



Appendix G: Core Component Naming Convention

Overview

FSA's Core Component Naming Convention will be applied to three entities.

- Core Component Dictionary Entry Names
- XML Type Definition Names
- XML Tag Names

Each of these different entities will have slight differences in their naming conventions; however they will be related to each other. The Core Component Dictionary Entry Name and the XML Type Definition Name will have a naming convention that will ensure each object has a unique name. XML Tag names will be simpler than the Dictionary Entry Names and XML Type Definition Names. This is because XML Tags will rely on their own name, and the context of the Aggregate the XML Tag is located within.

Core Component Dictionary Entry Name Convention

FSA's Core Component Naming convention is based on the Core Component naming convention described in the UNCEFACT Core Component Technical Specification. The UNCEFACT Core Component Technical Specification's naming convention is based on the standards outlined ISO 11179 Part 5 - Naming and Identification Principles for Data Elements. The Core Component Technical Specification expands upon the ISO 11179 naming convention standards to include Core Component Types and Business Information Entities.

The name a Core Component is referenced by in the Core Component Dictionary is the Core Component Dictionary Entry Name. This Dictionary Entry Name is based on the summation of its named parts.

- Object Class Terms
- Property Terms
- Representation Terms
- Qualifier Terms

Name uniqueness is guaranteed because the name is based on the combination of terms that make each Core Component Unique.

FSA's Core Component Naming Convention uses the following rules from the Core Component Technical Specification to produce the Dictionary Entry Names for Core Components:

- The name of an *Object Class* shall be unique throughout the dictionary and may consist of more than one word. The name of a *Property Term* shall occur naturally in the definition and may consist of more than one word. A name of a *Property Term* shall be



unique within the context of an *Object Class* but may be reused across different *Object Classes*.

- If the name of the *Property Term* uses the same word as the *Representation Term* (or an equivalent word), this *Property Term* shall be removed from *Dictionary Entry Name*. The *Representation Term* word in this case only will remain.
- The name of the *Representation Term* shall be one of the terms specified in the *List of Representation Terms* as included in this document.
- The name of the *Representation Term* shall not be truncated in the *Dictionary Entry Name*.
- The *Dictionary Entry Name* shall be unique.
- The *Dictionary Entry Name* shall be extracted from the *Core Component* definition.
- The *Dictionary Entry Name* shall be concise and shall not contain consecutive redundant words.
- The *Dictionary Entry Name* and all its components shall be in singular form unless the concept itself is plural.
- The *Dictionary Entry Name* shall not use non-letter characters unless required by language rules.
- The *Dictionary Entry Name* shall only contain verbs, nouns and adjectives (i.e. no words like *and, of, the, etc.*). This rule shall be applied to the English language, and may be applied to other languages as appropriate.
- Abbreviations and acronyms that are part of the *Dictionary Entry Name* shall be expanded or explained in the definition.
- The *Dictionary Entry Name* of a *Basic Core Component* shall consist of the name of an *Object Class*, the name of a *Property Term* and the name of a *Representation Term*
- The components of a *Dictionary Entry Name* shall be separated by dots. The space character shall separate words in multi-word *Object Classes* and/or multiword *Property Terms*. Every word shall start with a capital letter. To allow spell checking of the *Dictionary Entry Names'* words, the dots after
- *Object Class* and *Property Terms* shall be followed by a space character.
- The *Dictionary Entry Name* of a *Core Component Type* shall consist of a meaningful type name followed by a dot, a space character, and the term *Type*.
- The *Dictionary Entry Name* of an *Aggregate Core Component* shall consist of a meaningful *Object Class* followed by a dot, a space character, and the term *Details*. The *Object Class* may consist of more than one word.
- If the *Object Class* of a *Core Component* is Global, this *Object Class* is not included in the *Dictionary Entry Name*.

For example, if a Basic Core Component had the Object Class of "Person" and the Property Term of "Birth Date", then the Core Component Dictionary Entry Name would be "Person. Birth Date".

XML Types and XML Tags

In an XML instance document, data is placed between tags. Schema designers base these XML tags on types. XML Schema designers can either define their own types, or use previously



defined types. Since FSA's XML Schemas will be based off Core Components, Schema designers will be using already defined types. Both XML Type Definitions and XML Tag Definitions have names. FSA has a set of standards for creating both types of names from the Core Component Dictionary Entry Name. The following example shows the definition of an XML Type and an XML Tag which is based off the defined XML Type for the "Person. Birth Date" Core Component.

Example XML Type Definition

```
<xsd:simpleType name="PersonBirthDateType">  
  <xsd:restriction base="xsd:date"/>  
</xsd:simpleType>
```

Example XML Tag Definition

```
<xsd:element name="BirthDate" type="PersonBirthDateType"/>
```

There are a couple of things to note about XML Tag Names and XML Type Definition Names:

- XML Tag Names and XML Type Definition Names are case sensitive. The first letter of each concatenated word should be in uppercase, and the rest of the letters should be lowercase.
- Length should be considered when creating these names (especially for XML Tag Names), but to the point of sacrificing context. Well known abbreviations and acronyms may be used, but only if context is not lost.
- Representation terms are not necessary in XML Tag Names or XML Type Definition Names, if the representation term for a specific Core Component is not one of the following:
 - Date
 - Indicator
 - Code

XML Type Definition Name Convention

During the creation of a Core Component, a XML type definition is created. This type definition is the XML representation of a Core Component. The name of this XML type definition is based on the Dictionary Entry Name. Three steps need to occur to create an XML Type Definition Name out of a Core Component Dictionary Entry Name.

1. Remove Space Characters and Dots from the Dictionary Entry Name.
2. If the Representation Term is anything other than Date, Code, or Indicator, remove the Representation Term from the modified Dictionary Entry Name.
3. Add the following characters to the end of the modified Dictionary Entry Name "Type".

For example, if a Core Component Dictionary Entry Name was "Person. Birth Date", then the XML Type Definition would be "PersonBirthDateType".



XML Tag Name Convention

When designing an XML Schema, XML Type Definitions are given XML Tag Names. This tag name will provide the description of the data stored in an XML Document. It is critical that this name, as with the other names, be descriptive and concise. Abbreviations and acronyms may be used in XML Tag Names only if the abbreviation or acronyms is commonly known. The use of an abbreviation or acronym should not limit the descriptiveness of a tag name. There are two steps that need to happen to create a XML Tag Name out of a XML Type Definition Name.

1. Remove the characters "Type" from the end of the XML Type Definition Name.
2. If the XML Tag is going to be used within an Aggregate, then the Object Class can be removed from the front of the modified XML Type Definition Name. This can be done because the context for this element can be found from the Aggregate itself.

For Example, if a XML Type Definition Name was "PersonBirthDateType", and the XML Tag is going to be used in a Student Block, then the XML Tag Name would be "BirthDate".

Example Application of Naming Convention

The following table shows an additional example of how Core Component metadata is combined to form a Dictionary Entry Name, an XML Type Definition Name, and an XML Tag Name.

Metadata Field	Metadata Value
Object Class	Address
Property Term	City
Representation Term	Text
Core Component Dictionary Entry Name	Address. City.
XML Type Name	AddressCityType
XML Tag Name	City

Table -Example Application of Naming Convention