



PersonName 1.2

Recommendation 2001-Oct-16

This version:

PersonName-1_2

Previous version:

PersonName-1_1

Editors:

Kim Bartkus
Paul Kiel
Mark Marsden

Authors:

Members of the Cross-Process Object work group

Contributors:

Members of the Cross-Process Object work group

Copyright statement

©2001 HR-XML. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publisher. Printed in the United States of America.

Abstract

This document provides all necessary documentation for PersonName, including XML Schema, definitions, and examples.

Status of this Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119.

Table of Contents

1	Overview	3
1.1	<i>Objective</i>	3
1.2	<i>Relationship to Business Processes</i>	3
1.3	<i>Scope</i>	3
1.4	<i>Design Requirements</i>	4
2	Schema/DTD Design	5
2.1	<i>DTD:</i>	5
2.2	<i>XML Schema:.....</i>	6
2.3	<i>DTD/Schema Elements and Data Types Explained.....</i>	7
3	Reference Examples	11
4	Implementation Considerations.....	17
5	Known Limitations	19
6	Appendix A – Schema Source	19
7	Appendix B – DTD Source	20
8	Appendix C - Schema Revision History	21
9	Appendix D – Related Documents	21
10	Appendix E – Schema Examples.....	22
10.1	<i>Example 1: Restriction – Design requires a Legal Name only:.....</i>	22
10.2	<i>Example 2 – Using PersonNameType vs PersonName element.....</i>	23

1 Overview

1.1 *Objective*

Create a schema design for a person's name that is flexible enough to be a global standard, which may be used within other HR-XML Consortium schemas.

1.2 *Relationship to Business Processes*

Almost all HR Business processes pass information regarding individuals, the most common data being the person's name. Names have many components, and these components may vary by country or ethnicity - a Latin American name is very different from a Chinese name. Names may vary by purpose as well. For example, a resume for a performer may contain both a professional name and a legal name. Many business processes today either treat names as a monolithic string, or force names into first/middle/last components. This **PersonName** schema attempts to represent names for a broad array of cultures and purposes so that business processes can pass names reliably and completely, and in a format that can be efficiently processed.

1.3 *Scope*

1.3.1 Within Scope

Define the Name structure for use in other schemas defined by the HR-XML Consortium.

Address legal format for a name.

PersonName-1_2 is an expression of the PersonName-1_1 DTD expanded to include XML Schema syntax. The two are complimentary and instances are intended to be backwardly compatible.

1.3.2 Outside of Scope

Although Effective Dating has been addressed in the DateTime Data Types 1.0 and Effective Dating 1.0 documents, this document does not discuss any effective dating issues, due to backwards compatibility requirements for PersonName 1.0.

While it is important to understand how systems store a person's name and how the different elements of a name are concatenated to form a full name, this proposal does not recommend nor imply how a name should be stored in a database.

Sorting formats for Name are not within the scope of this project.

Due to the requirement for backward compatibility, there is no new functionality in this version other than what is inherent in using XML Schema syntax itself.

1.4 *Design Requirements*

Must be able to handle various name formats without a lot of overhead.

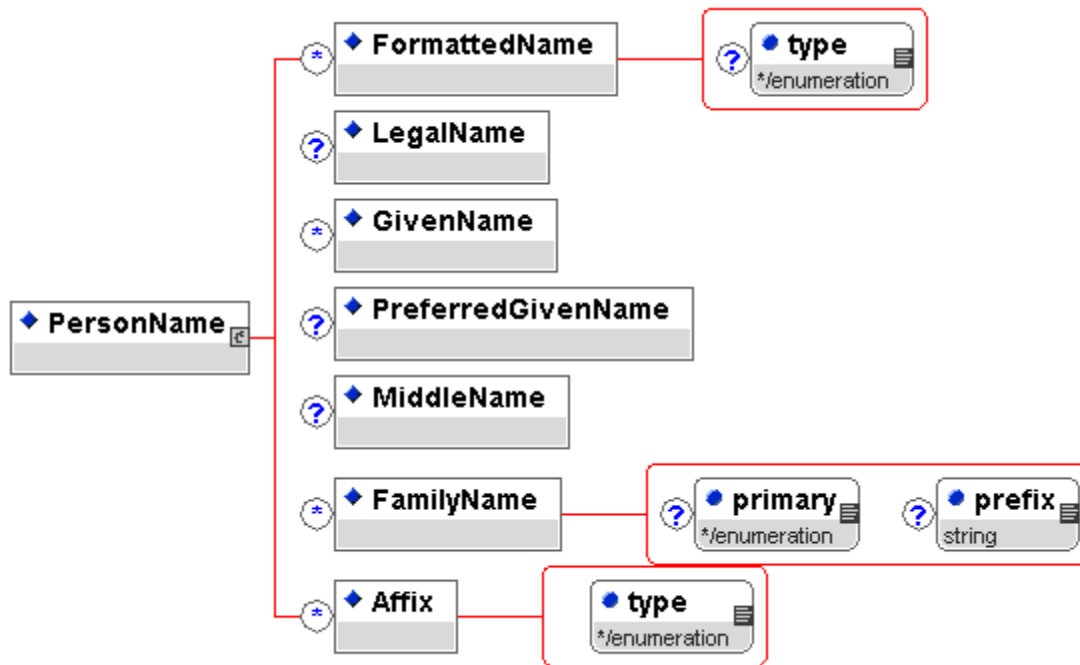
Syntax must be self-documenting.

