



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

Version 2.0

A Standard from the Customer Information Quality Technical Committee

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

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	7
2.1	EXTENSIBLE NAME AND ADDRESS LANGUAGE	7
2.2	THE GOAL OF xNAL	7
3.0	THE OBJECTIVE AND SCOPE	8
4.0	EXTENSIBLE NAME LANGUAGE (XNL).....	8
4.1	XNL	8
4.2	THE GOAL	8
4.3	THE CHALLENGE	8
4.4	WHAT DOES XNL NOT REPRESENT	9
5.0	USING THE XNL DTD/SCHEMA.....	9
5.1	PURPOSE OF THE XML DTD/SCHEMA FOR NAMES	9
5.2	FLEXIBILITY	9
5.2.1	Example	9
5.3	DON'T GET CONFUSED – KEEP IT SIMPLE	11
5.4	NAMESPACES AND VERSIONS	11
5.5	XML SCHEMA: EXTENSIBILITY	11
5.6	XML SCHEMA: DOCUMENT FRAGMENTS	12
5.7	DEEP NESTING VS. FLAT STRUCTURE.....	12
5.8	WHERE TO START	13
5.9	COMPATIBILITY BETWEEN DTD AND SCHEMA.....	14
5.10	DOCUMENT EXCHANGE BETWEEN DIFFERENT PARTIES	14
6.0	DEFINING NAMES IN XNL	14
6.1	RESTRICTIONS	14
7.0	XNL GRAMMAR.....	16
7.1	XNL ELEMENT	16
7.2	NAMELINE ELEMENT	17
7.3	NAMEDetails ELEMENT	18
7.3.1	Example	20
7.4	PERSONNAME ELEMENT	21
7.4.1	Example	24
7.5	ORGANISATIONNAMEDETAILS ELEMENT.....	25
7.5.1	Example	26
7.6	FORMERNAME ELEMENT	27
7.7	KNOWNAS ELEMENT.....	30
7.8	JOINTPERSONNAME ELEMENT	33
7.8.1	Example	34
7.9	DEPENDENCYNAME ELEMENT	34
7.9.1	Example	35
7.10	ORGANISATIONFORMERNAME ELEMENT	36
7.10.1	Example	37
7.11	ORGANISATIONKNOWNAS ELEMENT	38
7.11.1	Example	39

8.0	REFERENCES	40
------------	-------------------------	-----------

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 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 Rind, Consultant, International Address Databases, for his feedback/input on xNL and xAL standards and for his permission to use some of 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, Inc, USA
Mr. John Bennett	Parlo.com, USA
Mr. Joe Lubenow	Individual member of OASIS, USA
Mr. Nikolaj Nyholm	Ascio Technologies, Norway
Mr. Marcus Goncalves	Individual member of OASIS, USA
Mr. Mark Meadows	Microsoft, USA
Mr. Robert James	Individual member of OASIS, U.K
Mr. Max Voskob	MSI Business Soilutions, New Zealand

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 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.

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.

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 Using the xNL DTD/Schema

5.1 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.

5.2 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.

5.2.1 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>
```

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

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

5.3 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.

5.4 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.

5.5 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

5.6 XML Schema: Document Fragments

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

5.7 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>
```

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.

5.8 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.

5.9 Compatibility between DTD and Schema

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

5.10 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).

6.0 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.

6.1 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.

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.

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 *##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 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.

