. All Rights Reserved

7.4 PersonName Element

PersonName is the element that defines a person's name in detail by breaking it down into sub-elements.



Copyright© OASIS. All Rights Reserved

Name Elements	xNL Elements (XML Tags)	Description
Name of the Person	**PersonName**	This element is a sub-element of root element "NameDetails". This element can occur once and is optional (0 or 1). This element is a container and has sub-elements to define the name of a person in detail. This element provides the following attributes: *Type:* defines the type of name and is optional. Example: Former name, full name, etc. *NameDetailsKeyRef:* Reference to another "NameDetails" element with no foreign key reinforcement. The referenced element may be out of the document and the document is still valid. This attribute is optional. *Code:* Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services.
Name details of a person as a free format text	NameLine	This element is used to define the name of a person as a free format text field without breaking it into sub-elements. This element can occur multiple times and is optional (0 or more). See Section "NameLine Element" for more details.
Preceding Title	PrecedingTitle	This element is a sub-element of "PersonName" element. Can occur many times and is optional (0 or more). This element defines Preceding titles. Example: HIS EXCELLENCY, HONOURABLE, etc. This element provides the following attributes: *Type:* Defines the type of preceding title and is optional. Example: Honorary title *Code:* Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services.
Title	Title	This element is a sub-element of "PersonName" element. Can occur many times and is optional (0 or more). This element defines titles. This element provides the following attributes: *Type:* Defines the type of title and is optional. Example: Sex, Honorary, Profession, degree, greetings title, etc. *Code:* Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services
First Name	FirstName	This element is a sub-element of "PersonName" element. Defines first name of a person that is the position of the name in the name string. This element can occur multiple times and is optional (0 or more). This element provides the following attributes: *Type*: Defines the type for first name and is optional. Example: Old Name, Official, UnOfficial, Initials, etc. Some countries have two first names, one is official name and the other is an unofficial name. *NameType*: Defines the type of name and is optional. Example: Given Name, Christian Name, First Name, Surname, family name, etc. *Code:* Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services
Middle Name	MiddleName	This element is a sub-element of "PersonName" element and defines middle name of a person. This name is the position of the name in the name string. This element can occur multiple times and is optional (0 or more). This element provides the following attributes:

Copyright© OASIS. All Rights Reserved

Name Elements	xNL Elements (XML Tags)	Description
		Type: Defines the type for middle name and is optional. Example: Old Name, Official, UnOfficial, Initials, etc. *NameType*: Defines the type of name and is optional. Example: Given Name, Christian Name, First Name, etc. *Code:* Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services
Name prefix	NamePrefix	This element is a sub-element of "PersonName" element. This element can occur once and is optional. This element defines the prefix of a name. Example: van de, de la, etc. as in France and The Netherlands. This element provides the following attributes: *Type*: Defines the type NamePrefix and is optional. Example: Official, UnOfficial, etc. *NameType*: Defines the type of name that has the name prefix and is optional. Example: Last Name, SurName, Family Name, etc. *Code:* Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services
Last Name	LastName	This element is a sub-element of "PersonName" element and defines last name of a person. This element is the position of the name in the name string. This element can occur multiple times and is optional (0 or more). This element provides the following attributes: *Type*: Defines the type of last name and is optional. Example: Old Name, Official, UnOfficial, Initials, etc. *NameType*: Defines the type of name and is optional. Example: Last Name, SurName, Family Name, Father's name (as in India), Given name, Christian name etc. *Code:* Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services
Other Name	OtherName	This element is a sub-element of "PersonName" element and defines other names of a person. This element can occur multiple times and is optional (0 or more). Example: Yousuf Khan al Hatab al Sayad is the other name and is normally known as Yousuf Khan. Hatab is Yousuf's father's name and Sayad is Yousuf's grand father's name. This element provides the following attributes: *Type*: Defines the type for other name and is optional. Example: Old Name, Official, UnOfficial, etc. *NameType*: Defines the type of name and is optional. Example: Maiden Name, Patronymic name, Matronymic name, etc. *Code:* Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services
Former name	**FormerName**	See section "FormerName Element" for further details. This element can occur once and is optional (0 or 1).
Alias	Alias	This element is a sub-element of "PersonName" element and defines alias names of a person. This element can occur multiple times and is optional (0 or more). This element provides the following attributes: *Type*: Defines the type for other name and is optional. Example: Old Name, Official, UnOfficial, etc.

Copyright© OASIS. All Rights Reserved

Name Elements	xNL Elements (XML Tags)	Description
		NameType: Defines the type of name and is optional. Example: Nickname, pet name, etc. *Code:* Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services
Generation Identifier	GenerationIdentifier	This element is a sub-element of "PersonName" element and defines the generation identifier of a person. This element can occur multiple times and is optional (0 or more). Example: JNR, III, THE THIRD, etc. This element provides the following attributes: *Type*: Defines the type for generation identifier and is optional. Example: Family Title *Code:* Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services
Suffix	Suffix	This element is a sub-element of "PersonName" element and defines the suffixes associated with a person's name. This element can occur multiple times and is optional (0 or more). Example: PhD, VC, QC, etc. This element provides the following attributes: *Type*: Defines the type for suffix and is optional. *Code:* Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services
General Suffix	GeneralSuffix	This element is a sub-element of "PersonName" element and defines the general suffixes associated with a Person's name. This element can occur once and is optional (0 or 1). Example: Deceased, retired, etc. This element provides the following attributes: *Type*: Defines the type for general suffix and is optional. *Code:* Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services
Other known names	**KnownAs**	This element is a sub-element of "PersonName" element and is used to define a person's other known names in detail. This element can occur once and is optional (0 or 1). See section "KnownAs Element" for further details. Can occur once and is optional.

