



*Organization for the Advancement
of Structured Information Standards*

eXtensible Name Language (xNL) Specifications and Description Document

CHANGE HISTORY

Status	Version	Date	Author	Summary of Changes
Draft	1.0	1 March 2001	CIQ-TC	Initial Draft
Draft	1.1	17 May 2001	CIQ-TC	In conjunction with V1.1 of xNL DTD

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 implementors 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 which 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	5
2.0	INTRODUCTION	6
2.1	EXTENSIBLE NAME AND ADDRESS LANGUAGE	6
3.0	THE OBJECTIVE AND SCOPE	6
4.0	USING THE XNL SCHEMA	6
5.0	NAME	8
5.1	RESTRICTIONS	8
6.0	THE XNL DTD GRAMMAR	9
6.1	XNL ELEMENT	10
6.2	NAMEDETAILS ELEMENT	11
6.2.1	Example 1	13
6.2.2	Example 2	14
6.2.3	Example 3	14
6.3	PERSONNAME ELEMENT	14
6.3.1	Example	16
7.0	REFERENCES	17

1.0 Acknowledgements

OASIS wishes to acknowledge MasterSoft International/Research Pty. Ltd, Australia for initiating this standards work to OASIS by submitting its XML standards 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). 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 MasterSoft.

OASIS wishes to acknowledge AND Solution, Inc, Netherlands, for submitting its Global Address Standards to OASIS to be included in this standards effort. Vincent Buller (vincent.buller@and.com), Co-Chair of the Customer Information Quality TC of OASIS and the Senior Consultant of AND Solutions has played an equally significant role along with Ram Kumar in setting up the technical committee and get this standards work going.

OASIS thanks MasterSoft and AND Solutions for their continued support in this effort.

OASIS also wishes to acknowledge the contributions of the other members of the CIQ TC to this standards work.

2.0 Introduction

Customer 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.

3.0 The Objective and Scope

The objective of this document is to describe the eXtensible Name Language (xNL) Document Type Definition (DTD) in detail with examples.

4.0 Using the xNL Schema

The XML 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 (eg. 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.

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. Some users might feel that xNL too much information to represent a simple name for their application. This is not true. 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. Users who do not have an understanding of many of the tags need not use them. Enough flexibility is provided to make the name representation simple without using the detailed level of tags.

For example, let us see how we can use xNL to define the following name:

Mr.Ram Laxhman B Kumar
C/O Mr. Venkat Krishnan

Using xNL, one can define the above name as follows:

```
<xNL>
  <NameDetails CustomerType="Person">
    <Name> Mr.Ram Laxhman B Kumar
      C/O Mr. Venkat Krishnan
    </Name>
  </NameDetails>
</xNL>
```

OR

```
<xNL>
  <NameDetails CustomerType="Person">
    <Name> Mr.Ram Laxhman B Kumar
    </Name>
    <DependencyName DependencyType="C/O"> Mr. Venkat Krishnan</DependencyName>
  </NameDetails>
</xNL>
```

OR

```
<xNL>
  <NameDetails CustomerType="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="C/O"> Mr. Venkat Krishnan</DependencyName>
  </NameDetails>
</xNL>
```

OR

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

5.0 Name

Names can be classified into two, namely:

- Personal Names
- Organisation Names

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
MasterSoft
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 MasterSoft and its address, the xAL DTD must be used as MasterSoft is considered as part of the address.

In the example,

MasterSoft
PO Box: 773, Chatswood, NSW 2067

MasterSoft is the name of the Organisation and it can be represented in xNL. However, the address part must be represented using xAL.

In the example,

Attention: CEO
MasterSoft
PO Box: 773, Chatswood, NSW 2067

Attention and CEO can be represented in xNL. But MasterSoft and the address part must be represented in xAL.

