



# Taxonomy Package Specification v1.0.0

Copyright © 2011-2012 CoreFiling

# Table of Contents

<b>1 Introduction</b> . . . . .	1
1.1 Namespaces . . . . .	1
1.2 Document Conventions . . . . .	1
1.3 Copyright and Licence . . . . .	1
1.4 Definitions . . . . .	1
<b>2 Specification</b> . . . . .	2
2.1 Package Structure . . . . .	2
2.2 The .taxonomyPackage.xml file . . . . .	2
2.3 Taxonomy Meta-data . . . . .	2
2.3.1 The tp:name element . . . . .	2
2.3.2 The tp:description element . . . . .	2
2.3.3 The tp:version element . . . . .	2
2.4 Remappings . . . . .	2
2.4.1 The tp:remapping element . . . . .	2
2.4.2 Applying remappings . . . . .	3
2.5 Entry Points . . . . .	3
2.5.1 The tp:entryPoint element . . . . .	3
2.5.2 The tp:name element . . . . .	3
2.5.3 The tp:description element . . . . .	3
2.5.4 The tp:version element . . . . .	3
2.5.5 The tp:entryPointDocument element . . . . .	3
2.6 Multi-Lingual Elements . . . . .	4
<b>3 Taxonomy Package Schema</b> . . . . .	5
<b>4 Release history</b> . . . . .	7
References . . . . .	8

## Chapter 1. Introduction

eXtensible Business Reporting Language [XBRL] defines a standard XML-based syntax for business reports. This syntax is based around reporting concepts which are defined in an XBRL taxonomy. Due to the nature of the domains to which XBRL is typically applied, these taxonomies are often very complex, and made up of many constituent XML files. Typically, the majority of these files can be considered internal to the definition of the taxonomy, whilst a small number are intended to be used as "entry points" by XBRL tools.

For convenience XBRL taxonomies are typically distributed as ZIP files, with accompanying human-readable documentation describing which of the component files should be considered entry points. This specification defines a standard format and location for a manifest file that can be included in such ZIP files that allows compliant tools to identify the entry points automatically.

The specification also allows the inclusion of remappings, which provide public locations for the files within the package. This allows XBRL tools to treat the contents of the package as an offline copy of taxonomies published at an Internet location, without the need for additional configuration.

### 1.1. Namespaces

This document uses prefixes to refer to namespaces, as documented in the table below. There is no requirement for implementations of the specification to use these prefixes.

Table 1.1.

Prefix	Namespace
tp	<a href="http://www.corefiling.com/xbrl/taxonomypackage/v1">http://www.corefiling.com/xbrl/taxonomypackage/v1</a>

### 1.2. Document Conventions

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].

### 1.3. Copyright and Licence

This work is licensed under the Creative Commons Attribution-ShareAlike 3.0 Unported License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/3.0/>. This licence allows royalty-free use of the specification, including for commercial purposes, and allows for modification and distribution.

### 1.4. Definitions

**Taxonomy Package** An archive file that contains an XBRL taxonomy and additional metadata, conforming to the format and constraints defined in this specification.

## Chapter 2. Specification

### 2.1. Package Structure

A Taxonomy Package MUST conform to the .ZIP File Format Specification [ZIP]. A Taxonomy Package MUST contain exactly one file named `.taxonomyPackage.xml`. This file MAY appear within any directory within the ZIP file. It is recommended that if the ZIP file contains a single top level directory then the `.taxonomyPackage.xml` SHOULD appear in that directory, otherwise the `.taxonomyPackage.xml` SHOULD appear at the top level within the ZIP.

### 2.2. The `.taxonomyPackage.xml` file

The `.taxonomyPackage.xml` MUST be an XML file [XML] with a root element of `tp:taxonomyPackage`, and MUST conform to the Taxonomy Package Schema.

### 2.3. Taxonomy Meta-data

A Taxonomy Package can provide metadata about the taxonomy, comprising of a name, a description and a version number. The name and description may be provided in multiple languages. References to elements in this section refer only to those elements present as children of the `tp:taxonomyPackage`.

#### 2.3.1. The `tp:name` element

The `tp:name` element provides a human-readable name for the taxonomy. The `tp:name` element is a Multi-Lingual Element.