7.4.1 Example

His Excellency Professor Ram Kumar JNR I, PhD

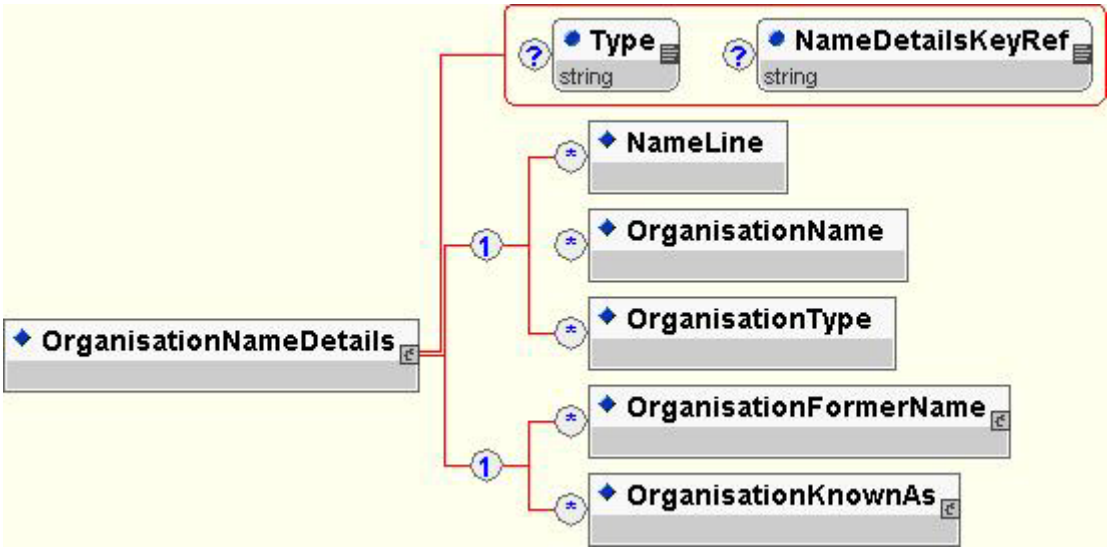
```
<xNL>
  <NameDetails PartyType="Person">
   <PersonName>
    <PrecedingTitle>His Excellency</PrecedingTitle>
    <Title>Professor</Title>
    <FirstName NameType="Given Name">Ram</FirstName>
    <LastName NameType="Family Name">Kumar</LastName>
    <OtherName NameType="Full Name">Ramkumar Venkatachalam</OtherName>
    <GenerationIdentifier>JNR I</GenerationIdentifier>
    <Suffix>PhD</Suffix>
   </PersonName>
  </NameDetails>
```

Copyright© OASIS. All Rights Reserved

```
</xNL>
```

7.5 OrganisationNameDetails Element

OrganisationNameDetails element defines an organisation's name in detail by breaking it down into sub-elements.



Name Elements	xNL Elements (XML Tags)	Description
Name of the organisation in detail	**OrganisationNameDetails**	This is the sub-element of root element "NameDetails". This element can occur once and is optional (0 or 1). This element is a container and has sub-elements to define the name of an organisation in detail. This element provides the following attributes: *NameDetailsKeyRef*: Reference to another "NameDetails" element with no foreign key reinforcement. The referenced element may be out of the document and the document is still valid. This attribute is optional. *Code:* Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services.
Name details of an organisation as a free format text	NameLine	This element is used to define the name of the organisation as a free format text field without breaking it into sub-elements. This element can occur multiple times and is optional (0 or more). See Section "NameLine Element" for more details.
Name of the organisation	OrganisationName	Sub-element of "OrganisationNameDetails" element and is used to define the name of the organisation. This element can occur many times and is optional (0 or more). Example: MSI Business Solutions. This element provides the following attributes: *Type:* Defines the Type of Organisation name and is optional. Example: Official, Legal, Un-official, etc *NameType*: Defines the name type of the Organisation name and is optional. Example: Former name, new name, abbreviated name etc.

Copyright© OASIS. All Rights Reserved

Name Elements	xNL Elements (XML Tags)	Description
		Code: Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services.
Type for Organisation	OrganisationType	Sub-element of "OrganisationNameDetails" element and is used to define the type for the organisation. This element can occur many times and is optional (0 or more). Defines the type for organisation. Example: "Pty.LTD" in "MSI Business Solutions Pty.LTD". Other examples: GmbH, Inc, Ltd, AB, A/S, OY. Note that one can also use OrganisationName element itself to define the name and type also. Example: <OrganisationName>MSI Business Solutions Pty. Ltd </OrganisationName> This element provides the following attributes: *Type:* Can be used to define the type and is optional. Example: Abbreviation, Legal Type, etc. *NameType:* Can be used to define the name type and is optional. Example: Official, unofficial, etc. *Code:* Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services
Former name of the organisation	**OrganisationFormerName**	See section "OrganisationFormerName Element" for further details. Can occur once and is optional.
Other name of the organisation or also known as	**OrganisationKnownAs**	See section "OrganisationKnownAs Element" for further details. Can occur once and is optional.