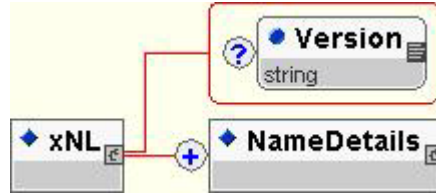
For ease of reading, under XML Elements column in the tables in the following sections, XML Tag names in **bold** are XML Containers (consisting of sub-XML elements), XML Tags in regular text are XML Elements and Tag names in *italics* in the Description column of the tables are Attributes of XML elements.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

```
<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.

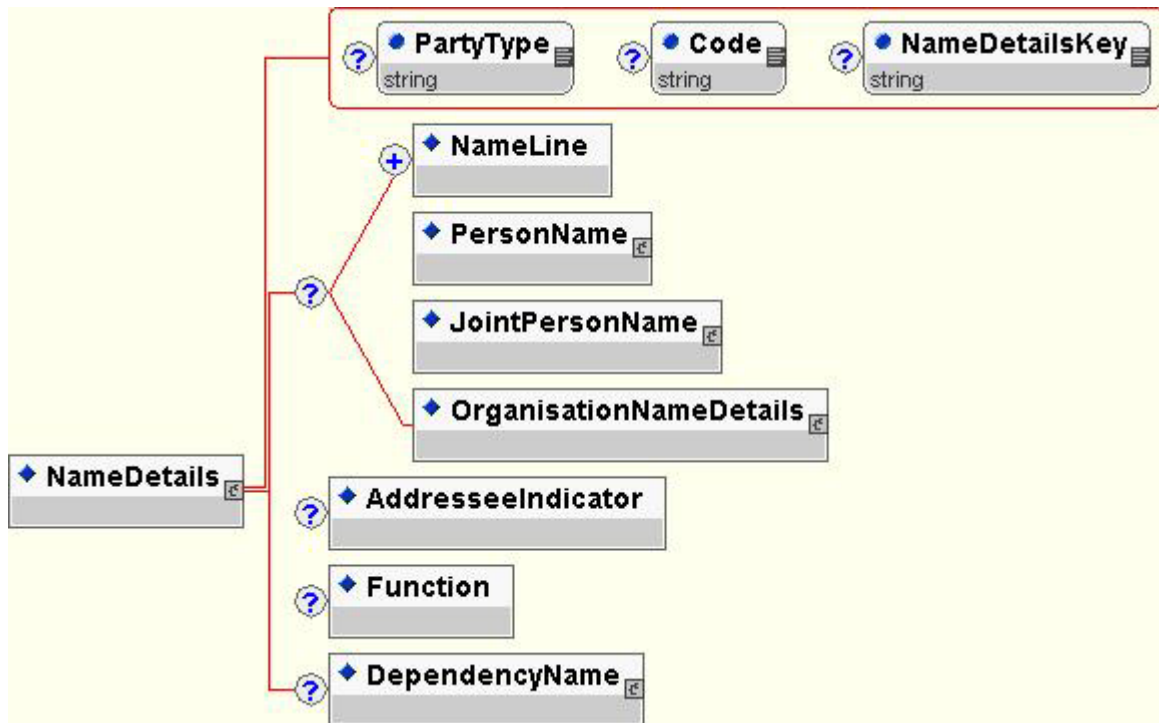
“NameType” attribute: Defines the meaning of the element. Example: First Name can be Christian name, Given name, first name, etc.

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

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.

Name Elements	xNL Elements (XML Tags)	Description
Name Details	NameDetails	<p>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:</p> <p><i>PartyType</i>: Defines the type of customer/party and is optional. Example, Club, Organisation, Person, etc.</p> <p><i>Code</i>: Some postal services use a special code to define the element. Example: ECCMA Code Tables for postal services.</p> <p><i>NameDetailsKey</i>: 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</p> <p>Example:</p> <pre><xNL> <NameDetails PartyType="Person"> <PersonName> </PersonName> </NameDetails> </xNL></pre>
Description of the Addressee	AddresseeIndicator	<p>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. Can occur once. Example: ATTENTION To, ter Attentie van (in Holland), etc. Has an attribute:</p> <p><i>Code</i>: Some postal services use a special code to define the element. Example: ECCMA Code Tables for postal services.</p>
Name as a general free format text field	NameLine	<p>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. Can occur multiple times and is optional (0 or more). See Section “NameLine Element” for more details.</p>
Name of the person	PersonName	<p>This element is a sub-element of “NameDetails” element and is used to define a person’s name in detail. Can occur once and is optional. See section “PersonName Element” for further details. Can occur once and is optional.</p>
Name of the organisation	OrganisationNameDetails	<p>This element is a sub-element of “NameDetails” element and is used to define an organisation’s name in detail. Can occur once and is optional. See section “PersonName Element” for further details. Can occur once and is optional.</p>
Function/Position/Role	Function	<p>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. Has an attribute:</p> <p><i>Code</i>: Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services.</p>
More than one person	JointPersonName	<p>This element is a sub-element of “NameDetails” element and is used to define a joint person’s name in detail. Can occur once and is optional. See</p>

Name Elements	xNL Elements (XML Tags)	Description