#### 2.3.2. The `tp:description` element

The `tp:description` element provides a human-readable description for the taxonomy. The `tp:description` element is a Multi-Lingual Element.

#### 2.3.3. The `tp:version` element

The `tp:version` element provides a version identifier for the taxonomy as a whole.

### 2.4. Remappings

A Taxonomy Package can provide remappings that allow one URL to be substituted for another during URL resolution. The typical usage for this is to allow a public, absolute URL (typically using the `http` scheme) to be resolved to a file within a Taxonomy Package. This allows processors to use copies of published taxonomies provided by a Taxonomy Package, rather than retrieving the taxonomy files from the Internet.

#### 2.4.1. The `tp:remapping` element

The `tp:remapping` element defines an individual remapping. The `@prefix` attribute defines a string which is matched against a URL, and the `@replacewith` attribute defines a string that replaces the prefix

if it matches. Note that both of these attributes are strings, and not URIs [URI]. As such, they do not need to be valid URIs, do not undergo normalization, and are not subject to XML Base [XML] processing.

## 2.4.2. Applying remappings

A processor SHOULD apply remappings to any URL that needs to be resolved in the course of processing XBRL files. Remappings are applied to a URL as follows:

Remappings are considered in turn, in the order that they are defined in the `.taxonomyPackage.xml` file. If the prefix matches the URL, then the matching portion of the URL is replaced with the value of the `@replaceWith` attribute, and no further remappings are considered for that URL.

A prefix is considered to match a URL if the URL begins with the prefix string in its entirety. When performing this comparison, the URL MUST undergo Syntax-Based Normalization, Case Normalization, Percent-Encoding Normalization, Path Segment Normalization, and Scheme-Based Normalization, as defined in [URI].

## 2.5. Entry Points

A Taxonomy Package MAY provide an ordered list of entry points. An entry point is a set of URLs that define a logical starting point for the DTS discovery process, as defined in [XBRL]. Each entry point can be documented with name, description and version number. If more than one entry point is defined, the document order in which they are defined SHOULD be used to provide a default ordering when presenting the contents of the Taxonomy Package.

### 2.5.1. The `tp:entryPoint` element

The `tp:entryPoint` element defines an entry point. References to elements in the following sections refer only to those elements present as children of the `tp:entryPoint` element.

### 2.5.2. The `tp:name` element

The `tp:name` element provides a human-readable name for the entry point. The `tp:name` element is a Multi-Lingual Element.

### 2.5.3. The `tp:description` element

The `tp:description` element provides a human-readable description for the entry point. The `tp:description` element is a Multi-Lingual Element.

### 2.5.4. The `tp:version` element

The `tp:version` element provides a version identifier for the entry point.

### 2.5.5. The `tp:entryPointDocument` element

The `tp:entryPointDocument` defines a document that forms part of this entry point. The `@href` attribute provides a URL to the document. This URL is subject to remappings, and as such the URL SHOULD

be the canonical, published location of the document rather than a relative reference to a file within the package.

Relative URLs SHOULD undergo XML Base resolution [XML].

After applying remappings, the `tp:entryPointDocument` will typically resolve to a document within the taxonomy package, but this is not required.

## 2.6. Multi-Lingual Elements

Where an element is specified as being a Multi-Lingual Element, the following additional constraints are applicable:

- The element MUST be subject to an applicable, non-empty `xml:lang` declaration, as defined by [XML]. It should be noted that the `xml:lang` attribute MAY appear on an ancestor element.
- The language identified by the applicable `xml:lang` declaration MUST be unique across sibling occurrences of the same multi-lingual element.

## Chapter 3. Taxonomy Package Schema

The normative version of the Taxonomy Package Schema is published at <http://schemas.corefiling.com/taxonomypackage/1.0.0/taxonomypackage.xsd>. It is included below for convenience.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!--
  (c) CoreFiling S.A.R.L.

