XML Résumé Library User Guide

For XML Résumé Library Version 1.5.1

by Bruce Christensen

XML Résumé Library User Guide: For XML Résumé Library Version 1.5.1

by Bruce Christensen

Published 30 November 2002

Table of Contents

Introduction

Table of Contents

What is the XML Résumé Library?

The XML Résumé Library is, at its heart, a DTD, a Document Type Definition. That means it is really someone's idea of how a résumé should, or at least could, be structured in XML. Beyond that, it comes with

- Several XSLT style sheets (used to convert the XML version of a résumé to other formats such as plaintext, HTML, PDF, and if you're lucky, RTF).
- A Java-based filter to help you target your résumé(s) toward a particular industry, employer, or job.
- CSS stylesheets, used to control the presentation of the HTML version.
- A helpful `Makefile` that you can customize for your own purposes.

Part I. Getting Started

Chapter 1. Setting up the XML Résumé Library

Peter Hutnick
Edited by Bruce ChristensenMark Miller

Table of Contents

This chapter explains how to install the XML Résumé Library, an Open Source, XML based résumé management system, on UNIX systems. It is written based on the author's experience installing on a Red Hat Linux system. Other systems may vary somewhat.

Prerequisites

Java

The resources in this document provide enough information to convert your XML résumé to HTML, PDF, and plain text. The programs to exploit these resources, however, are not included. Any capable XML parser, XSLT stylesheet processor and XSL formatting objects processor will get the job done. Many of the XML Résumé Library developers use the free tools produced by the Apache XML Project, and we recommend that you do as well. These tools include Xerces, Xalan, and FOP.

For the purposes of this document a functioning JRE (Java Runtime Environment) is a requirement. Obtaining and installing a JRE is outside the scope of this document. However, you might find Sun's J2SE web site [http://java.sun.com/j2se/] a good starting point.

XML

A minimal understanding of XML is an asset for using the XML Résumé Library. Most users find that they can get by following the examples in the absence of XML experience.

For a more in-depth tutorial on authoring a résumé in XML, see the section called “Writing your first XML Résumé”. For a reference on XML elements available in the XML Résumé Library DTD, see Element Reference.

Getting the Software

XML Résumé Library

The XML Résumé Library is available for download at http://sourceforge.net/project/showfiles.php?group_id=29512 and is linked from the main XML Résumé Library page at http://xmlresume.sourceforge.net/.

Xerces

Xerces is an XML parser, which is needed by the tools used in later steps of processing. It converts an XML document into a format that other computer programs can more easily use.

The main Xerces for Java2 page is http://xml.apache.org/xerces2-j/, with the download page at http://xml.apache.org/dist/xerces2-j/.

Xalan

Xalan does the actual conversion of a résumé from XML to other formats. When combined with the XSL stylesheets provided by the XML Résumé Library, it produces directly usable HTML and plain text. It also produces an intermediate format called XSL-FO (XSL Formatting Objects) that can be converted to PDF by an FO processor.

The Xalan-Java project page is http://xml.apache.org/xalan-j/, with downloads at http://xml.apache.org/dist/xalan-j/.

FOP

FOP is a print formatter for XSL formatting objects. It converts XSL-FO documents to PDF.

The main FOP page is http://xml.apache.org/fop/, and the download page is http://xml.apache.org/fop/download.html.

Installing and Configuring the Software

XML Résumé Library

The XML Résumé Library comes in a tarball (or zip archive aimed at Windows users) that includes:

- The DTD
- XSL stylesheets, for converting your XML résumé to plain text, HTML, or XSL Formatting Objects (XSL-FO). (XSL-FO can then be rendered to PDF by an FO processor like FOP.)
- Some sample résumés in XML format
- A Makefile, for generating résumé formats with a simple **make** command
- A few CSS stylesheets that can be used to fine-tune the appearance of the HTML version of your résumé.

The most straightforward way to make use of these resources is to recursively copy the "examples" directory to where you want to work with your résumé. For instance:

```
[resume-1_5_1]$ cp -a examples ~/resume
```

The Makefile uses the filename `resume.xml` by default. If this is unacceptable you will need to edit the line `resume = resume` to match the filename you use, such as `resume = myresume` if your résumé is `myresume.xml`.