		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. Can occur once and is optional. See section “DependencyName Element” for further details. Can occur once and is optional.

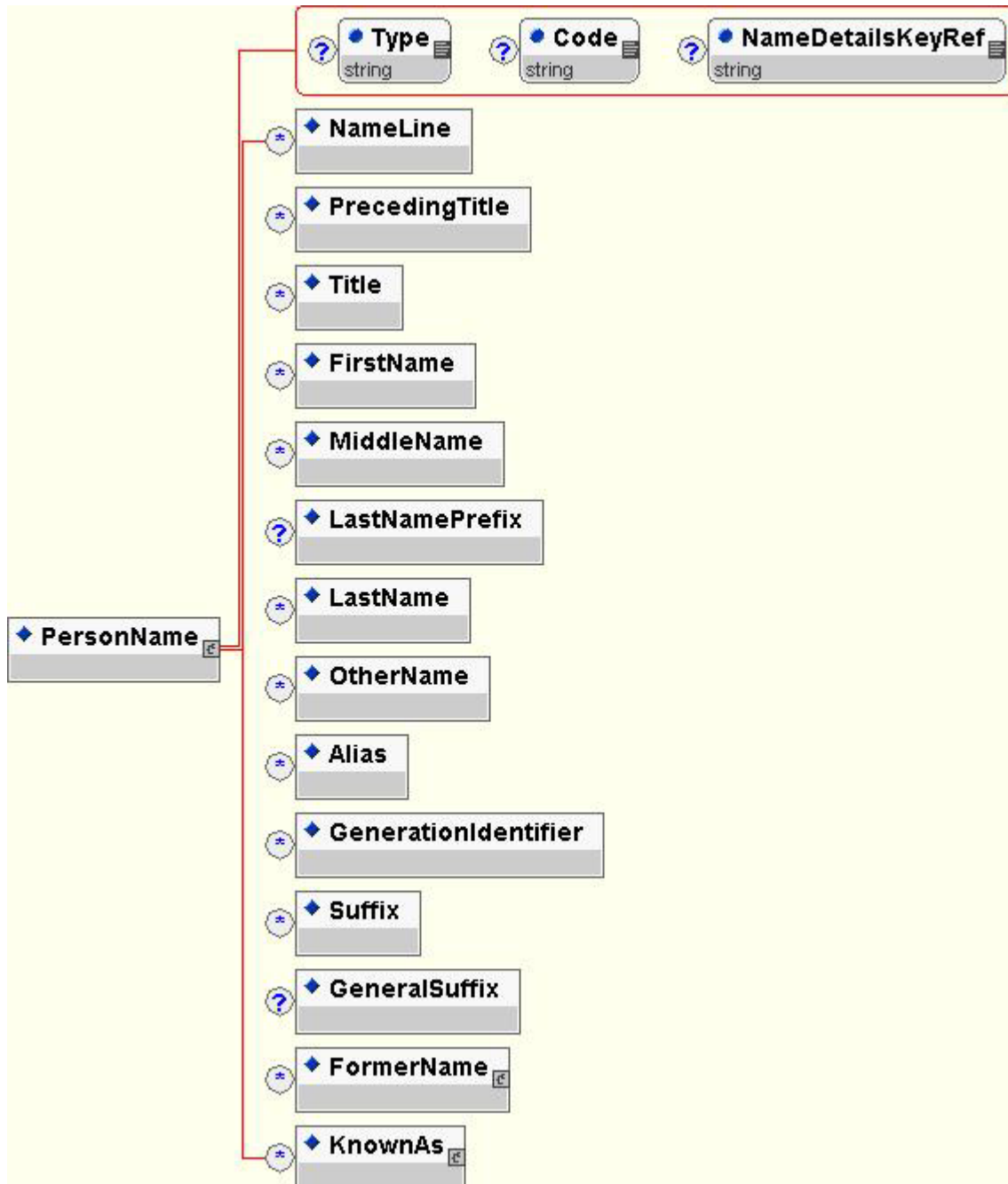
7.3.1 Example

**ATTENTION : Mr. Ram Kumar
CEO**

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

7.4 PersonName Element

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



Name Elements	xNL Elements (XML Tags)	Description
Name of the Person	PersonName	<p>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. Has attributes:</p> <p><i>Type</i>: defines the type of name and is optional. Example: Former name, full name, etc.</p> <p><i>NameDetailsKeyRef</i>: 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.</p> <p><i>Code</i>: Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services.</p>
Name details of a person as a free format text	NameLine	<p>This element is used to define the name of a 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.</p>
Preceding Title	PrecedingTitle	<p>This element is a sub-element of “PersonName” element. Can occur many times and is optional (0 or more). Defines Preceding titles. Example: HIS EXCELLENCY, HONARABLE, etc. Has attributes:</p> <p><i>Type</i>: Defines the type of preceding title and is optional. Example: Honorary title</p> <p><i>Code</i>: Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services.</p>
Title	Title	<p>This element is a sub-element of “PersonName” element. Can occur many times and is optional (0 or more). Defines titles. Has attributes:</p> <p><i>Type</i>: defines the type of title and is optional. Example: Sex, Honorary, Profession, etc.</p> <p><i>Code</i>: Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services</p>
First Name	FirstName	<p>This element is a sub-element of “PersonName” element. Defines first name of a person. Can occur multiple times and is optional (0 or more). Has the following attributes:</p> <p><i>Type</i>: Defines the type for first name and is optional. Example: Old Name, Official, UnOfficial, Initials, etc.</p> <p>Some countries have two first names, one is official name and the other is an unofficial name.</p> <p><i>NameType</i>: Defines the type of name and is optional. Example: Given Name, Christian Name, First Name, etc.</p> <p><i>Code</i>: Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services</p>
Middle Name	MiddleName	<p>This element is a sub-element of “PersonName” element. Defines middle name of a person. Can occur multiple times and is optional (0 or more). Has the following attributes:</p> <p><i>Type</i>: Defines the type for middle name and is optional. Example: Old Name, Official, UnOfficial, Initials, etc.</p> <p><i>NameType</i>: Defines the type of name and is optional. Example: Given Name, Christian Name, First Name, etc.</p> <p><i>Code</i>: Some postal services use a special code to define the element and is</p>

Name Elements	xNL Elements (XML Tags)	Description