  This work is licensed under the Creative Commons Attribution-ShareAlike 3.0
  Unported License. To view a copy of this license, visit
  http://creativecommons.org/licenses/by-sa/3.0/
-->

<xsd:schema xmlns:tp="http://www.corefiling.com/xbrl/taxonomypackage/v1" xmlns:xsd="http://www.w3.org/2001/XMLSchema" >
  <xsd:import namespace="http://www.w3.org/XML/1998/namespace" schemaLocation="http://www.w3.org/XML/1998/namespace" />

  <xsd:element name="taxonomyPackage" type="tp:taxonomyPackageType" />

  <xsd:complexType name="taxonomyPackageType">
    <xsd:sequence>
      <xsd:group ref="tp:documentationGroup" minOccurs="0" maxOccurs="unbounded" />
      <xsd:element name="version" type="tp:stringType" minOccurs="0" maxOccurs="1" />
      <xsd:element name="remappings" type="tp:remappingsType" minOccurs="0" maxOccurs="1" />
      <xsd:element name="entryPoints" type="tp:entryPointsType" minOccurs="0" maxOccurs="1" />
      <xsd:any namespace="##other" minOccurs="0" maxOccurs="unbounded" processContents="lax" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##any" processContents="lax" />
  </xsd:complexType>

  <xsd:complexType name="remappingsType">
    <xsd:sequence>
      <xsd:element name="remapping" type="tp:remappingType" minOccurs="0" maxOccurs="unbounded" />
      <xsd:any namespace="##other" minOccurs="0" maxOccurs="unbounded" processContents="lax" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##any" processContents="lax" />
  </xsd:complexType>

  <xsd:complexType name="remappingType">
    <xsd:sequence minOccurs="0" maxOccurs="unbounded">
      <xsd:any namespace="##other" processContents="lax" />
    </xsd:sequence>
    <xsd:attribute name="prefix" type="xsd:string" use="required" />
    <xsd:attribute name="replaceWith" type="xsd:string" use="required" />
    <xsd:anyAttribute namespace="##any" processContents="lax" />
  </xsd:complexType>

  <xsd:complexType name="entryPointsType">
    <xsd:sequence>
      <xsd:element name="entryPoint" type="tp:entryPointType" minOccurs="0" maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
```

```
<xsd:any namespace="##other" minOccurs="0" maxOccurs="unbounded" processContents=
</xsd:sequence>
<xsd:anyAttribute namespace="##any" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="entryPointType">
  <xsd:sequence>
    <xsd:group ref="tp:documentationGroup" minOccurs="0" maxOccurs="unbounded" />
    <xsd:element name="version" type="tp:stringType" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="entryPointDocument" type="tp:entryPointDocumentType" minOccurs=
    <xsd:any namespace="##other" minOccurs="0" maxOccurs="unbounded" processContents=
  </xsd:sequence>
  <xsd:anyAttribute namespace="##any" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="entryPointDocumentType">
  <xsd:sequence minOccurs="0" maxOccurs="unbounded">
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:sequence>
  <xsd:attribute name="href" type="xsd:anyURI" use="required" />
  <xsd:anyAttribute namespace="##any" processContents="lax" />
</xsd:complexType>

<xsd:group name="documentationGroup">
  <xsd:choice>
    <xsd:element name="name" type="tp:stringType" />
    <xsd:element name="description" type="tp:stringType" />
  </xsd:choice>
</xsd:group>

<xsd:complexType name="stringType">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string">
      <xsd:anyAttribute namespace="##any" processContents="lax" />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

</xsd:schema>
```



## Chapter 4. Release history

Table 4.1.

Version	Date	Description
1.0.0	2012-05-04	First public release.

# References

[XBRL] *XBRL 2.1 Specification*<sup>1</sup>.

[ZIP] *.ZIP File Format Specification*<sup>2</sup>.

[XML] *Extensible Markup Language (XML) 1.0*<sup>3</sup>.

[XML] *XML Base (Second Edition)*<sup>4</sup>.

[URI] *RFC 3986: Uniform Resource Identifier (URI): Generic Syntax*<sup>5</sup>.

[RFC 2119] *RFC 2119: Key words for use in RFCs to Indicate Requirement Levels*<sup>6</sup>.

<sup>1</sup> <http://www.xbrl.org/Specification/XBRL-RECOMMENDATION-2003-12-31+Corrected-Errata-2008-07-02.htm>

<sup>2</sup> <http://www.pkware.com/documents/casestudies/APPNOTE.TXT>

<sup>3</sup> <http://www.w3.org/TR/REC-xml/>

<sup>4</sup> <http://www.w3.org/TR/xmlbase/>

<sup>5</sup> <http://tools.ietf.org/html/rfc3986>

<sup>6</sup> <http://tools.ietf.org/html/rfc2119>