Tip

You can also indicate your résumé filename on the command line when you build your résumé, e.g.:

```
[resume]$ make resume=myresume
```

Note

The DTD and XSL files you've downloaded to your computer aren't used by default when building your résumé. Instead, the versions on the XML Résumé Library web site are used. This ensures that you're always using the most recent version; however, it also means that builds take longer (since file have to be downloaded each time you build your résumé) and that you can't build a résumé offline.

If you'd like to use your local copy of the XSL stylesheets, edit the `xsl_base` variable in the example Makefile to point to your local XSL directory. For example:

```
xsl_base = ~/resume-1_5_1/xsl
```

To use your local copy of the DTD, change the `DOCTYPE` declaration at the top of your resume to look like this:

```
<!DOCTYPE resume PUBLIC "-//Sean Kelly//DTD Resume 1.5.1//EN"
  "~/resume-1_5_1/dtd/resume.dtd">
```

Xerces, Xalan, and FOP

Each of these three packages need to be untarred and their jar files copied to your JRE's `ext/` directory. This is typically `/usr/java/jdk1.3.1_02/jre/lib/ext/`, but may vary depending on who packaged your JRE. We will simply refer to it as the `ext/` directory.

Xerces consists of the jars `xmlParserAPIs.jar`, `xercesSamples.jar`, and `xercesImpl.jar`, which are in the top directory of the Xerces zip archive.

Xalan consists of the jars `runtime.jar`, `xalanservlet.jar`, `xalansamples.jar`, `regexp.jar`, `JLex.jar`, `java_cup.jar`, `bsf.jar`, `xml-apis.jar`, `BCEL.jar`, `xsltc.jar`, `xalan.jar`, and `xercesImpl.jar` in the `bin/` directory. All of these need to be copied to the `ext/` directory.

Chapter 2. Exploring the Features of XML Résumé Library

Peter Hutnick
Mark Miller
Edited by Bruce Christensen

Table of Contents

This chapter provides an overview of the features and usage of the XML Résumé Library, an Open Source, XML based résumé management system, on UNIX systems.

Formatting an example resume

Converting to other formats

To get a feel for the power of the XML Résumé Library you should run the example résumés through their paces and see how they turn out after processing. You can do this by renaming the file `example2.xml` to `resume.xml` with the command **mv example2.xml resume.xml**. Then simply issue the command **gmake** while connected to the internet. This should generate the files `resume.txt`, `resume.html`, `resume.fo`, and `resume.pdf` — all versions of Alexis Kernighan's résumé.

Advanced Features of the XML Résumé Library

Filtering to target your resume

Filtering allows you to maintain a single, complete resume.xml file that represents the entirety of your career/academic record, while allowing for the simple creation of targeted résumés. In other words, you can start to think of your XML Résumé as a database of your every skill, every accomplishment, every _____ without worrying about cluttering up the résumé your employer sees. Read on to find out how.

Consider the following snippet of Jane Doe's resume:

```
<skillset>
            <skill>Carpentry</skill>
            <skill>Welding</skill>
            <skill>Java Programming</skill>
            <skill>XML</skill>
            <skill>C++</skill>
            <skill>Good communicator</skill>
            <skill>Gourmet Pastry Creation</skill>
            <skill>Cooking for construction workers</skill>
          </skillset>
```

The problem here is that Jane needs to maintain a record of all her skills, but she doesn't need to tell a potential employer about all of those skills. Sure, Jane could selectively comment out portions of her résumé, but this can be tedious. Instead, she should use attributes to describe the categories of résumé to which a given element applies:

```
<skillset>
            <skill targets="construction,woodworking">Carpentry</skill>
            <skill targets="construction">Welding</skill>
            <skill targets="programming">Java Programming</skill>
            <skill targets="programming">XML</skill>
            <skill targets="programming">C++</skill>
            <skill>Good communicator</skill>
            <skill targets="foodservice">Gourmet Pastry Creation</skill>
            <skill targets="foodservice+construction">Cooking for construction workers</skill>
          </skillset>
```

After defining her targets, Jane can filter her résumé to produce beautifully-formatted, well-focused résumés to send to employers. Her "construction" résumé will contain only those elements applicable to the construction industry, and the job foreman won't have to read about her vast knowledge of meringues and bundt cakes, or her experience coding up an application server in Java.

You may have noticed that some of Jane's skills apply to more than one target, or apply only when two (or more) targets are defined. By specifying a list of targets (separated by commas), Jane can ensure that her Carpentry skill will be included in both "construction" and "woodworking" résumés. Similarly, her "Cooking for construction workers" skill will apply only to résumés where both "foodservice" AND "construction" are targeted.