		optional. Example: ECCMA Code Tables for postal services
Last name prefix	LastNamePrefix	This element is a sub-element of “PersonName” element. Can occur once and is optional. Defines the prefix of last name. Example: van de, de la, etc. as in France and Netherlands. Has an attribute: <i>Type</i> : Defines the type LastNamePrefix and is optional. Example: Official, UnOfficial, etc. <i>Code</i> : 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. Defines last name of a person. Can occur multiple times and is optional (0 or more). Has the following attributes: <i>Type</i> : Defines the type of last name and is optional. Example: Old Name, Official, UnOfficial, Initials, etc. <i>NameType</i> : Defines the type of name and is optional. Example: last Name, SurName, Family Name, Father’s name (as in India) etc. <i>Code</i> : 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. Defines other names of a person. 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 nameHas the following attributes: <i>Type</i> : Defines the type for other name and is optional. Example: Old Name, Official, UnOfficial, etc. <i>NameType</i> : Defines the type of name and is optional. Example: Maiden Name, Patronymic name, Matronymic name, etc. <i>Code</i> : 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. Can occur once and is optional.
Alias	Alias	This element is a sub-element of “PersonName” element. Defines alias names of a person. Can occur multiple times and is optional (0 or more). Has the following attributes: <i>Type</i> : Defines the type for other name and is optional. Example: Old Name, Official, UnOfficial, etc. <i>NameType</i> : Defines the type of name and is optional. Example: Nickname, pet name, etc. <i>Code</i> : 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. Defines the generation identifier of a person. Can occur multiple times and is optional (0 or more). Example: JNR, III, THE THIRD, etc. Has attributes: <i>Type</i> : Defines the type for generation identifier and is optional. Example: Family Title <i>Code</i> : Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services

Name Elements	xNL Elements (XML Tags)	Description
Suffix	Suffix	This element is a sub-element of “PersonName” element. Defines the suffixes associated with a person’s name. Can occur multiple times and is optional (0 or more). Example: PhD, VC, QC, etc. Has attributes: <i>Type</i> : Defines the type for suffix and is optional. <i>Code</i> : 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. Defines the general suffixes associated with a Person’s name. Can occur once and is optional. Example: Deceased, retired, etc. Has attributes: <i>Type</i> : Defines the type for general suffix and is optional. <i>Code</i> : 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. Can occur once and is optional. 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

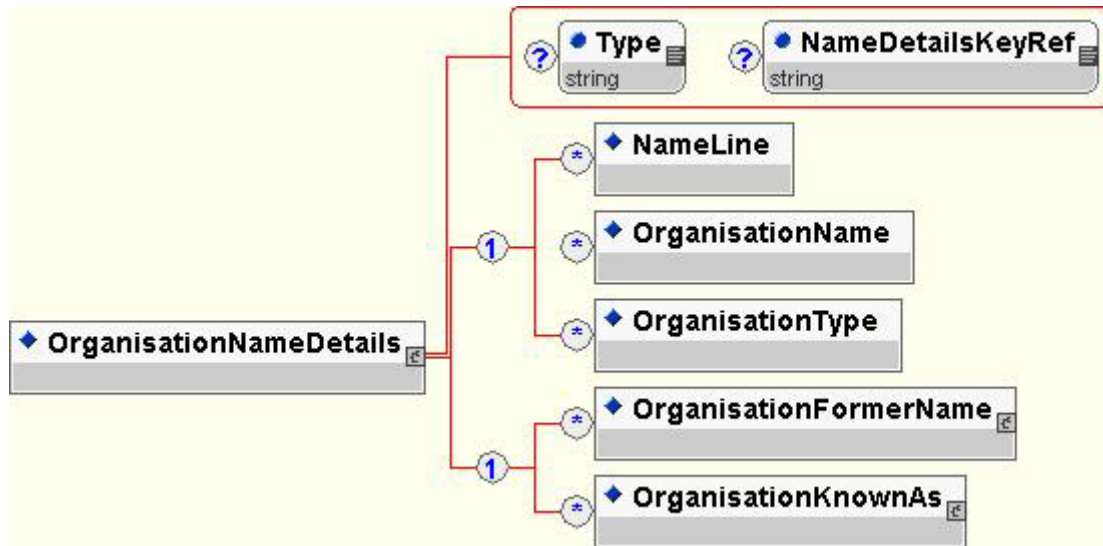
```