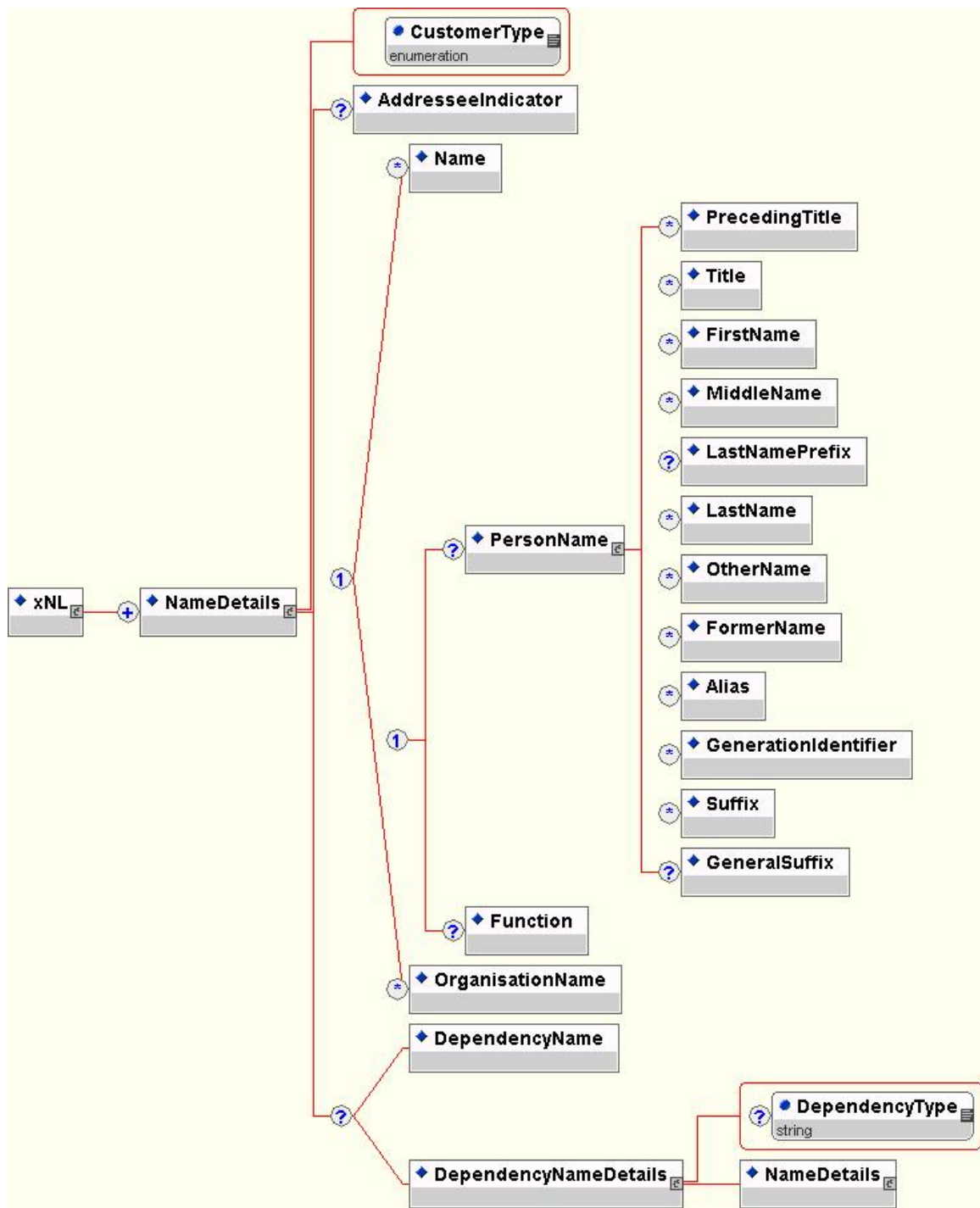
In case of dependency names, both the names can be represented in detail. For example,

Ram Kumar
C/O MasterSoft
PO Box: 773, Chatswood, NSW 2067

Ram Kumar, C/O and MasterSoft can be represented in xNL.