Must take cultural context into account. Cultural context drives the sort order for a name. It also determines how the various parts of the name are put together to form the whole name.

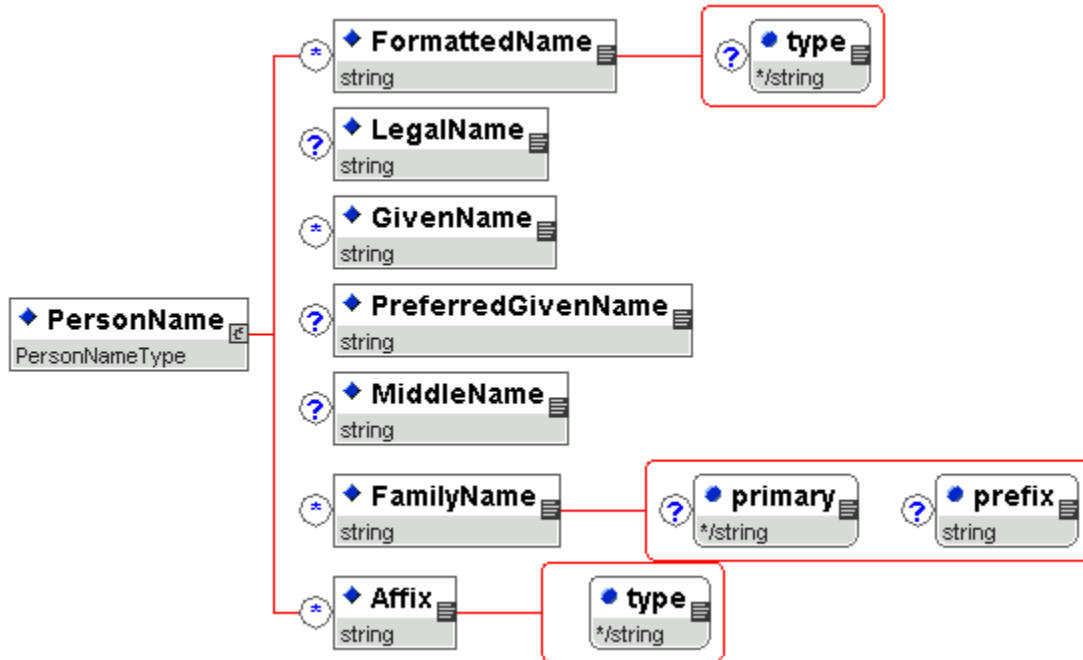
Must be able to handle multiple purposes or contexts of the name (i.e. Employee, Supervisor, Dependant, Beneficiary, etc).

2 Schema/DTD Design

2.1 DTD:



2.2 XML Schema:



This Schema is designed to make instances backwardly compatible with version 1_1. To this end, there are no changes to existing data nodes nor are there any new ones. However, additional with XML Schema is the expanded support for data typing. This specification uses mostly “strings” which correspond to the previous “PCDATA” or “CDATA” data types in DTD. The exception to this is the root element of “PersonName.” While this element is defined with the same content model as version 1_1, its data type has a property unique to XML Schema. The **PersonNameType** data type is a globally defined type that can be utilized elsewhere in the master schema. Should one want to reuse this data type, it can be declared as the data type of another element. For example, an element “ParticipantName” could be defined as having this same data type of “PersonNameType”. As with DTD, element names should be chosen in accordance with Technical Steering Committee guidelines on consistency and clarity.