<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>
</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. Has attributes: <i>NameDetailsKeyRef</i> : 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. <i>Code</i> : 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. 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. Can occur many times and is optional (0 or more). Example: MSI Business Solutions. Has attributes: <i>Type</i> : Defines the Type of Organisation name and is optional. Example: Official, Legal, Un-official, etc

Name Elements	xNL Elements (XML Tags)	Description
		<p><i>NameType</i>: Defines the name type of the Organisation name and is optional. Example: Former name, new name, abbreviated name etc.</p> <p><i>Code</i>: Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services.</p>
Type for Organisation	OrganisationType	<p>Sub-element of “OrganisationNameDetails” element and is used to define the type for the organisation. 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></p> <p>Has attributes:</p> <p><i>Type</i>: Can be used to define the type and is optional. Example: Abbreviation, Legal Type, etc.</p> <p><i>NameType</i>: Can be used to define the name type and is optional. Example: Official, unofficial, etc.</p> <p><i>Code</i>: Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services</p>
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

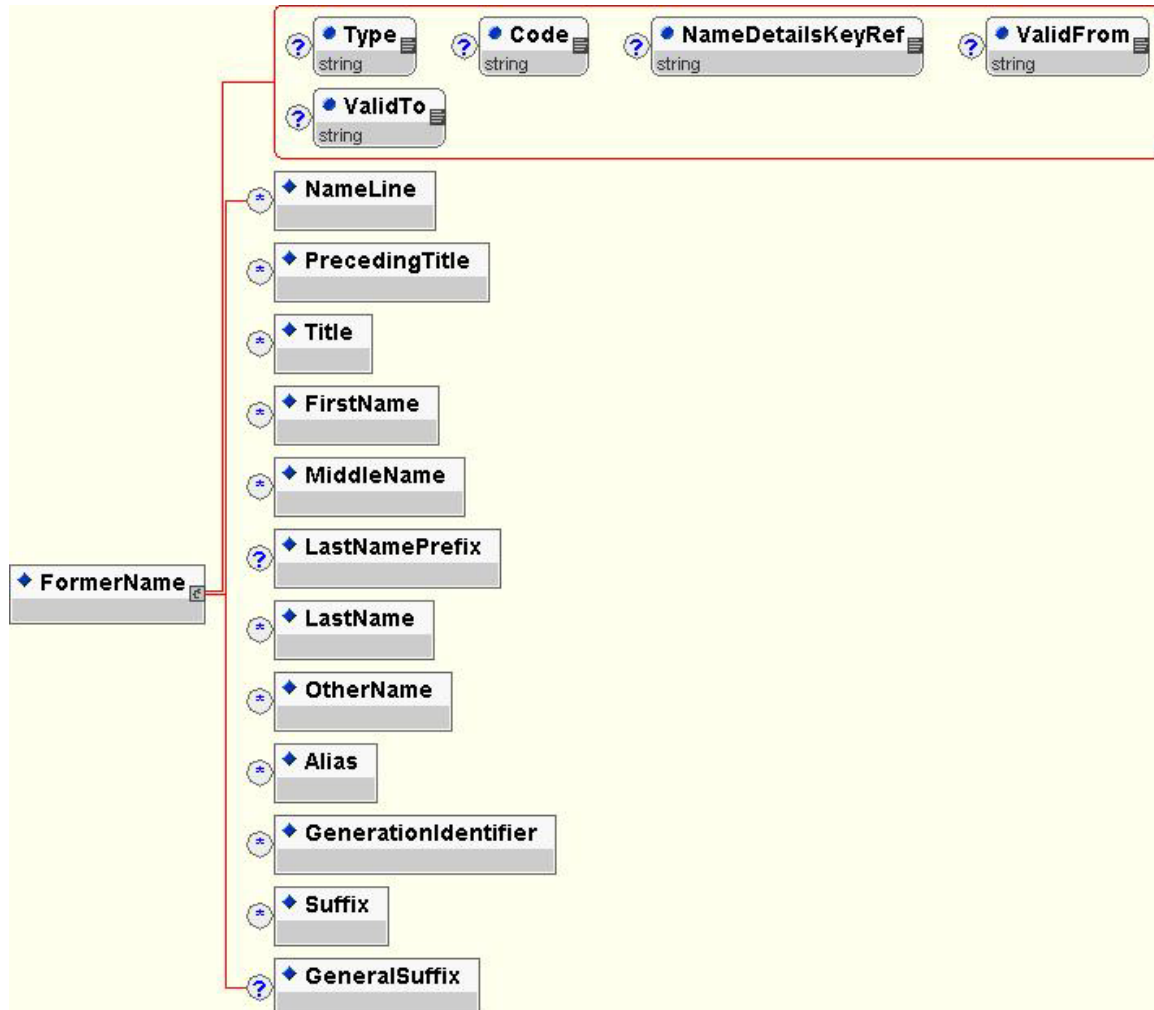
```

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

```

7.6 FormerName Element

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



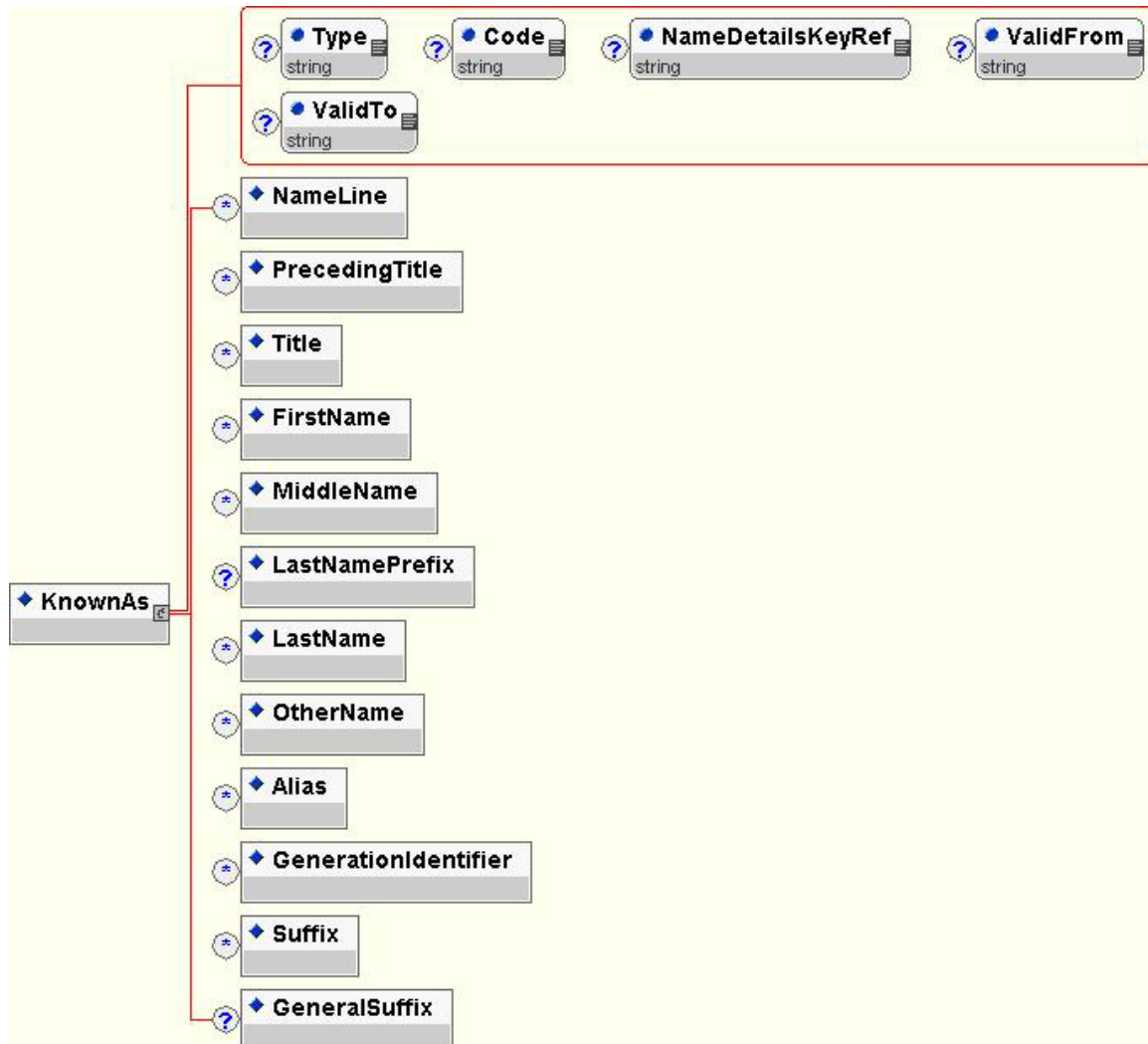
Name Elements	xNL Elements (XML Tags)	Description
Former Name of the Person	FormerName	<p>This element is the sub-element of root element “NameDetails”. This element can occur once and is optional (0 or 1). This element is a container and has sub-elements to define the former name of a person in detail. Has attributes:</p> <p><i>Type</i>: defines the type of name and is optional. Example: full name, etc.</p> <p><i>NameDetailsKeyRef</i>: 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</p> <p><i>Code</i>: Some postal services use a special code to define the element and is</p>