Elements that have no "targets" attribute are always included in the filtered output. Thus Jane's communications skills will apply to every résumé.

The "targets" attribute can be used in ANY element, but keep in mind that using targets on a high-level element (e.g., history or academics element) will affect all children of that element. Thus if the history element includes a 'targets="foo"' attribute, and "foo" is not included in the targets list during the filtering process, then the entire history section will be filtered out, even if there are subelements of history that do not have the 'targets="foo"' attribute.

Now, keep in mind that defining targets is entirely up to you. You can use the targets attribute in any element, and you can specify any number of values for the target— just keep in mind that comma (,) and plus (+) represent OR and AND, respectively, as per the example above. If the list of possible targets starts to pile up, just look at the end of a filtered resume.xml file: a list of the possible and selected targets is included for convenience.

To get a better idea of how this all works, take a look at `examples/example2.xml`. Trying making the complete resume (with **make all resume=example2**), and then the filtered version (**make filter resume=example2**). Then compare `example2.txt` with `example2-filtered.txt`. To create résumés targeted to other purposes, change **filter_targets** in `Makefile`.

Note

You will need to add the `xmlresume-filter.jar` file to your java CLASSPATH before you can make use of the filter. It is located in the `java` directory of the XML Résumé Library distribution.

Customizing your résumé

With just a few tweaks, you can dramatically change the look of your résumés.

Modifying the XSL Parameters

The first thing to do to customize your resume is to check out the parameters files. The main file is `xsl/params.xsl`, which contains general parameters that you can change. We suggest making a backup of this file by issuing the command **cp params.xsl params.xsl.dist**, and then playing around with the values inside `params.xsl` to see how they affect the formatting and output of your resumes.

Note

> You will need to change `xsl_base` in `Makefile` to point to your local xsl directory before any of your changes can take effect. Otherwise, the default parameters stored on the XMLRésumé website are used instead.

Located in the `xsl/paper` directory are the `a4.xsl` and `letter.xsl` files. These files specify things like margin size and indent size for their respective paper formats.

The `xsl/country` directory contains country-specific parameter files, which contain translations for everything from "Résumé" to "Mobile Telephone". If you would prefer your résumé to be called a Curriculum Vitae, this is where to look. The files are sorted by a 2-letter country code (e.g., nl.xsl for the Netherlands). If you don't see your country there, or if there are errors/omissions in the translations, please attempt a translation and submit the changes back to the project-- while XML may work across all languages, the gentle developers of the XML Résumé Library Project do not.

Modifying the look of your HTML Résumé

Cascading Style Sheets (CSS) are a powerful way to format the look of your HTML Résumé. A selection of sample CSS files are provided in the `css` directory. If you have not already done so, try changing the value of `css.href` in `params.xsl` to a different stylesheet (you may have to adjust the location of the stylesheets). Once you find one you like, you can tweak it by editing the CSS file by hand, or create one that is entirely your own. If you think other people would find your CSS file useful, please consider submitting it back to the project.

Writing your first XML Résumé

This is the fun part. Take a look at both example résumés and choose one suits you best. Then replace the existing information with your own. You may also be able to find example résumés of real people by search for `XML resume` on the internet.

For complete information on the elements valid for the XML Résumé Library see Element Reference.

Where to Go from Here

Documentation

More detailed information about creating and building résumés is available in other sections of this user guide.

For more information about the XML Résumé Library, see the HTML documentation included in the `doc/` directory.

Support

For questions about the package and the DTD, there is a mailing list and a support forum web site.

The online support forum is located at http://sourceforge.net/tracker/?group_id=29512&atid=396336.

To join the mailing list, either visit http://lists.sourceforge.net/lists/listinfo/xmlresume-devel or send a message with `subscribe` *`[youraddress@yourdomain.tld]`* in the body to `<xmlresume-devel-request@lists.sourceforge.net>`.

About this chapter

This chapter was originally written by Peter Hutnick. It was edited and converted to DocBook format by Bruce Christensen. Further editing and additions were done by Mark Miller

Please send comments, suggestions, and especially corrections for this chapter to `<peter at hutnick dot com>`.

Part II. Reference

Element Reference

XML Résumé files are XML files composed mostly of elements. This part of the user guide documents the semantics (meaning) and syntax (structure) of all of the elements that you can use to construct a résumé.