6.0 The xNL DTD Grammar

The figure below shows the complete DTD Grammar for xNL:



6.1 xNL Element

xNL is the root element that contains all name elements to define names in detail.



xNL consists of a sub-element called NameDetails that must occur at least once. xNL container can have multiple NameDetails elements

Example:

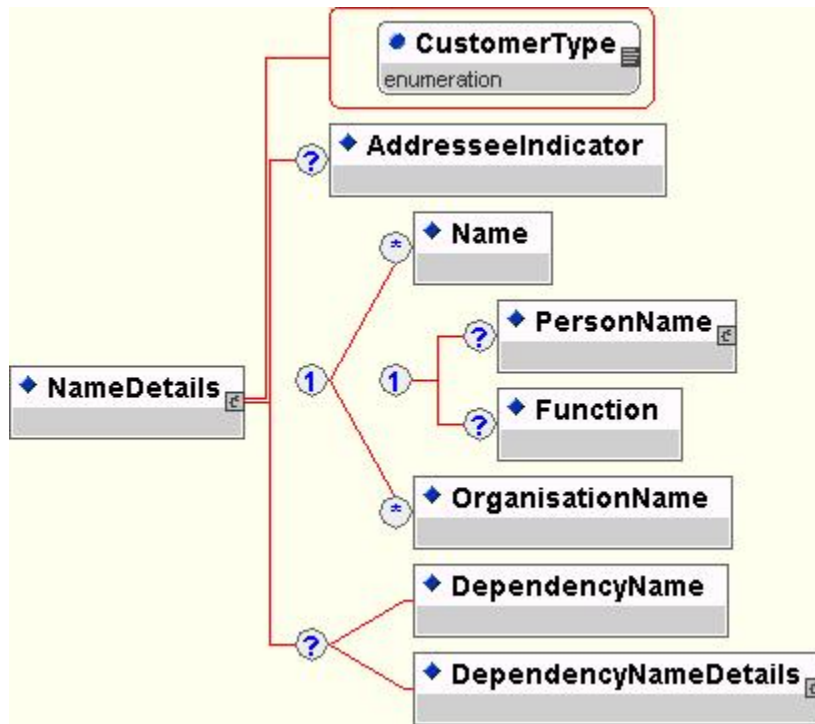
```
<xNL>
  <NameDetails>
    .....
    .....
  </NameDetails>
  <NameDetails>
    .....
    .....
  </NameDetails>
</xNL>
```

6.2 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 (as shown in the figure):

(AddresseeIndicator, (Name **OR** (PersonName, Function) **OR** OrganisationName),
Function, (DependencyName **OR** DependencyNameDetails))



Values in “bold” under XML Tags column indicate that the element has sub-elements.

Name Elements	XML Tags	Description
Name Details	NameDetails	This 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 provides the following attribute: <i>CustomerType</i> : To define the type of customer and is mandatory. Allows only two values namely, “Person”, or “Organisation”. Eg. <xNL> <NameDetails CustomerType=”Person”> <PersonNameDetails> </PersonNameDetails> </NameDetails> </xNL>
Description of the Addressee	AdresseIndicator	A sub-element of “NameDetails” that helps to define the description of the addressee and this is purely for mailing purposes and is optional. Can occur once. Eg. ATTENTION To, ter Attentie van (in Holland), etc.
Name as a general free format text field	Name	A sub-element of “NameDetails” helps 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). Has attribute: <i>NameType</i> : Defines the type of name and is optional. Eg. Joint Names, free format name, etc. Example: <xNL> <NameDetails CustomerType=”Person”> <Name>Mr. Ram Kumar, PhD</Name>