Name Elements	xNL Elements (XML Tags)	Description
		optional. Example: ECCMA Code Tables for postal services. <i>ValidFrom</i> : Defines the date this name was valid from and is optional. <i>ValidTo</i> : 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. 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). Defines Preceding titles. Example: HIS EXCELLENCY, HONARABLE, etc. Has attributes: <i>Type</i> : Defines the type of preceding title and is optional. Example: Honorary title <i>Code</i> : 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). Defines titles. Has attributes: <i>Type</i> : defines the type of title and is optional. Example: Sex, Honorary, Profession, etc. <i>Code</i> : 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 the first name of a person. Can occur multiple times and is optional (0 or more). Has the following attributes: <i>Type</i> : 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. <i>NameType</i> : Defines the type of name and is optional. Example: Given Name, Christian Name, First Name, etc. <i>Code</i> : 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 the middle name of a person. Can occur multiple times and is optional (0 or more). Has the following attributes: <i>Type</i> : Defines the type for middle name and is optional. Example: Old Name, Official, UnOfficial, Initials, etc. <i>NameType</i> : Defines the type of name and is optional. Example: Given Name, Christian Name, First Name, etc. <i>Code</i> : Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services
Last name prefix	LastNamePrefix	This element can occur once and is optional. Defines the prefix of last name. Example: van de, de la, etc. as in France and Netherlands. Has an attribute: <i>Type</i> : Defines the type LastNamePrefix and is optional. Example: Official, UnOfficial, etc. <i>Code</i> : 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. Can occur multiple times and is optional (0 or more). Has the following attributes: <i>Type</i> : Defines the type of last name and is optional. Example: Old Name,

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

7.7 KnownAs Element

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



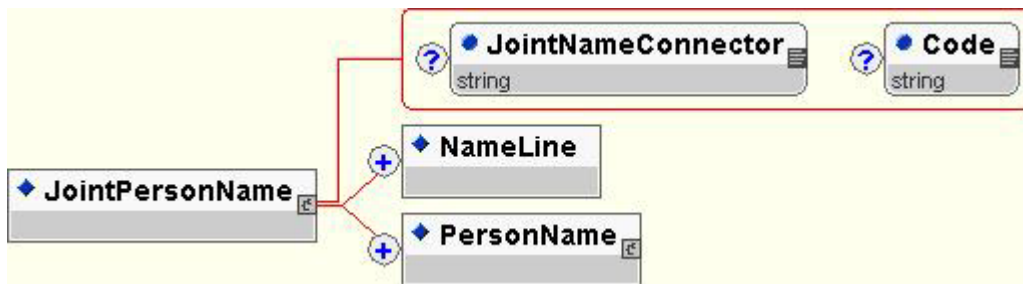
Name Elements	xNL Elements (XML Tags)	Description
Other known names of a Person	KnownAs	<p>This element is the sub-element of root element “NameDetails”. This element can occur once and is optional (0 or 1). This element is a container and has sub-elements to define the other known names of a person in detail. Has attributes:</p> <p><i>Type</i>: defines the type of name and is optional. Example: full name, etc.</p> <p><i>NameDetailsKeyRef</i>: 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</p> <p><i>Code</i>: Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services.</p> <p><i>ValidFrom</i>: Defines the date this name was valid from and is optional.</p> <p><i>ValidTo</i>: Defines the date this name was valid to and is optional.</p>
Name details of a person as a free format text	NameLine	<p>This element is used to define the name of the 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.</p>
Preceding Title	PrecedingTitle	<p>This element can occur many times and is optional (0 or more). Defines Preceding titles. Example: HIS EXCELLENCY, HONARABLE, etc. Has attributes:</p> <p><i>Type</i>: Defines the type of preceding title and is optional. Example: Honorary title</p> <p><i>Code</i>: Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services.</p>
Title	Title	<p>This element can occur many times and is optional (0 or more). Defines titles. Has attributes:</p> <p><i>Type</i>: defines the type of title and is optional. Example: Sex, Honorary, Profession, etc.</p> <p><i>Code</i>: Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services</p>
First Name	FirstName	<p>This element defines the first name of a person. Can occur multiple times and is optional (0 or more). Has the following attributes:</p> <p><i>Type</i>: Defines the type for first name and is optional. Example: Old Name, Official, UnOfficial, Initials, etc.</p> <p>Some countries have two first names, one is official name and the other is an unofficial name.</p> <p><i>NameType</i>: Defines the type of name and is optional. Example: Given Name, Christian Name, First Name, etc.</p> <p><i>Code</i>: Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services</p>
Middle Name	MiddleName	<p>This element defines the middle name of a person. Can occur multiple times and is optional (0 or more). Has the following attributes:</p> <p><i>Type</i>: Defines the type for middle name and is optional. Example: Old Name, Official, UnOfficial, Initials, etc.</p> <p><i>NameType</i>: Defines the type of name and is optional. Example: Given Name, Christian Name, First Name, etc.</p>