2.3 DTD/Schema Elements and Data Types Explained

Element / Attribute	Data Type	Content Model	Description
PersonName	PersonName	Root element.	Element used to identify the Person element as a PersonName .
	PersonNameType	Globally defined complex data type	Data type used to identify any person related element with a PersonName content model.
PersonName FormattedName	String, scoped locally	Optional element. 0 to 1 occurrences allowed. 0 to many occurrences deprecated.	Contains, in one string, a fully formatted name with all of its pieces in their proper place. This includes all of the necessary punctuation. This de-normalized form of the name cannot be easily parsed. The use of multiple occurrences are officially deprecated in this release and will be removed in a future version.
PersonName FormattedName type=	Enumeration	Defaulted attribute.	Do not use. This attribute will be deprecated in v1.1 and completely removed in a future version. Defines the purpose of the formatted name. type="presentation" Default value. Associated element contains the full name exactly as it would appear on a purchase order, envelope, etc. type="sortOrder" Associated element contains the name, as it would appear

Element / Attribute	Data Type	Content Model	Description
			<p>in ordered lists.</p> <p>type="legal"</p> <p>Associated element contains the full name exactly as it would appear on legal documents.</p>
PersonName LegalName	string, scoped locally	Optional element. 0 – 1 occurrence.	<p>Legal name used for legal documentation or other legal purposes.</p> <p>Contains, in one string, a fully formatted name with all of its pieces in their proper place. This includes all of the necessary punctuation.</p>
PersonName GivenName	string, scoped locally	Optional element. 0 to many occurrences.	<p>Contains the given or chosen name. Also known as a person's first name. If multiple givenNames are used, the order is implied.</p>
PersonName PreferredGivenName	string, scoped locally	Optional element. One occurrence.	<p>Contains the chosen name by which the person prefers to be addressed.</p> <p>Note: This name may be a name other than a given name, such as a nickname.</p>
PersonName MiddleName	string, scoped locally	Optional element. 0 to many occurrences.	<p>Contains a person's middle name or initial.</p>
PersonName FamilyName	string, scoped locally	Optional element. 0 to many occurrences.	<p>Contains the non-chosen or inherited name.</p> <p>Also known as a person's last name in the Western context.</p> <p>The order is implied by the order of appearance of the FamilyName elements in the</p>

Element / Attribute	Data Type	Content Model	Description
			XML document.
PersonName FamilyName primary=	enumeration	Defaulted attribute.	Defines the order when multiple family names are used. primary="true" Specifies that this occurrence of familyName is primary. It would appear first when multiple familyNames are used. primary="false" Specifies that this occurrence of familyName is not primary. It would appear after the primary familyName. When multiple 'False' primary familyNames are used, the order is implied. primary="undefined" Default specifies that multiple familyNames do not apply. If multiple familyNames are used with this type, order is implied.
PersonName FamilyName prefix=	string, scoped locally	Optional attribute.	Defines the context for the PersonName prefix, such as family name prefix or aristocratic prefix.
PersonName Affix	string, scoped locally	Optional element. 0 to many occurrences.	Contains the remaining parts of the PersonName as defined by the type attribute.
PersonName Affix type=	enumeration	Required attribute.	Defines the context for the affix. type="aristocraticTitle" . e.g. Baron, Graf, Earl, etc. type="formOfAddress" . Contains the Salutation, e.g. Mr., Mrs., Hon., Dr., Major, etc. type="generation" . e.g. Sr., Jr., III (the third),

Element / Attribute	Data Type	Content Model	Description
			<p>etc.</p> <p>type="qualification". Contains the letters used to describe academic or other type qualifications held by a person and/or the distinctions conferred upon them. e.g. PhD, MD, CPA, MCSD, etc.</p> <p>type="academicGrade". Deprecated in v01.01. This will be replaced by the qualification type attribute.</p> <p>type="aristocraticPrefix". Deprecated in v01.01. This will be replaced by the FamilyName prefix attribute.</p> <p>type="familyNamePrefix". Deprecated in v01.01. This will be replaced by the FamilyName prefix attribute.</p> <p>type="familyNameSuffix". Deprecated in v01.01.</p>

3 Reference Examples

The following table lists the methods that various cultures use for representing a person's name. It is intended to make the implementer aware of how diverse a person's name can be. This table should serve as a guide and not the final word on how a culture represents a person's name.

Country	Examples
English (AngloAmerican)	<p>The first and last name may be combined into one name or parsed into given and family name. The following two examples will display 'John Smith'.</p> <pre data-bbox="727 730 1149 840"><PersonName> <GivenName>John</GivenName> <FamilyName>Smith</FamilyName> </PersonName></pre> <pre data-bbox="727 867 1274 947"><PersonName> <FormattedName>John Smith</FormattedName> </PersonName></pre> <p>Military titles may be described with the Affix type 'formOfAddress'. e.g. Major John Smith</p> <pre data-bbox="727 1052 1222 1182"><PersonName> <GivenName>John</GivenName> <FamilyName>Smith</FamilyName> <Affix type="formOfAddress">Major</Affix> </PersonName></pre> <p>Mrs. Jane H. Doe</p> <pre data-bbox="727 1255 1198 1413"><PersonName> <GivenName>Jane</GivenName> <MiddleName>H.</MiddleName> <FamilyName>Doe</FamilyName> <Affix type="formOfAddress">Mrs.</Affix> </PersonName></pre>
French	<p>The abbreviation is used before the proper name or title of the person. If unsure a woman is married, use the married form.</p> <ul data-bbox="706 1619 1365 1791" style="list-style-type: none">• M. Paul Martin (masculine, singular)• M^{me} Pauline Martin (feminine, singular, married woman)• M^{lle} Anne Martin (singular, unmarried)• M. Paul Martin