Name Elements	XML Tags	Description
		</NameDetails> </xNL>
Name of the person	PersonName	See section “PersonName Element” for further details. Can occur once and is optional.
Name of the organisation	OrganisationName	A sub-element of “NameDetails” element that defines the name of the organisation. Can occur multiple times and is optional (0 or more). Has two attributes: <i>Type</i> : Defines the type of the entity and is optional. Eg. Pty. Ltd, Inc, etc. <i>NameType</i> : Defines the name type of the organisation and is optional. Eg. Official, Unofficial, legal, etc.
Function/Position/Role	Function	This is a sub-element of “NameDetails” element that defines the position of the person and is optional. This is purely for mailing purposes. Eg. Managing Director, CEO, etc. See section 4.1 for explanation.
Dependent name on another name as a free format text.	DependencyName	This is a sub-element of “NameDetails” element that defines the dependency name as a free format text field. This can occur once and is optional. Has attribute: <i>DependencyType</i> : Defines the type of dependency and is optional. Eg. C/o, W/o, S/o, etc.
Dependent name on another name details	DependencyNameDetails	This is a sub-element of “NameDetails” element that defines the dependency name in detail. Uses NameDetails element as its sub-element. This is an element re-use in XML. Can occur once and is optional. Has attribute: <i>DependencyType</i> : Defines the type of dependency and is optional. Eg. C/o, W/o, S/o, etc.

6.2.1 Example 1

Name: *Mr.Ram V.Kumar*
C/O Mr. Venkat Krishnan

```
<xNL>
  <NameDetails CustomerType="Person">
    <PersonName>
      <Title>Mr</Title>
      <FirstName NameType="GivenName">Ram</FirstName>
      <MiddleName>Laxhman</MiddleName>
      <MiddleName Type="Initial">B</MiddleName>
      <LastName NameType="SurName">Kumar</LastName>
      <FormerName>Ramkumar</FormerName>
      <Alias>Ram</Alias>
    </PersonName>
    <DependencyNameDetails DependencyType="C/O">
      <NameDetails CustomerType="Person">
        <PersonName>
          <Title>Mr</Title>
          <FirstName NameType="GivenName">Venkat</FirstName>
          <LastName>Krishnan</LastName>
        </PersonName>
      </NameDetails>
    </DependencyNameDetails>
  </NameDetails>
</xNL>
```

```
</DependencyNameDetails>
</NameDetails>
```

6.2.2 Example 2

Name: ***Mrs. S. Devi Sadasivan***
 C/o NRMA Pty. Ltd

```
<NameDetails CustomerType="Person">
  <PersonName>
    <Title>Mrs</Title>
    <FirstName>S</FirstName>
    <MiddleName>Devi</MiddleName>
    <LastName>Sadasivan</LastName>
    <Alias>Baby</Alias>
  </PersonName>
  <DependencyNameDetails>
    <NameDetails CustomerType="Organisation">
      <OrganisationName Type="Pty.Ltd">NRMA
    </OrganisationName>
    </NameDetails>
  </DependencyNameDetails>
</NameDetails>
```