7.5.1 Example

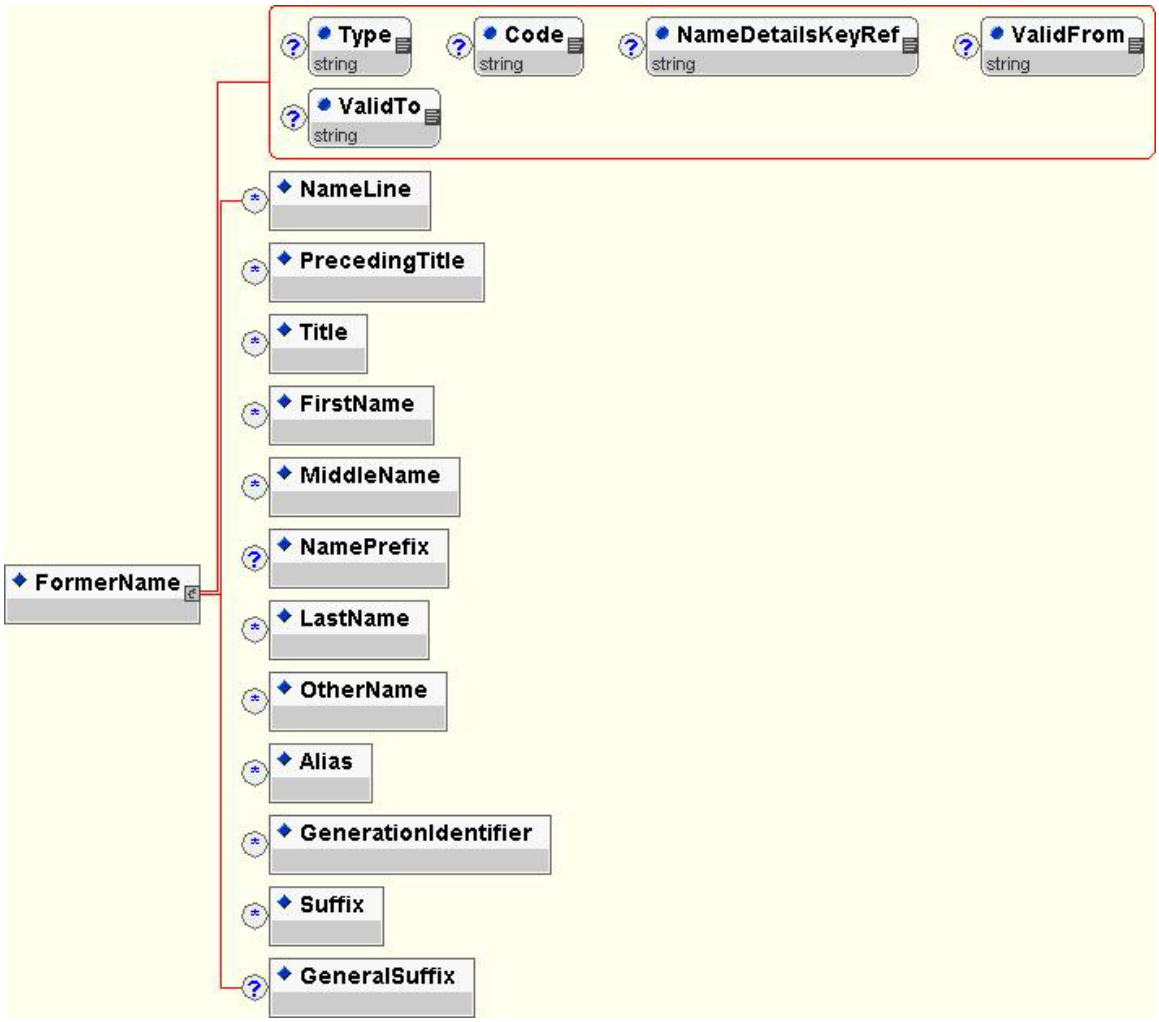
MSI Business Solutions Pty.Ltd

```
<xNL>
   <NameDetails PartyType="Organisation">
      <OrganisationNameDetails>
         <OrganisationName Type="Software Vendor">
              MSI Business Solutions
         </OrganisationName>
         <OrganisationType>Pty.Ltd</OrganisationType>
     </OrganisationNameDetails>
  </NameDetails>
</xNL>
```

Copyright© OASIS. All Rights Reserved

7.6 FormerName Element

FormerName element defines the former name of a person in detail by breaking it down into sub-elements.



Name Elements	xNL Elements (XML Tags)	Description
Former Name of the Person	**FormerName**	This element is the sub-element of root element "NameDetails". This element can occur once and is optional (0 or 1). This element is a container and has sub-elements to define the former name of a person in detail. This element provides the following attributes: *Type:* defines the type of name and is optional. Example: full name, etc. *NameDetailsKeyRef*: Reference to another NameDetails element with no foreign key reinforcement. The referenced element may be out of the document and the document is still valid. This attribute is optional *Code:* Some postal services use a special code to define the element and is

Copyright© OASIS. All Rights Reserved

Name Elements	xNL Elements (XML Tags)	Description
		optional. Example: ECCMA Code Tables for postal services. *ValidFrom*: Defines the date this name was valid from and is optional. *ValidTo:* Defines the date this name was valid to and is optional.
Name details of a person as a free format text	NameLine	This element is used to define the name of the person as a free format text field without breaking it into sub-elements. This element can occur multiple times and is optional (0 or more). See Section "NameLine Element" for more details.
Preceding Title	PrecedingTitle	This element can occur many times and is optional (0 or more) and defines Preceding titles. Example: HIS EXCELLENCY, HONOURABLE, etc. This element provides the following attributes: *Type:* Defines the type of preceding title and is optional. Example: Honorary title *Code:* Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services.
Title	Title	This element can occur many times and is optional (0 or more) and defines titles. This element provides the following attributes: *Type:* Defines the type of title and is optional. Example: Sex, Honorary, Profession, degree, greetings title, etc. *Code:* Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services
First Name	FirstName	This element defines first name of a person that is the position of the name in the name string. This element can occur multiple times and is optional (0 or more). This element provides the following attributes: *Type*: Defines the type for first name and is optional. Example: Old Name, Official, UnOfficial, Initials, etc. Some countries have two first names, one is official name and the other is an unofficial name. *NameType*: Defines the type of name and is optional. Example: Given Name, Christian Name, First Name, Surname, family name, etc. *Code:* Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services
Middle Name	MiddleName	This element defines middle name of a person. This name is the position of the name in the name string. This element can occur multiple times and is optional (0 or more). This element provides the following attributes: *Type*: Defines the type for middle name and is optional. Example: Old Name, Official, UnOfficial, Initials, etc. *NameType*: Defines the type of name and is optional. Example: Given Name, Christian Name, First Name, etc. *Code:* Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services
Name prefix	NamePrefix	This element defines the prefix of a name and can occur once and is optional (0 or 1). Example: van de, de la, etc. as in France and The Netherlands. This element provides the following attributes: *Type*: Defines the type NamePrefix and is optional. Example: Official, UnOfficial, etc. *NameType*: Defines the type of name that has the name prefix and is optional.