Country	Examples
	<pre data-bbox="732 317 1214 449"><PersonName> <GivenName>Paul</GivenName> <FamilyName>Martin</FamilyName> <Affix type="formOfAddress">M.</Affix> </PersonName></pre>
Dutch	<p data-bbox="656 499 1438 562">When a family name contains a prefix, it may be transmitted in the 'prefix' attribute.</p> <p data-bbox="656 596 906 625">Mevr. Maria de Wit</p> <pre data-bbox="727 646 1230 779"><PersonName> <GivenName>Maria</GivenName> <FamilyName prefix='de'>Wit</FamilyName> <Affix type="formOfAddress">Mevr.</Affix> </PersonName></pre>
Korean	<p data-bbox="656 829 841 858">Kim, Chul-soo</p> <pre data-bbox="727 879 1138 991"><PersonName> <GivenName>Chul-soo</GivenName> <FamilyName>Kim</FamilyName> </PersonName></pre> <p data-bbox="656 1003 1438 1066">The name represents 3 Korean characters. „Kim“ is the last name, and „Chul-Soo“ is the first name.</p> <p data-bbox="656 1100 1214 1129">In Korea, the last name always comes first.</p> <p data-bbox="656 1163 1365 1268">The State department recommends using a hyphen between the two Korean characters of any Korean first name as shown in the exemplary name above.</p> <p data-bbox="656 1302 1438 1398">Another rule to keep in mind is the second Korean character of any Korean first name starts with a small character as a small ,s' is used in „Chul-soo“.</p>
Mexico	<p data-bbox="656 1434 1045 1463">Sr. Fernando Martínez Urrutia</p> <pre data-bbox="727 1484 1252 1617"><PersonName> <GivenName>Fernando</GivenName> <FamilyName>Martinez Urrutia</FamilyName> <Affix type="formOfAddress">Sr.</Affix> </PersonName></pre> <p data-bbox="656 1619 691 1648">OR</p> <pre data-bbox="727 1650 1354 1808"><PersonName> <GivenName>Fernando</GivenName> <FamilyName primary="false">Martinez</FamilyName> <FamilyName primary="true">Urrutia</FamilyName> <Affix type="formOfAddress">Sr.</Affix> </PersonName></pre>

Country	Examples
	<p>The following abbreviations are used before the proper name or title of the person:</p> <ul style="list-style-type: none"> • Sr. Fernando Martínez Urrutia (masculine, singular.)* • Srita. Eugenia González Pérez (feminine, singular, unmarried)** • Sra. Teresa Fernández de Aguilar (feminine, singular, married woman)*** <p>*Males always keep two family names; the father's family name first and the mother's maiden name last.</p> <p>**Single women use the father's family name first and the mother's maiden name last.</p> <p>***Married woman keep the father's family name and add the husband's family name, followed by the preposition de.</p> <p>If the individual holds one of the following degrees they should be acknowledged as it follows:</p> <ul style="list-style-type: none"> • Ing. (Engineer; masculine and feminine) • Lic. (It indicates that the individual it is, either a lawyer, or holds a Bachelor's Degree; masculine and feminine) • Dr. (Masculine for a physician, surgeon, dentist, veterinarian or the person that holds a PhD) • Dra. (Femenine for a physician, surgeon, dentist, veterinarian or the person that holds a PhD)
Portuguese	<p>The following abbreviations are used before the proper name or title of the person:</p> <ul style="list-style-type: none"> • Sr. João da Silva (masculine, singular) • Sra. Maria da Silva (feminine, singular, married woman) • Srta. Maria Garcia da Silva (feminine, singular, unmarried) <pre> <PersonName> <GivenName>Maria</GivenName> <FamilyName primary=true>Garcia</FamilyName> <FamilyName primary=false prefix='da'>Silva</FamilyName> <Affix type="formOfAddress">Srta.</Affix> </PersonName> </pre>
Japanese	<p>Yamada Taro</p> <pre> <PersonName> <GivenName>Yamada</GivenName> </pre>