Name Elements	xNL Elements (XML Tags)	Description
		<i>Code</i> : Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services
Last name prefix	LastNamePrefix	This element can occur once and is optional. Defines the prefix of last name. Example: van de, de la, etc. as in France and Netherlands. Has an attribute: <i>Type</i> : Defines the type LastNamePrefix and is optional. Example: Official, UnOfficial, etc. <i>Code</i> : 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. Can occur multiple times and is optional (0 or more). Has the following attributes: <i>Type</i> : Defines the type of last name and is optional. Example: Old Name, Official, UnOfficial, Initials, etc. <i>NameType</i> : Defines the type of name and is optional. Example: last Name, Sur Name, Family Name, Father's name (as in India) etc. <i>Code</i> : 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. 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. Has the following attributes: <i>Type</i> : Defines the type for other name and is optional. Example: Old Name, Official, UnOfficial, etc. <i>NameType</i> : Defines the type of name and is optional. Example: Maiden Name, Patronymic name, Matronymic name, etc. <i>Code</i> : 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. Can occur multiple times and is optional (0 or more). Has the following attributes: <i>Type</i> : Defines the type for other name and is optional. Example: Old Name, Official, UnOfficial, etc. <i>NameType</i> : Defines the type of name and is optional. Example: Nickname, pet name, etc. <i>Code</i> : 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. Can occur multiple times and is optional (0 or more). Example: JNR, III, THE THIRD, etc. Has attributes: <i>Type</i> : Defines the type for generation identifier and is optional. Example: Family Title <i>Code</i> : 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. Can occur multiple times and is optional (0 or more). Example: PhD, VC, QC, etc. Has attributes: <i>Type</i> : Defines the type for suffix and is optional.

Name Elements	xNL Elements (XML Tags)	Description
		<i>Code</i> : 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. Can occur once and is optional. Example: Deceased, retired, etc. Has attributes: <i>Type</i> : Defines the type for general suffix and is optional. <i>Code</i> : 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. Has attributes: <i>JointNameConnector</i> : Defines the connector for the joint names and is optional. Example: "and" in Mrs Johnson and Mrs.Johnson. <i>Code</i> : Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services.
Name details of joint 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. 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. Can occur many times and must occur at least once (1 or more). Defines name of person. See section "PersonName" element for further details about this element.

7.8.1 Example

Mrs Joanne Johnshon and Mr.Peter Johnson

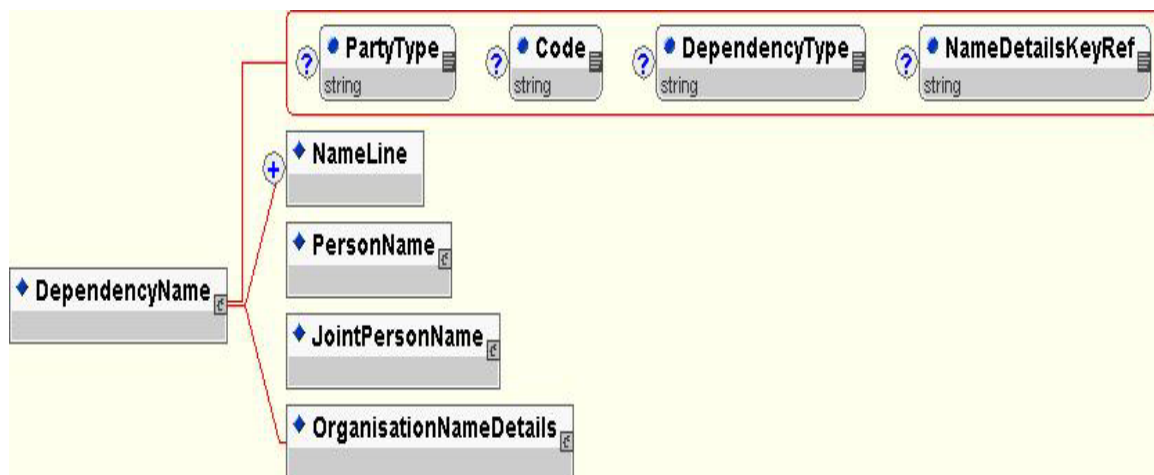
```
<xNL>
  <NameDetails PartyType="Person">
    <JointPersonName>
      <NameLine>Mrs. Joanne Johnson and Mr. Peter Johnson</NameLine>
    </JointPersonName>
  </NameDetails>
</xNL>
```

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.