Copyright© OASIS. All Rights Reserved

Name Elements	xNL Elements (XML Tags)	Description
		Example: Last Name, SurName, Family Name, etc. *Code:* Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services
Last Name	LastName	This element defines the last name of a person. This element is the position of the name in the name string. This element can occur multiple times and is optional (0 or more). This element provides the following attributes: *Type*: Defines the type of last name and is optional. Example: Old Name, Official, UnOfficial, Initials, etc. *NameType*: Defines the type of name and is optional. Example: Last Name, SurName, Family Name, Father's name (as in India), Given name, Christian name etc. *Code:* Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services
Other Name	OtherName	This element defines the other names of a person. This element can occur multiple times and is optional (0 or more). Example: Yousuf Khan al Hatab al Sayad. This element provides the following attributes: *Type*: Defines the type for other name and is optional. Example: Old Name, Official, UnOfficial, etc. *NameType*: Defines the type of name and is optional. Example: Maiden Name, Patronymic name, Matronymic name, etc. *Code:* Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services
Alias	Alias	This element defines alias names of a person. This element can occur multiple times and is optional (0 or more). This element provides the following attributes: *Type*: Defines the type for other name and is optional. Example: Old Name, Official, UnOfficial, etc. *NameType*: Defines the type of name and is optional. Example: Nickname, pet name, etc. *Code:* Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services
Generation Identifier	GenerationIdentifier	This element defines the generation identifier of a person. This element can occur multiple times and is optional (0 or more). Example: JNR, III, THE THIRD, etc. This element provides the following attributes: *Type*: Defines the type for generation identifier and is optional. Example: Family Title *Code:* Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services
Suffix	Suffix	This element defines the suffixes associated with a person's name. This element can occur multiple times and is optional (0 or more). Example: PhD, VC, QC, etc. This element provides the following attributes: *Type*: Defines the type for suffix and is optional. *Code:* Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services
General Suffix	GeneralSuffix	This element defines the general suffixes associated with a Person's name. This element can occur once and is optional (0 or 1). Example: Deceased,

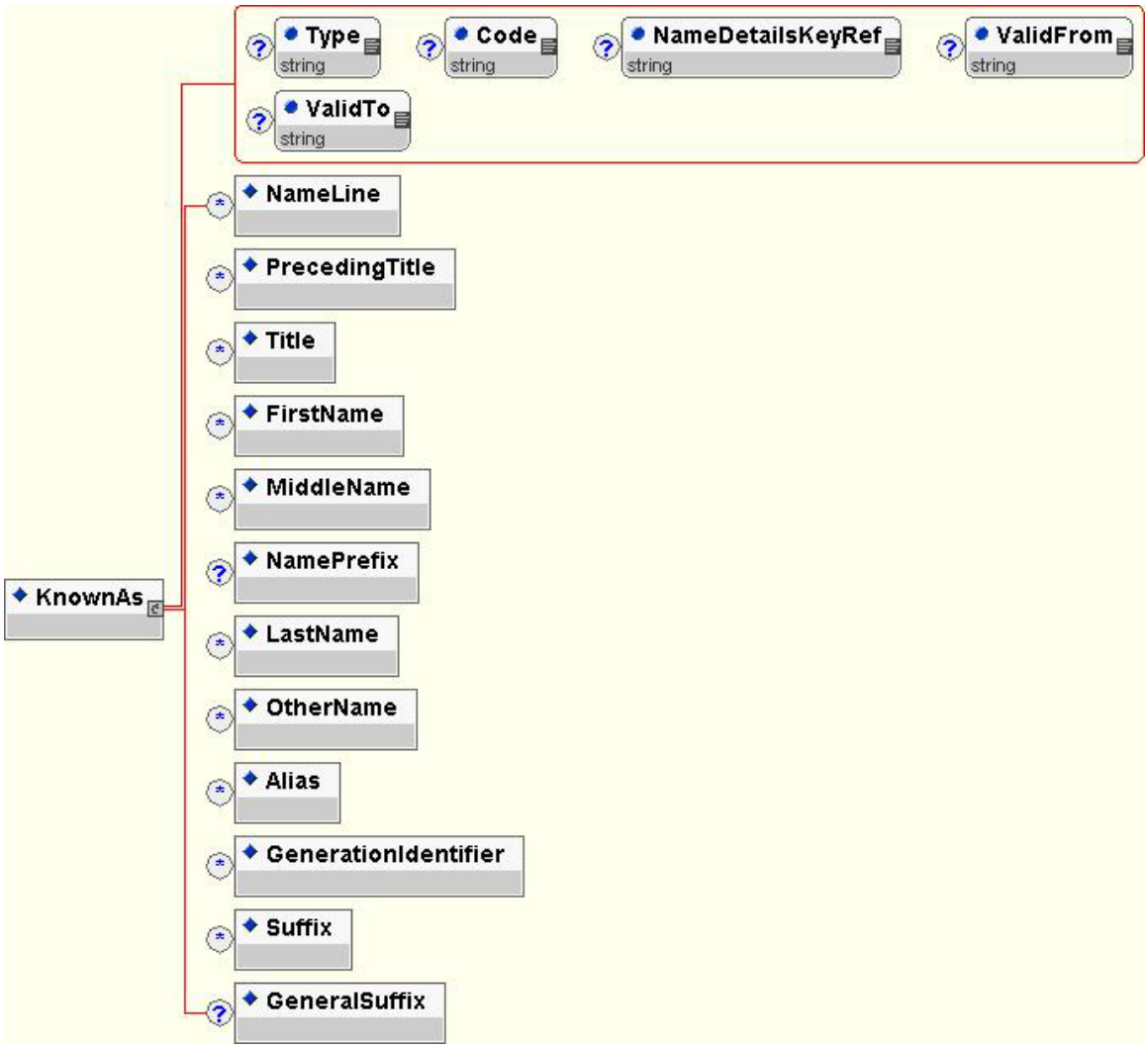
Copyright© OASIS. All Rights Reserved

Name Elements	xNL Elements (XML Tags)	Description
		retired, etc. This element provides the following attributes: *Type*: Defines the type for general suffix and is optional. *Code:* Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services