6.2.3 Example 3

Name: ***Ram Kumar***
 C/o MasterSoft International

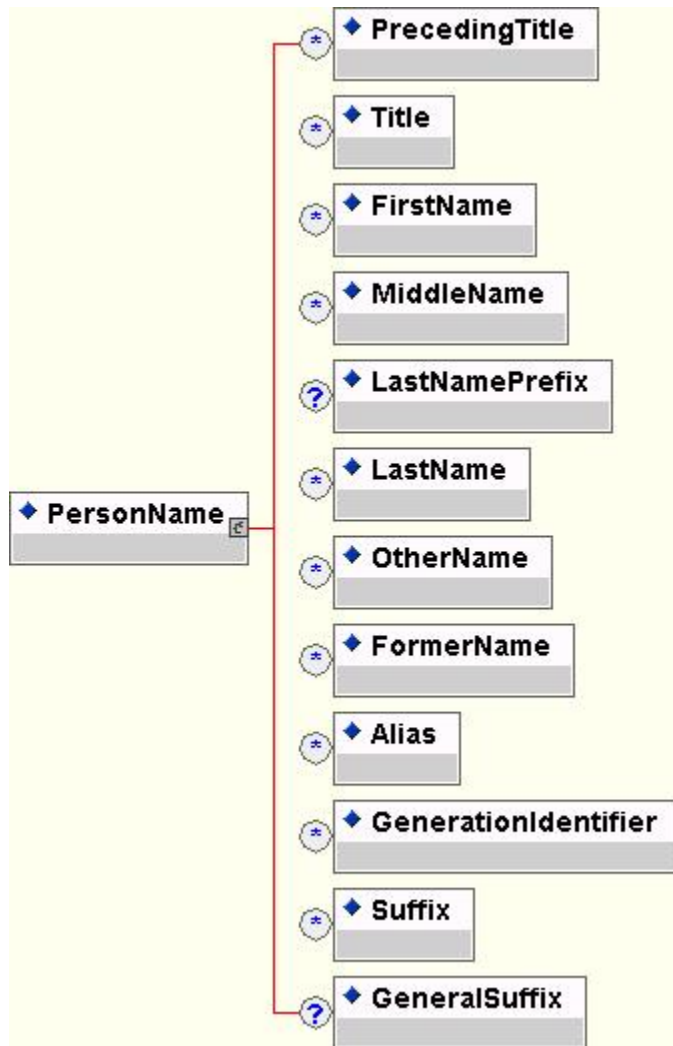
```
<NameDetails CustomerType="Person">
  <Name>Ram Kumar</Name>
  <DependencyName>
    C/O MasterSoft International Pty. Ltd
  </DependencyName>
</NameDetails>
```

6.3 PersonName Element

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

A PersonName Element has the following elements and their relationship is (as shown in the figure):

(PrecedingTitle, Title, FirstName, MiddleName, LastNamePrefix, LastName, OtherName, FormerName, Alias, GenerationIdentifier, Suffix, GeneralSuffix)



Name Elements	XML Tags	Description
Name of the Person	PersonName	This is the sub-element of root element “NameDetails”. This element can occur once and is optional (0 or 1). This element has sub-elements to define the name of a person in detail.
Preceding Title	PrecedingTitle	Sub-element of “PersonName” element. Can occur many times and is optional (0 or more). Defines Preceding titles. Eg. HIS EXCELLENCY, HONARABLE, etc.
Title	Title	Sub-element of “PersonName” element. Can occur many times and is optional (0 or more). Defines titles. Has an attribute called “Type” to define the type of title, eg. Sex, Honorary, Profession, etc.
First Name	FirstName	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: <i>NameType</i> : Defines the type of name and is optional. Eg. Given Name, Christian Name, First Name, etc. <i>Type</i> : Defines the type for first name and is optional. Eg. Old Name, Official, UnOfficial, Initials, etc.