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: <i>PartyType</i> : Defines the type of customer/party and is optional. Example, Club, Organisation, Person, etc. <i>DependencyType</i> : Defines the type of dependency and is optional. For example, Care of, son of, father of, etc. <i>NameDetailsKeyRef</i> : 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. <i>Code</i> : 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. 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. Can occur once and is optional.
Name of the organisation	OrganisationNameDetails	This element is used to define the name of the dependency organisation. See section “OrganisationNameDetails Element” for further details. Can occur once and is optional.
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. Can occur once and is optional.

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>
```

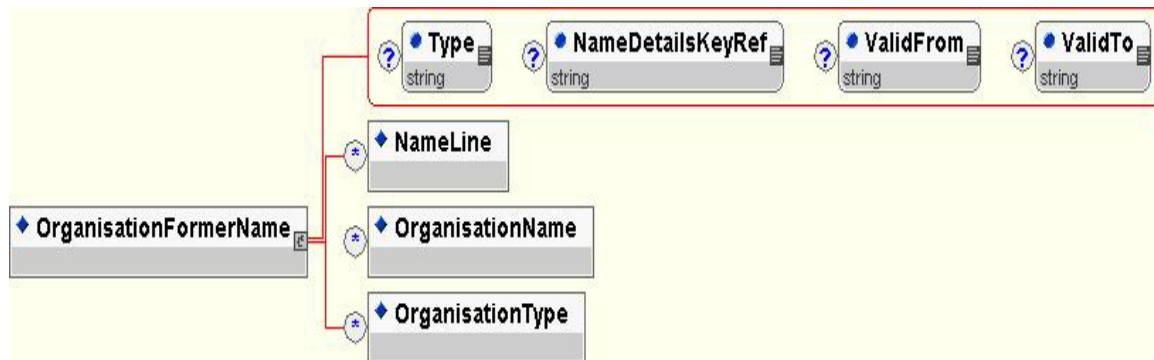
```

        <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. Has attributes: <i>NameDetailsKeyRef</i> : 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. <i>Code</i> : Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services. <i>ValidFrom</i> : Defines the date this name was valid from and is optional. <i>ValidTo</i> : 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. 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. Can occur many times and is optional (0 or more). Example: MSI Business Solutions. Has attributes: <i>Type</i> : Defines the Type of Organisation name and is optional. Example: Official, Legal, Un-official, etc

Name Elements	xNL Elements (XML Tags)	Description
		<p><i>NameType</i>: Defines the name type of the Organisation name and is optional. Example: Former name, new name, abbreviated name etc.</p> <p><i>Code</i>: Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services.</p>
Type for Organisation	OrganisationType	<p>This element is a sub-element of “OrganisationNameDetails” element and is used to define the type for the organisation. 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></p> <p>Has attributes:</p> <p><i>Type</i>: Can be used to define the type and is optional. Example: Abbreviation, Legal Type, etc.</p> <p><i>NameType</i>: Can be used to define the name type and is optional. Example: Official, unofficial, etc.</p> <p><i>Code</i>: Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services</p>

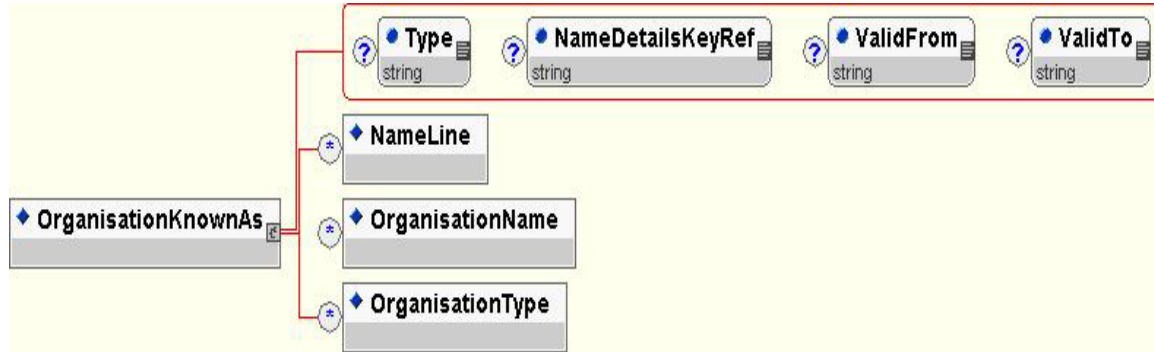
7.10.1 Example

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>
```

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. Has an attribute: <i>NameDetailsKeyRef</i> : 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. <i>Code</i> : Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services. <i>ValidFrom</i> : Defines the date this name was valid from and is optional. <i>ValidTo</i> : 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. 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. Can occur many times and is optional (0 or more). Example: MSI Business Solutions. Has attributes: <i>Type</i> : Defines the Type of Organisation name and is optional. Example: Official, Legal, Un-official, etc <i>NameType</i> : Defines the name type of the Organisation name and is optional. Example: Former name, new name, abbreviated name etc. <i>Code</i> : 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. Can occur many times and is optional (0 or more). Defines the type for organisation. Example: “Pty.LTD”

Name Elements	xNL Elements (XML Tags)	Description
		<p>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></p> <p>Has attributes:</p> <p><i>Type</i>: Can be used to define the type and is optional. Example: Abbreviation, Legal Type, etc.</p> <p><i>NameType</i>: Can be used to define the name type and is optional. Example: Official, unofficial, etc.</p> <p><i>Code</i>: Some postal services use a special code to define the element and is optional. Example: ECCMA Code Tables for postal services</p>

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>
```

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