Copyright© OASIS. All Rights Reserved

7.7 KnownAs Element

KnownAs element defines the other known names of a person in detail by breaking it down into sub-elements.



Copyright© OASIS. All Rights Reserved

Name Elements	xNL Elements (XML Tags)	Description
Other known names of a Person	**KnownAs**	This element is the sub-element of root element "NameDetails". This element can occur once and is optional (0 or 1). This element is a container and has sub-elements to define the other known names of a person in detail. This element provides the following attributes: *Type:* defines the type of name and is optional. Example: full name, etc. *NameDetailsKeyRef:* Reference to another NameDetails element with no foreign key reinforcement. The referenced element may be out of the document and the document is still valid. This attribute is optional *Code:* Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services. *ValidFrom*: Defines the date this name was valid from and is optional. *ValidTo:* Defines the date this name was valid to and is optional.
Name details of a person as a free format text	NameLine	This element is used to define the name of the person as a free format text field without breaking it into sub-elements. This element can occur multiple times and is optional (0 or more). See Section "NameLine Element" for more details.
Preceding Title	PrecedingTitle	This element can occur many times and is optional (0 or more). This element defines Preceding titles. Example: HIS EXCELLENCY, HONOURABLE, etc. This element provides the following attributes: *Type:* Defines the type of preceding title and is optional. Example: Honorary title *Code:* Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services.
Title	Title	This element can occur many times and is optional (0 or more). This element defines titles. This element provides the following attributes: *Type:* Defines the type of title and is optional. Example: Sex, Honorary, Profession, degree, greetings title, etc. *Code:* Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services
First Name	FirstName	This element defines first name of a person that is the position of the name in the name string. This element can occur multiple times and is optional (0 or more). This element provides the following attributes: *Type*: Defines the type for first name and is optional. Example: Old Name, Official, UnOfficial, Initials, etc. Some countries have two first names, one is official name and the other is an unofficial name. *NameType*: Defines the type of name and is optional. Example: Given Name, Christian Name, First Name, Surname, family name, etc. *Code:* Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services
Middle Name	MiddleName	This element defines middle name of a person. This name is the position of the name in the name string. This element can occur multiple times and is optional (0 or more). This element provides the following attributes: *Type*: Defines the type for middle name and is optional. Example: Old Name,

Copyright© OASIS. All Rights Reserved

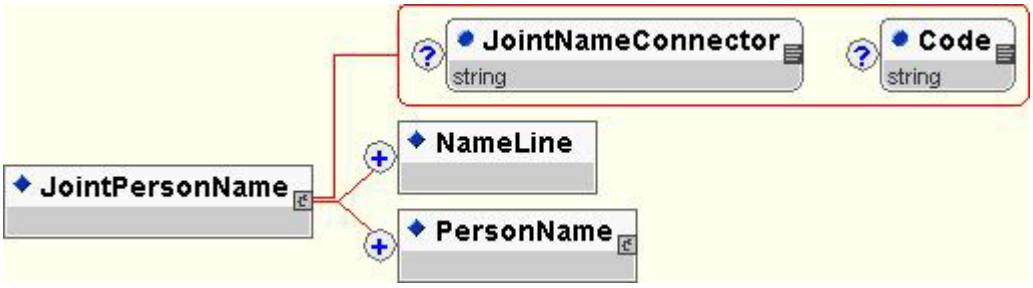
Name Elements	xNL Elements (XML Tags)	Description
		Official, UnOfficial, Initials, etc. *NameType*: Defines the type of name and is optional. Example: Given Name, Christian Name, First Name, etc. *Code:* Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services
Name prefix	NamePrefix	This element defines the prefix of a name and can occur once and is optional. Example: van de, de la, etc. as in France and The Netherlands. This element provides the following attributes: *Type*: Defines the type NamePrefix and is optional. Example: Official, UnOfficial, etc. *NameType*: Defines the type of name that has the name prefix and is optional. Example: Last Name, SurName, Family Name, etc. *Code:* Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services
Last Name	LastName	This element defines the last name of a person. This element is the position of the name in the name string. This element can occur multiple times and is optional (0 or more). This element provides the following attributes: *Type*: Defines the type of last name and is optional. Example: Old Name, Official, UnOfficial, Initials, etc. *NameType*: Defines the type of name and is optional. Example: Last Name, SurName, Family Name, Father's name (as in India), Given name, Christian name etc. *Code:* Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services
Other Name	OtherName	This element defines the other names of a person. This element can occur multiple times and is optional (0 or more). Example: Yousuf Khan al Hatab al Sayad is the other name and is normally known as Yousuf Khan. Hatab is Yousuf's father's name and Sayad is Yousuf's grandfather's name. This element provides the following attributes: *Type*: Defines the type for other name and is optional. Example: Old Name, Official, UnOfficial, etc. *NameType*: Defines the type of name and is optional. Example: Maiden Name, Patronymic name, Matronymic name, etc. *Code:* Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services
Alias	Alias	This element defines alias names of a person. This element can occur multiple times and is optional (0 or more). This element provides the following attributes: *Type*: Defines the type for other name and is optional. Example: Old Name, Official, UnOfficial, etc. *NameType*: Defines the type of name and is optional. Example: Nickname, pet name, etc. *Code:* Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services
Generation Identifier	GenerationIdentifier	This element defines the generation identifier of a person. This element can occur multiple times and is optional (0 or more). Example: JNR, III, THE