Country	Examples
	<p data-bbox="732 268 1138 317" style="text-align: center;"> <FamilyName>Taro</FamilyName> </PersonName> </p> <p data-bbox="656 365 1442 632"> Japanese have two native alphabets: Hiragana and Katakana. Hiragana is used for writing native Japanese words. Katakana is used for writing non-Japanese words. A Japanese name would be written in Hiragana, a non-Japanese name in Katakana. Together the alphabets are known as Kana. Koseki is a family register and may be used with Kana or Romaji. The roman character set is known as Romaji. </p> <p data-bbox="656 663 1425 762"> It is recommended that an implementation using multiple character sets, would use a different PersonName element for each character set. </p>
<p data-bbox="235 800 618 863">Simplified Chinese (Mainland China)</p>	<p data-bbox="656 800 1377 898"> In writing, there are two forms of salutation: in Chinese characters or in Pinyin, the National Phonetic Alphabet, which is a standard system. </p> <p data-bbox="656 930 1446 1098"> In Chinese characters: the person's full name (formal) or last name (informal) is followed by •• (Mr.), or •• (Mrs.), or •• (Ms.), depending upon the person's gender and marriage status. It is always safe to use •• (Mrs) if you don't know the lady's marriage status. The following are some examples: </p> <p data-bbox="656 1129 769 1161">(Formal)</p> <p data-bbox="656 1182 1036 1276"> ••••• (Mr. Zhang, Xiaowei) •••• (Mrs. Jin, Li) •••• (Ms. Liu, Fang) </p> <p data-bbox="656 1297 786 1329">(Informal)</p> <p data-bbox="656 1350 873 1444"> ••• (Mr. Zhang) ••• (Mrs. Jin) ••• (Ms. Liu) </p> <p data-bbox="656 1465 1442 1528"> In Pinyin: Mr./Mrs./Ms. is used before the person's full name or last name. For example: </p> <p data-bbox="656 1560 769 1591">(Formal)</p> <p data-bbox="656 1612 1198 1707"> Mr. Xiaowei Zhang or Mr. Zhang, Xiaowei Mrs. Li Jin or Jin, Li. Ms. Fang Liu or Ms. Liu, Fang </p> <p data-bbox="656 1791 786 1822">(Informal)</p>

Country	Examples
	<p>Mr. Zhang Mrs. Jin Ms. Liu</p> <p>Note: Please be aware that in Pinyin form, it can be either first name first or last name first, except that in the latter, the last name and first name should be separated by a comma. In practice, last name first method is preferred.</p> <p>Also, note that the sort order for Chinese characters is often based on the number of strokes in the symbol. The Chinese character set is used here. However, some operating systems don't recognize this set and may be displayed as asterisks or other symbols.</p>
Traditional Chinese	<p>A person's last name is followed by the Chinese equivalent of Mr. and Ms. in both written and spoken Chinese.</p> <p>If a female is known to be married, her husband's last name followed by Mrs. is also acceptable.</p>
Spain	<p>Married females hold their original name, father's surname first and the mother's maiden name last.</p>
Argentina and other South American Countries	<p>In these countries only one last name is used, males and females, which is the father's last name. Married women use the husband's last name optionally.</p>
India	<p>"Prabhakar Santhanam". Considered a first name and a last name. Some Indians have middle names. Often written as "S. Gowtham" - the family name initial and then the given name. Nicknames are common, in this example "Prabhu".</p>
Greece	<p>"Theodoros Papangelis". Considered a first name and last name. There are no middle names in Greece. Wives almost always take the husband's name. The first name is commonly taken from the paternal grandparents, either the grandmother or the grandfather depending on the gender of the child. Most first names are names of Saints.</p>
Russia	<p>"Mikhail Sergeyevich Gorbachov". Considered a given name, middle name, family name. The family name has a different form for males vs. females. The middle name is often a variant of the father's given name. Nick names (also called 'short name') are common, e.g. "Misha". There is no</p>

Country	Examples
	"Dr." or "Jr." or any other common suffix or prefix. Aristocratic titles are at the front, but are exceedingly rare since the onset of communism.

4 Implementation Considerations

- A person may have multiple family names. Some systems break out these multiple family names into separate data elements. Other systems store the them as one data element. This design will handle both approaches to family name. Implementations of this schema must be aware that the family name could be sent using either format, and respond accordingly. It is expected that if a person has multiple family names, and the names are passed as separate elements, they are passed in the order they would be concatenated to build a name string. (i.e. Fernando Martinez Urrutia)
correct: <GivenName>Fernando</GivenName> <FamilyName primary="false">Martinez</FamilyName> <FamilyName primary="true">Urrutia</FamilyName>.
incorrect: <GivenName>Fernando</GivenName> <FamilyName primary="true">Urrutia</FamilyName> <FamilyName primary="false">Martinez</FamilyName>
- A person has different names for different situations. For example, a woman may use her maiden name for professional reasons and her married name for legal/other reasons. If several versions of a person's name need to be passed for a given context, then the schema design should allow for multiple names at the context level.
- A person's name may be represented using different alphabets. For example, in Japan, a person could have a western style name, a Kana name and/or a Romaji name. Each name uses a different alphabet and may require different elements. (See known limitations)
- Certain parts of a name could end with a period. For example, an initial or salutation typically ends with a period. Some systems may store the period as part of the name. Other systems may not store the period, but add it when formatting the name.
- The <MiddleName> element could contain an initial, followed by a period, or a full name.
- If there is a need to pass a person's initials (i.e. <initials>MJM</initials>), the element would not be part of the PersonName content model. It would need to be defined separately.
- When a hyphenated name is required, the schema may treat the two family names as one. For example, a woman marries and changes her family name to include her husband's family name (i.e. Courtney Thorne-Smith would be <FamilyName>Thorne-Smith</FamilyName>).