xNL Specifications and Description Document

Name Elements	XML Tags	Description
		Some countries have two first names, one is official name and the other is an unofficial name
Middle Name	MiddleName	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: <i>NameType</i> : Defines the type of name and is optional. Eg. Given Name, Christian Name, First Name, etc. <i>Type</i> : Defines the type for middle name and is optional. Eg. Old Name, Official, UnOfficial, Initials, etc.
Last name prefix	LastNamePrefix	Sub-element of “PersonName” element. Can occur once and is optional. Defines the prefix of last name. Eg. van de, de la, etc. as in France and Netherlands
Last Name	LastName	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>NameType</i> : Defines the type of name and is optional. Eg. last Name, Sur Name, Family Name, Father’s name (as in India) etc. <i>Type</i> : Defines the type for middle name and is optional. Eg. Old Name, Official, UnOfficial, Initials, etc.
Other Name	OtherName	Sub-element of “PersonName” element. Defines other names of a person. Can occur multiple times and is optional (0 or more). Has the following attributes: <i>NameType</i> : Defines the type of name and is optional. Eg. Maiden Name, Patronymic name, Matronymic name, etc.
Former Name	FormerName	Sub-element of “PersonName” element. Defines former names of a person. Can occur multiple times and is optional (0 or more). Has the following attributes: <i>NameType</i> : Defines the type of name and is optional.
Alias	Alias	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>NameType</i> : Defines the type of name and is optional. Eg. Nick name, pet name, etc.
Generation Identifier	GenerationIdentifier	Sub-element of “PersonName” element. Defines alias names of a person. Can occur multiple times and is optional (0 or more). Eg. JNR, III, THE THIRD, etc.
Suffix	Suffix	Sub-element of “PersonName” element. Defines alias names of a person. Can occur multiple times and is optional (0 or more). Eg. PhD, VC, QC, etc.
General Suffix	GeneralSuffix	Sub-tag of “PersonNameDetails” tag. Can occur once and is optional. Eg. Deceased, retired, etc.

6.3.1 Example

Name: *His Excellency Professor Ram Kumar JNR I, PhD*

```
<NameDetails CustomerType="Person">  
  <PersonName>  
    <PrecedingTitle>His Excellency</PrecedingTitle>  
    <Title>Professor</Title>  
    <FirstName NameType="First Name">Ram</FirstName>
```



```
<LastName NameType="Last Name">Kumar</LastName>  
<OtherName NameType="Full Name">Ramkumar Venkatachalam</OtherName>  
<GenerationIdentifier>JNR I</GenerationIdentifier>  
<Suffix>PhD</Suffix>  
</PersonName>  
</NameDetails>
```

7.0 References

- Name and Address Markup Language (NAML) Specifications document (Version 1-1.3), MasterSoft International, April 2000
- Global Address Specifications document (Version 1-1.2), December 2000
- 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

```

<?xml version="1.0" encoding="UTF-8" ?>
- <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="unqualified">
- <xs:annotation>
  <xs:documentation xml:lang="en">xNL: eXtensible Name Language This
    is an XML Schema for xNL DTD V1.1 Date of Creation: 15 November
    2001 Copyright(c) 2001, OASIS. All Rights Reserved
    [http://www.oasis-open.org] Contact: Customer Information Quality
    Technical Committee, OASIS http://www.oasis-
    open.org/committees/ciq Version: 1.1 [in line with V1.1 of xNL DTD]
  </xs:documentation>
</xs:annotation>
- <xs:element name="xNL">
  - <xs:complexType>
    - <xs:sequence>
      <xs:element ref="NameDetails" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
- <xs:annotation>
  <xs:documentation xml:lang="en">Specific for name and address where
    the addressee is specified. eg. ATTENTION, ter attentie van (in
    Holland), etc</xs:documentation>
</xs:annotation>
  <xs:element name="AddresseeIndicator" type="xs:string" />
- <xs:annotation>
  <xs:documentation xml:lang="en">Nick Name, Pet name,
    etc..</xs:documentation>
</xs:annotation>
- <xs:element name="Alias">
  - <xs:complexType>
    - <xs:simpleContent>
      - <xs:extension base="xs:string">
        <xs:attribute name="Type" type="xs:string" />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
- <xs:annotation>
  <xs:documentation xml:lang="en">Person-Person/Person-Organisation
    Relationship</xs:documentation>
</xs:annotation>
- <xs:element name="DependencyName">
  - <xs:complexType>
    - <xs:simpleContent>
      - <xs:extension base="xs:string">
        <xs:attribute name="DependencyType" type="xs:string" />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>