Copyright© OASIS. All Rights Reserved

Name Elements	xNL Elements (XML Tags)	Description
		THIRD, etc. This element provides the following attributes: *Type*: Defines the type for generation identifier and is optional. Example: Family Title *Code:* Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services
Suffix	Suffix	This element defines the suffixes associated with a person's name. This element can occur multiple times and is optional (0 or more). Example: PhD, VC, QC, etc. Has attributes: *Type*: Defines the type for suffix and is optional. *Code:* Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services
General Suffix	GeneralSuffix	This element defines the general suffixes associated with a Person's name. This element can occur once and is optional. Example: Deceased, retired, etc. This element provides the following attributes: *Type*: Defines the type for general suffix and is optional. *Code:* Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services

7.8 JointPersonName Element

JointPersonNameDetails element defines joint person names in detail by breaking it down into sub-elements.



Name Elements	xNL Elements (XML Tags)	Description
Joint name of persons	**JointPersonName**	This element is the sub-element of root element "NameDetails". This element can occur once and is optional (0 or 1). This element is a container and has sub-elements to define joint names in detail. This element provides the following attributes: *JointNameConnector:* Defines the connector for the joint names and is optional. Example: "and" in Mrs Johnson and Mrs.Johnson. *Code:* Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services.
Name details	NameLine	This element is used to define the name of the person as a free format text

Copyright© OASIS. All Rights Reserved

Name Elements	xNL Elements (XML Tags)	Description
of joint person as a free format text		field without breaking it into sub-elements. Can occur multiple times and is optional (0 or more). See Section "NameLine Element" for more details.
Name of person	**PersonName**	This element is a sub-element of "JointNameDetails" element. This element defines the name of person. This element can occur many times and must occur at least once (1 or more). See section "PersonName" element for further details about this element.

Copyright© OASIS. All Rights Reserved

7.8.1 Example

Mrs Joanne Johnshon and Mr.Peter Johnson

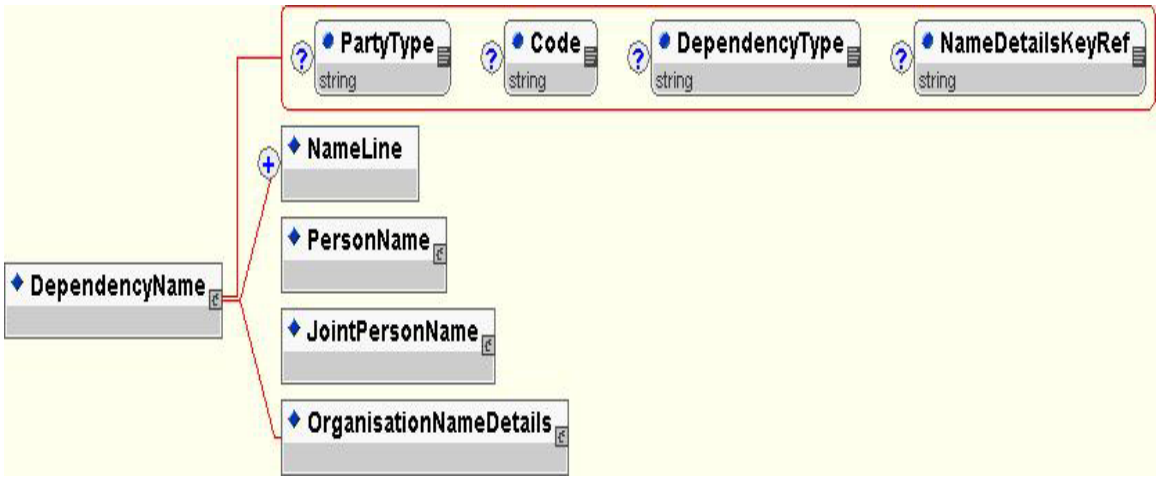
```
<xNL>
   <NameDetails PartyType="Person">
      <JointPersonName>
       <NameLine>Mrs. Joanne Johnson and Mr. Peter Johnson</NameLine>
      </JointPersonName>
   </NameDetails>
</xNL>
```

OR

```
<xNL>
    <NameDetails PartyType="Person">
      <JointPersonName JointNameConnector="AND">
         <PersonName>
            <Title>Mrs</Title>
            <FirstName>Joanne</FirstName>
            <LastName>Johnson</LastName>
         </PersonName>
         <PersonName>
            <Title>Mr</Title>
            <FirstName>Peter</FirstName>
            <LastName>Johnson</LastName>
         </PersonName>
      </JointPersonName>
   </NameDetails>
</xNL>
```

7.9 DependencyName Element

DependencyName element defines dependent name in detail by breaking it down into sub-elements.



Copyright© OASIS. All Rights Reserved

Name Elements	xNL Elements (XML Tags)	Description
Dependency Name Details	**DependencyName**	This element is the sub-element of "NameDetails" element. This element can occur once and is optional. This element is a container to define the dependent name of customer. Example: Ram Kumar, Care of MSI Business Solutions. This element provides the following attributes: *PartyType:* Defines the type of customer/party and is optional. Example, Club, Organisation, Person, etc. *DependencyType:* Defines the type of dependency and is optional. For example, Care of, son of, father of, etc. *NameDetailsKeyRef:* Reference to another NameDetails element with no foreign key reinforcement. The referenced element may be out of the document and the document is still valid. This attribute is optional. *Code:* Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services.
Name as a general free format text field	NameLine	This element is used to define the name of the person as a free format text field without breaking it into sub-elements. This element can occur multiple times and is optional (0 or more). See Section "NameLine Element" for more details.
Name of the person	**PersonName**	This element is used to define the name of the dependency person. See section "PersonName Element" for further details. This element can occur once and is optional (0 or 1).
Name of the organisation	**OrganisationNameDetails**	This element is used to define the name of the dependency organisation. See section "OrganisationNameDetails Element" for further details. This element can occur once and is optional (0 or 1).
More than one person	**JointPersonName**	This element is used to define the joint names of the people who are the dependency entities of a person. See section "JointPersonName Element" for further details. This element can occur once and is optional (0 or 1).