- The PersonName content model has no required elements. This is meant to keep the design flexible in terms of handling a wide variation of names. Someone, such as Madonna or Prince, may only have one name. It may be considered their given name, or their family name. This representation should be put into the FamilyName element. In general, however, the given name and the family name are elements that span the most cultures and fit the most contexts.
- If a previous family name (also known as maiden name) needs to be sent for reasons other than building/formatting a person's name, such as a security token, it is assumed the Schema will define a separate data element for this purpose.
- Business rules must be enforced to eliminate the possibility of multiple primary FamilyName elements.
- The FormattedName attribute has been deprecated in version 1.1 and only one occurrence of FormattedName is allowed.
- When a person's name needs to be sent for legal purposes, it is recommended the sender use the LegalName element in addition to sending the other separate name elements. This insures the properly formatted name for legal use.

5 Known Limitations

- This design may not represent names in all cultural contexts. We need the ability to extend the schema.
- Some cultures use different alphabets to represent the same name. In Japan, a person has their name represented using a western alphabet, using a native Japanese alphabet (Kana) and using a roman alphabet (Romaji). This design does not handle multiple representations of the same name within a single **PersonName** content model.

6 Appendix A – Schema Source

```
<!--
"Copyright statement should be included here. It was deleted to conserve space.
-->
<xsd:schema xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd = "http://www.w3.org/2001/XMLSchema"
  elementFormDefault = "unqualified">
  <xsd:complexType name = "PersonNameType">
    <xsd:sequence>
      <xsd:element name = "FormattedName" minOccurs = "0" maxOccurs =
"unbounded">
        <xsd:complexType>
          <xsd:simpleContent>
            <xsd:extension base = "xsd:string">
              <xsd:attribute name = "type" default = "presentation">
                <xsd:simpleType>
                  <xsd:restriction base = "xsd:string">
                    <xsd:enumeration value = "presentation"/>
                    <xsd:enumeration value = "legal"/>
                    <xsd:enumeration value = "sortOrder"/>
                  </xsd:restriction>
                </xsd:simpleType>
              </xsd:attribute>
            </xsd:extension>
          </xsd:simpleContent>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name = "LegalName" type = "xsd:string" minOccurs = "0"/>
      <xsd:element name = "GivenName" type = "xsd:string" minOccurs = "0"
maxOccurs = "unbounded"/>
      <xsd:element name = "PreferredGivenName" type = "xsd:string"
minOccurs = "0"/>
      <xsd:element name = "MiddleName" type = "xsd:string" minOccurs =
"0"/>
      <xsd:element name = "FamilyName" minOccurs = "0" maxOccurs =
"unbounded">
        <xsd:complexType>
          <xsd:simpleContent>
            <xsd:extension base = "xsd:string">
              <xsd:attribute name = "primary" default = "undefined">
```

```

        <xsd:simpleType>
            <xsd:restriction base = "xsd:string">
                <xsd:enumeration value = "true"/>
                <xsd:enumeration value = "false"/>
                <xsd:enumeration value = "undefined"/>
            </xsd:restriction>
        </xsd:simpleType>
    </xsd:attribute>
    <xsd:attribute name = "prefix" type = "xsd:string"/>
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
</xsd:element>
<xsd:element name = "Affix" minOccurs = "0" maxOccurs = "unbounded">
    <xsd:complexType>
        <xsd:simpleContent>
            <xsd:extension base = "xsd:string">
                <xsd:attribute name = "type" use = "required">
                    <xsd:simpleType>
                        <xsd:restriction base = "xsd:string">
                            <xsd:enumeration value = "academicGrade"/>
                            <xsd:enumeration value = "aristocraticPrefix"/>
                            <xsd:enumeration value = "aristocraticTitle"/>
                            <xsd:enumeration value = "familyNamePrefix"/>
                            <xsd:enumeration value = "familyNameSuffix"/>
                            <xsd:enumeration value = "formOfAddress"/>
                            <xsd:enumeration value = "generation"/>
                        </xsd:restriction>
                    </xsd:simpleType>
                </xsd:attribute>
            </xsd:extension>
        </xsd:simpleContent>
    </xsd:complexType>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
<xsd:element name = "PersonName" type = "PersonNameType"/>
</xsd:schema>