```

```

- <xs:element name="DependencyNameDetails">
- <xs:complexType>
- <xs:sequence>
  <xs:element ref="NameDetails" />
</xs:sequence>
  <xs:attribute name="DependencyType" type="xs:string" />
</xs:complexType>
</xs:element>
- <xs:element name="FirstName">
- <xs:complexType>
- <xs:simpleContent>
  - <xs:extension base="xs:string">
    <xs:attribute name="NameType" type="xs:string" />
    <xs:attribute name="Type" type="xs:string" />
  </xs:extension>
</xs:simpleContent>
</xs:complexType>
</xs:element>
- <xs:element name="FormerName">
- <xs:complexType>
- <xs:simpleContent>
  - <xs:extension base="xs:string">
    <xs:attribute name="NameType" type="xs:string" />
  </xs:extension>
</xs:simpleContent>
</xs:complexType>
</xs:element>
- <xs:annotation>
  <xs:documentation xml:lang="en">Managing Director, CEO, Marketing
  Manager, etc.</xs:documentation>
</xs:annotation>
  <xs:element name="Function" type="xs:string" />
- <xs:annotation>
  <xs:documentation xml:lang="en">DECEASED,
  RETIRED ...</xs:documentation>
</xs:annotation>
  <xs:element name="GeneralSuffix" type="xs:string" />
- <xs:annotation>
  <xs:documentation xml:lang="en">JNR, THE THIRD,
  III</xs:documentation>
</xs:annotation>
  <xs:element name="GenerationIdentifier" type="xs:string" />
- <xs:element name="LastName">
- <xs:complexType>
- <xs:simpleContent>
  - <xs:extension base="xs:string">
    <xs:attribute name="NameType" type="xs:string" />
    <xs:attribute name="Type" type="xs:string" />
  </xs:extension>
</xs:simpleContent>
</xs:complexType>

```

```

</xs:element>
- <xs:annotation>
  <xs:documentation xml:lang="en">DE, LA, VAN DE....</xs:documentation>
</xs:annotation>
<xs:element name="LastNamePrefix" type="xs:string" />
- <xs:element name="MiddleName">
  - <xs:complexType>
    - <xs:simpleContent>
      - <xs:extension base="xs:string">
        <xs:attribute name="NameType" type="xs:string" />
        <xs:attribute name="Type" type="xs:string" />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
- <xs:element name="Name">
  - <xs:complexType>
    - <xs:simpleContent>
      - <xs:extension base="xs:string">
        <xs:attribute name="NameType" type="xs:string" />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
- <xs:annotation>
  <xs:documentation xml:lang="en">CUSTOMER "PERSON" NAME
  DETAILS</xs:documentation>
</xs:annotation>
- <xs:element name="NameDetails">
  - <xs:complexType>
    - <xs:sequence>
      <xs:element ref="AddresseeIndicator" minOccurs="0" />
    - <xs:choice>
      <xs:element ref="Name" minOccurs="0"
      maxOccurs="unbounded" />
    - <xs:sequence>
      <xs:element ref="PersonName" minOccurs="0" />
      <xs:element ref="Function" minOccurs="0" />
    </xs:sequence>
      <xs:element ref="OrganisationName" minOccurs="0"
      maxOccurs="unbounded" />
    </xs:choice>
    - <xs:choice minOccurs="0">
      <xs:element ref="DependencyName" />
      <xs:element ref="DependencyNameDetails" />
    </xs:choice>
  </xs:sequence>
  - <xs:attribute name="CustomerType" use="required">
    - <xs:simpleType>
      - <xs:restriction base="xs:NMTOKEN">
        <xs:enumeration value="Person" />