7.9.1 Example

Mr. Ram Laxhman B Kumar
C/O Mr. Venkat (Venki) Krishnan

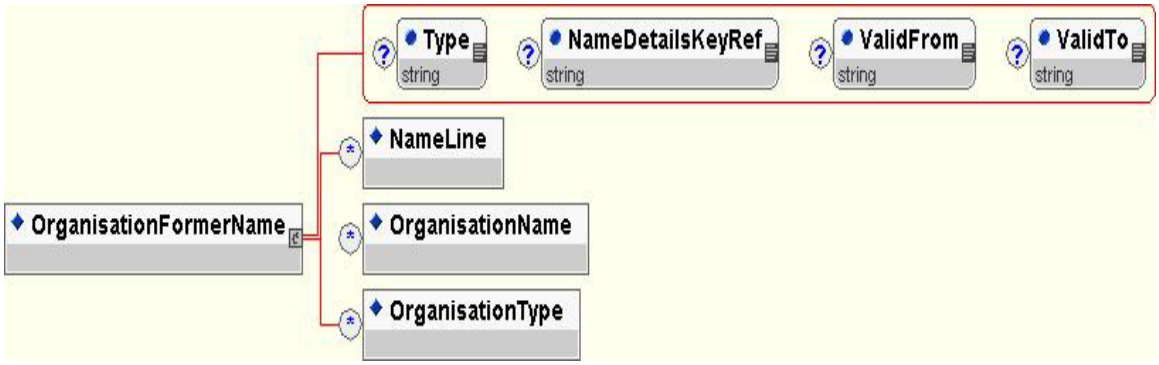
```
<xNL>
   <NameDetails  PartyType="Person">
     <PersonName>
        <Title>Mr</Title>
        <FirstName Type="GivenName">Ram</FirstName>
        <MiddleName>Laxhman</MiddleName>
        <MiddleName Type="Initial">B</MiddleName>
        <LastName NameType="SurName">Kumar</LastName>
     </PersonName>
     <DependencyName PartyType="Person"
                     DependencyType="C/O">
        <PersonName>
           <Title>Mr</Title>
           <FirstName Type="Official"
                      NameType="GivenName">Venkat</FirstName>
```

Copyright© OASIS. All Rights Reserved

```
            <FirstName Type="Unofficial"
                       NameType="GivenName">Venki</FirstName>
            <LastName>Krishnan</LastName>
          </PersonName>
        </DependencyName>
    </NameDetails>
</xNL>
```

7.10 OrganisationFormerName Element

OrganisationFormerName element defines former name of an organisation in detail by breaking it down into sub-elements.



Name Elements	xNL Elements (XML Tags)	Description
Former Name of the organisation in detail	**OrganisationFormerName**	This element is the sub-element of the element "OrganisationNameDetails". This element can occur multiple times and is optional (0 or 1). This element is a container and has sub-elements to define the former name of an organisation in detail. This element provides the following attributes: *NameDetailsKeyRef:* Reference to another NameDetails element with no foreign key reinforcement. The referenced element may be out of the document and the document is still valid. This attribute is optional. *Code:* Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services. *ValidFrom*: Defines the date this name was valid from and is optional. *ValidTo:* Defines the date this name was valid to and is optional.
Name details of an organisation as a free format text	NameLine	This element is used to define the former name of the organisation as a free format text field without breaking it into sub-elements. This element can occur multiple times and is optional (0 or more). See Section "NameLine Element" for more details.
Name of the organisation	OrganisationName	This element is a sub-element of "OrganisationNameDetails" element and is used to define the name of the organisation. This element can occur multiple times and is optional (0 or more). Example: MSI Business Solutions. This element provides the following attributes:

Copyright© OASIS. All Rights Reserved

Name Elements	xNL Elements (XML Tags)	Description
		Type: Defines the Type of Organisation name and is optional. Example: Official, Legal, Un-official, etc *NameType*: Defines the name type of the Organisation name and is optional. Example: Former name, new name, abbreviated name etc. *Code:* Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services.
Type for Organisation	OrganisationType	This element is a sub-element of "OrganisationNameDetails" element and is used to define the type for the organisation. This element can occur many times and is optional (0 or more). Example: "Pty.LTD" in "MSI Business Solutions Pty.LTD". Other examples: GmbH, Inc, Ltd, AB, A/S, OY. Note that one can also use OrganisationName element itself to define the name and type also. Example: <OrganisationName>MSI Business Solutions Pty. Ltd</OrganisationName> This element provides the following attributes: *Type:* Can be used to define the type and is optional. Example: Abbreviation, Legal Type, etc. *NameType:* Can be used to define the name type and is optional. Example: Official, unofficial, etc. *Code:* Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services