```

7 Appendix B – DTD Source

```

<!--
Copyright statement should be included here. It was deleted to conserve space.
-->
<!ELEMENT PersonName (FormattedName*, LegalName?, GivenName*,
PreferredGivenName?, MiddleName?, FamilyName*, Affix*)>
<!ELEMENT FormattedName (#PCDATA)>
<!ATTLIST FormattedName
    type (presentation | legal | sortOrder) "presentation">
<!ELEMENT LegalName (#PCDATA)>
<!ELEMENT GivenName (#PCDATA)>
<!ELEMENT PreferredGivenName (#PCDATA)>
<!ELEMENT MiddleName (#PCDATA)>
<!ELEMENT FamilyName (#PCDATA)>
<!ATTLIST FamilyName
    primary (true | false | undefined) "undefined"
    prefix CDATA #IMPLIED>
<!ELEMENT Affix (#PCDATA)>
<!ATTLIST Affix

```

```

type (academicGrade | aristocraticPrefix | aristocraticTitle |
familyNamePrefix | familyNameSuffix | formOfAddress | generation |
qualification) #REQUIRED>

```

8 Appendix C - Schema Revision History

Version	Date	Revisions
1_2	2001-08-16	Initial draft based on version 1_1 specification
1_2	2001-09-04	<p>Modified this document to more closely match the TSC document template.</p> <p>Modified Schema design changing the PersonName type to be PersonNameType. Updated all examples and references</p> <p>Added Appendix D – Schema Examples</p>
1_2	2001-Oct-16	Approved recommendation by HR-XML Consortium

9 Appendix D – Related Documents

Date/Time Data Types 1.0

Effective Dating 1.0

10 Appendix E – Schema Examples

Since the Schema language lends itself to more of an object oriented approach, schema designers can restrict the Person Name schema to match the context.

10.1 Example 1: Restriction – Design requires a Legal Name only:



```
<xsd:complexType name = "MyLegalName">
  <xsd:complexContent>
    <xsd:restriction base = "PersonName">
      <xsd:sequence>
        <xsd:element name = "LegalName" type = "xsd:string"/>
      </xsd:sequence>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>

<xsd:element name = "TaxFiler" type = "MyLegalName"/>
```

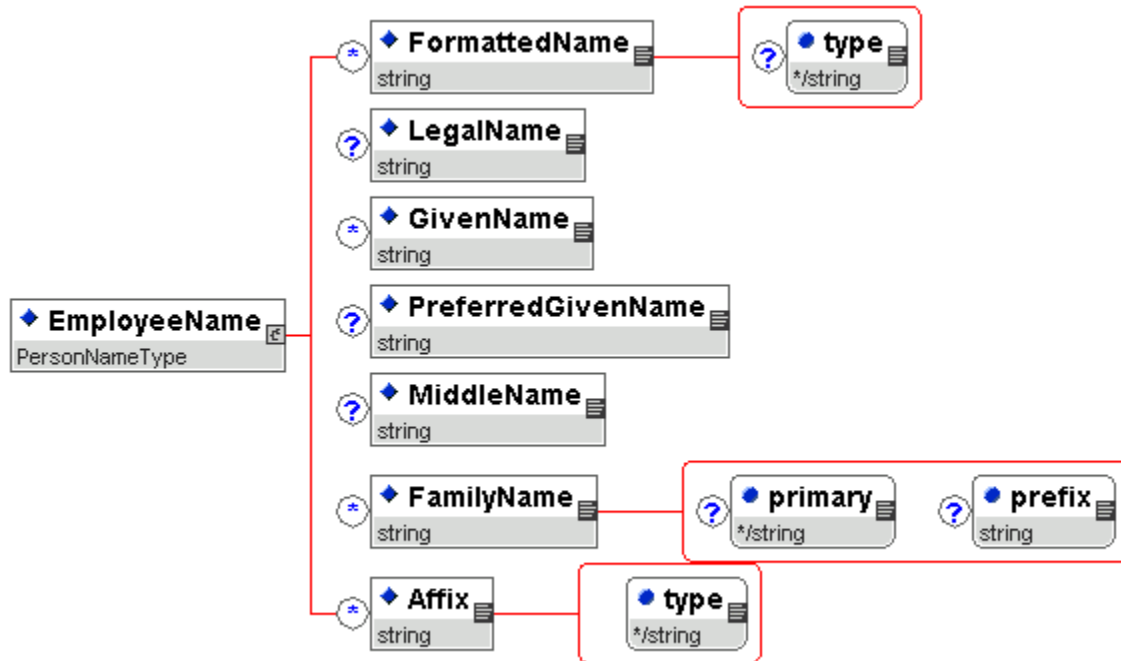
The xml document would contain the following:

```
<TaxFiler>
  <LegalName>John P. Smith III</LegalName>
</TaxFiler>
```

Note that the other elements of PersonName are not allowed to appear in MyLegalName. Also, note that the data element LegalName is now a required data element that can appear once and only once.