```

```

        <xs:enumeration value="Organisation" />
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:complexType>
</xs:element>
- <xs:annotation>
  <xs:documentation xml:lang="en">CUSTOMER "ORGANISATION" NAME
  DETAILS</xs:documentation>
</xs:annotation>
- <xs:element name="OrganisationName">
  - <xs:complexType>
    - <xs:simpleContent>
      - <xs:extension base="xs:string">
        <xs:attribute name="Type" type="xs:string" />
        <xs:attribute name="NameType" type="xs:string" />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
- <xs:element name="OtherName">
  - <xs:complexType>
    - <xs:simpleContent>
      - <xs:extension base="xs:string">
        <xs:attribute name="NameType" type="xs:string" />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
- <xs:element name="PersonName">
  - <xs:complexType>
    - <xs:sequence>
      <xs:element ref="PrecedingTitle" minOccurs="0"
        maxOccurs="unbounded" />
      <xs:element ref="Title" minOccurs="0" maxOccurs="unbounded" />
      <xs:element ref="FirstName" minOccurs="0"
        maxOccurs="unbounded" />
      <xs:element ref="MiddleName" minOccurs="0"
        maxOccurs="unbounded" />
      <xs:element ref="LastNamePrefix" minOccurs="0" />
      <xs:element ref="LastName" minOccurs="0"
        maxOccurs="unbounded" />
      <xs:element ref="OtherName" minOccurs="0"
        maxOccurs="unbounded" />
      <xs:element ref="FormerName" minOccurs="0"
        maxOccurs="unbounded" />
      <xs:element ref="Alias" minOccurs="0" maxOccurs="unbounded" />
      <xs:element ref="GenerationIdentifier" minOccurs="0"
        maxOccurs="unbounded" />
      <xs:element ref="Suffix" minOccurs="0"
        maxOccurs="unbounded" />
      <xs:element ref="GeneralSuffix" minOccurs="0" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

```

    </xs:sequence>
  </xs:complexType>
</xs:element>
- <xs:annotation>
  <xs:documentation xml:lang="en">HIS EXCELLENCY, ESTATE OF THE
    LATE .....</xs:documentation>
</xs:annotation>
  <xs:element name="PrecedingTitle" type="xs:string" />
- <xs:annotation>
  <xs:documentation xml:lang="en">Could be compressed initials - PhD, VC,
    QC</xs:documentation>
</xs:annotation>
  <xs:element name="Suffix" type="xs:string" />
- <xs:element name="Title">
  - <xs:complexType>
    - <xs:simpleContent>
      - <xs:extension base="xs:string">
        <xs:attribute name="Type" type="xs:string" />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
</xs:schema>

```

```

<?xml version="1.0" ?>
<!-- Examples based on xNL V1.1 Schema -->
- <xNL xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="xNL.xsd">
- <NameDetails CustomerType="Person">
  - <PersonName>
    <Title>Mr</Title>
    <FirstName Type="GivenName">Ram</FirstName>
    <MiddleName>Laxhman</MiddleName>
    <MiddleName Type="Initial">B</MiddleName>
    <LastName NameType="SurName">Kumar</LastName>
    <FormerName>Ramkumar</FormerName>
    <Alias>Ram</Alias>
  </PersonName>
  - <DependencyNameDetails DependencyType="C/O">
    - <NameDetails CustomerType="Person">
      - <PersonName>
        <Title>Mr</Title>
        <FirstName NameType="GivenName"
          Type="Official">Venkat</FirstName>
        <FirstName NameType="GivenName"
          Type="Unofficial">Venki</FirstName>
        <LastName>Krishnan</LastName>
      </PersonName>
    </NameDetails>
  </DependencyNameDetails>
</NameDetails>
- <NameDetails CustomerType="Person">
  - <PersonName>
    <Title>Mrs</Title>
    <FirstName>S</FirstName>
    <MiddleName>Devi</MiddleName>
    <LastName>Sadasivan</LastName>
    <Alias>Baby</Alias>
  </PersonName>
  - <DependencyNameDetails>
    - <NameDetails CustomerType="Organisation">
      <OrganisationName Type="Pty.Ltd">NRMA</OrganisationName>
    </NameDetails>
  </DependencyNameDetails>
</NameDetails>
- <NameDetails CustomerType="Person">
  <Name>Ram Kumar</Name>
  <DependencyName>C/O MasterSoft International Pty.
    Ltd</DependencyName>
</NameDetails>
</xNL>

```