7.10.1 Example

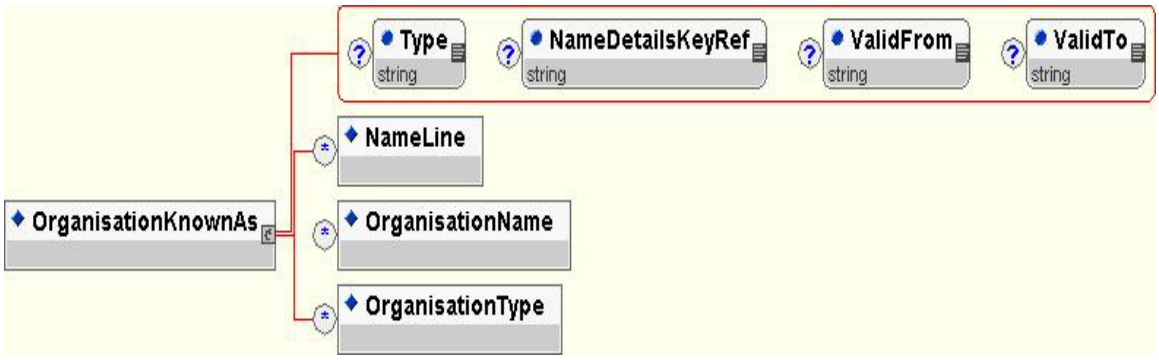
MSI Business Solutions Pty. Ltd
Formerly Known as MasterSoft International Pty. Ltd (valid from June 1989 to September 30, 2001)

```
<xNL>
   <NameDetails PartyType="Organisation">
     <OrganisationNameDetails>
       <OrganisationName>MSI Business Solutions</OrganisationName>
       <OrganisationType>Pty.Ltd</OrganisationType>
       <OrganisationFormerName  ValidFrom="June 1989"
                                ValidTo="September 30, 2001">
          <OrganisationName Type="Software Vendor">
              MasterSoft International
          </OrganisationName>
          <OrganisationType>Pty.Ltd</OrganisationType>
       </OrganisationFormerName>
    </OrganisationNameDetails>
  </NameDetails>
</xNL>
```

Copyright© OASIS. All Rights Reserved

7.11 OrganisationKnownAs Element

OrganisationKnownAs element defines the other known names of an organisation in detail by breaking it down into sub-elements.



Name Elements	xNL Elements (XML Tags)	Description
Other known names of the organisation in detail	**OrganisationKnownAs**	This element is the sub-element of the element "OrganisationNameDetails". This element can occur multiple times and is optional (0 or 1). This element is a container and has sub-elements to define the other known names of an organisation in detail. This element provides the following attributes: *NameDetailsKeyRef:* Reference to another NameDetails element with no foreign key reinforcement. The referenced element may be out of the document and the document is still valid. This attribute is optional. *Code:* Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services. *ValidFrom*: Defines the date this name was valid from and is optional. *ValidTo:* Defines the date this name was valid to and is optional.
Name details of an organisation as a free format text	NameLine	This element is used to define the former name of the organisation as a free format text field without breaking it into sub-elements. This element can occur multiple times and is optional (0 or more). See Section"NameLine Element" for more details.
Name of the organisation	OrganisationName	This element is a sub-element of "OrganisationNameDetails" element and is used to define the name of the organisation. This element can occur many times and is optional (0 or more). Example: MSI Business Solutions. This element provides the following attributes: *Type:* Defines the Type of Organisation name and is optional. Example: Official, Legal, Un-official, etc *NameType*: Defines the name type of the Organisation name and is optional. Example: Former name, new name, abbreviated name etc. *Code:* Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services.
Type for Organisation	OrganisationType	This element is a sub-element of "OrganisationNameDetails" element and is used to define the type for the organisation. This element can occur many

Copyright© OASIS. All Rights Reserved

Name Elements	xNL Elements (XML Tags)	Description
		times and is optional (0 or more). Defines the type for organisation. Example: "Pty.LTD" in "MSI Business Solutions Pty.LTD". Other examples: GmbH, Inc, Ltd, AB, A/S, OY. Note that one can also use OrganisationName element itself to define the name and type also. Example: <OrganisationName>MSI Business Solutions Pty. Ltd</OrganisationName> This element provides the following attributes: *Type:* Can be used to define the type and is optional. Example: Abbreviation, Legal Type, etc. *NameType:* Can be used to define the name type and is optional. Example: Official, unofficial, etc. *Code:* Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services

7.11.1 Example

MSI Business Solutions Pty. Ltd
Formerly Known as MasterSoft International Pty. Ltd (valid from June 1989 to September 30, 2001)
Also known as MSI

```
<xNL>
   <NameDetails PartyType="Organisation">
     <OrganisationNameDetails>
      <OrganisationName Type="IT Consulting">
          MSI Business Solutions
      </OrganisationName>
      <OrganisationType>Pty.Ltd</OrganisationType>
      <OrganisationFormerName  ValidFrom="June 1989"
                               ValidTo="September 30, 2001">
        <OrganisationName Type="Software Vendor">
          MasterSoft International
        </OrganisationName>
        <OrganisationType>Pty.Ltd</OrganisationType>
      </OrganisationFormerName>
      <OrganisationKnownAs>
          <NameLine>MSI</NameLine>
      </OrganisationKnownAs>
    </OrganisationNameDetails>
  </NameDetails>
</xNL>
```

Copyright© OASIS. All Rights Reserved

8.0 References

- Name and Address Markup Language (NAML) Specifications document (Version 1-1.3), MasterSoft International, April 2000
- xNAL Specifications Document for W3C DTD/Schema Version 2.0, OASIS CIQ TC, http://www.oasis-open.org/committees/ciq, May 2002
- xAL Specifications Document for W3C DTD/Schema Version 2.0, OASIS CIQ TC, http://www.oasis-open.org/committees/ciq, May 2002
- Global Address Specifications document (Version 1-1.2), December 2000
- Ram Kumar, XML Standards for Customer Information Quality Management, XML Journal, Vol.1, No.2, July 2000, pp.41-45.
- Graham Rhind, "The Global Source Book for Address Data Management, 1998
- The Universal Name and Address Format (UNA), MasterSoft International, 1992.
- 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, May 7, 2002

Copyright© OASIS. All Rights Reserved