10.2 Example 2 – Using PersonNameType vs PersonName element

In the following Schema design, the data element EmployeeName has all of the characteristics of PersonName, simply by defining EmployeeName as type PersonNameType.



Schema Code:

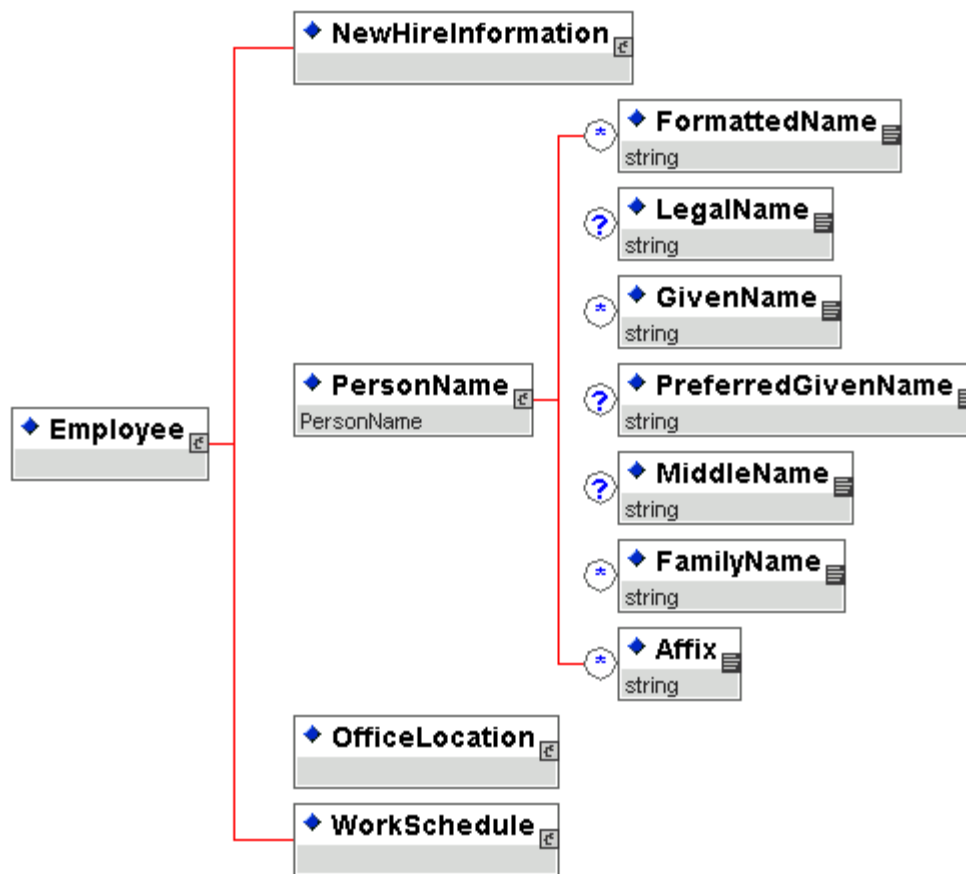
```
<xsd:element name = "EmployeeName" type = "PersonNameType"/>
```

Instance Document Example:

```
<EmployeeName>
  <GivenName>Lorri</GivenName>
  <MiddleName>Marie</MiddleName>
  <FamilyName>Furay</FamilyName>
  <Affix type = "formOfAddress">Ms</Affix>
</EmployeeName>
```

Example 2 continued:

In this example, the PersonName data element is just one of many data elements describing the employee.



Schema Code:

```
<xsd:element name = "Employee">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name = "NewHireInformation">
        <xsd:complexType>
          <xsd:sequence/>
        </xsd:complexType>
      </xsd:element>
      <xsd:element ref = "PersonName"/>
      <xsd:element name = "OfficeLocation">
        <xsd:complexType>
          <xsd:sequence/>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name = "WorkSchedule">
        <xsd:complexType>
          <xsd:sequence/>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```



```
        </xsd:complexType>
    </xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
```

Instance Document Example:

```
<Employee>
  <NewHireInformation/>
  <PersonName>
    <GivenName>Paul</GivenName>
    <FamilyName>Martin</FamilyName>
    <Affix type = "formOfAddress">Mr</Affix>
  </PersonName>
  <OfficeLocation/>
  <WorkSchedule/>
</Employee>
```