To get started with creating a résumé, take a look Chapter 2, which is a tutorial on creating an XML résumé. Alternatively, you could examine the content model for the `resume` element, and drill down from there.

Changes to the DTD

This sections documents changes to the XML Résumé Library DTD.

Warning

Deprecated elements will still be formatted, and are still valid elements in an XML résumé. However, their use is discouraged, and they will be removed in a future version of the DTD.

Version ???

- Changed elements: `projects` may now be included in the `degree` element.

Version 1.5.0

- Changed elements: `date` can now be used in place of `period` in `membership` and `job`.
- Changed elements: `period` can be used in place of `date` in `award` and `clearance`.
- Changed element: `major` is no longer a required child of `degree`.
- Changed element: The `project` element now supports an optional `title` attribute.

Version 1.4.2

No DTD changes.

Version 1.4.1

- New elements: `clearances` and `clearance`.
- New elements: `gpa`, `score`, and `possible`.
- New element: `location`. It may be contained in `job`, `degree`, or `membership`.
- Change: `name` may now contain a `title`.
- Change: `institution` and `organization` may now contain inline elements (`emphasis`, `citation`, `url`, and `link`).

Version 1.4.0

- New element: `minor`.
- New element: `lastModified`.
- New elements: `awards` and `award`.
- New elements: `interests` and `interest`.
- New elements: `fax`, `pager`, and `instantMessage`. These may be contained in `contact`.
- New attribute: the `phone` element now has a `location` attribute.
- New attribute: the `skill` element now has a `level` attribute.
- Change: the `degree` element may now contain multiple `major` elements.
- Change: the `pub` element may now contain a `url` element.
- Change: the `contact` element may now contain any number of its allowed child elements, and they may appear in any order.
- Change: the `resume` element may now directly contain`skillarea` elements.
- Change: the `skillset` element may now directly contain `skill` elements.
- Change: the `degree` element may now contain either a `period` or a `date` element. It could previously contain only a `date`.
- Change: the `artTitle` and The `bookTitle` elements may now contain `link` elements.
- Change: the `referee` element may now contain an optional `title` element and an optional `organization` element.
- Change: the `month` element is no longer required in the `date` element. (However, it must appear if the `date` contains a `dayOfMonth` element.)
- Deprecated element: `skillareas`. You may simply remove the start and end tags from your résumé.
- Deprecated element: `skills`. You may simply remove the start and end tags from your résumé.
- Deprecated element: `pubDate`. Replace it with a `date` element.
- Deprecated elements: `docpath`, `head`, `node`, `tail`, `label`, and `uri`. Instances of these elements should be removed.

Version 1.3.3

- New element: `link`.
- New attribute: the `address` element now has an optional `format` attribute.
- New attributes: Added namespace support. More specifically, added `xmlns`, `xmlns:xsi`, and `xsi:schemaLocation` attributes to the `resumes` and `resume` elements.
- Change: `employer` may now contain inline elements (`emphasis`, `citation`, `url`, and `link`).
- Deprecated element: `break`. Instances of this element should be removed.

- Deprecated element: `company`. Instances of this element should be removed.
- Deprecated element: `street2`. These elements should be converted to `street` elements.

Version 1.3.2

No DTD changes.

Name

academics -- Container for information about academic experience

academics

Synopsis

Content Model

```
(degrees,note?)
```

Attributes

Name	Type	Default
targets	CDATA	*None*

Description

The `academics` element contains information about a person's academic experience.

Parents

`academics` may be contained in these elements: `resume`

See Also

`pubs`, `memberships`.

Examples

```
<academics>
  <degrees>
    <degree>
      <level>Ph.D</level>
      <major>Toothpick Manufacturing</major>
      <institution>Stanford University</institution>
    </degree>
  </degrees>
  <note>
    <para>
      I also have an honorary doctorate in toothpick structural analysis from
      MIT.
    </para>
  </note>
</academics>

<academics>
  <degrees>
    <degree>
      <level>BA</level>
      <major>Culinary Psychology</major>
    </degree>
  </degrees>
</academics>
```

Name

achievement -- An accomplishment made at a job

achievement

Synopsis

Content Model

```
(#PCDATA|emphasis|citation|url|link)*
```

Attributes

Name	**Type**	**Default**
targets	CDATA	*None*

Description

The `achievment` element contains a single accomplishment made while working at a particular job (e.g. “increased sales by 20%”, or “improved employee efficiency”).

Parents

`achievement` may be contained in these elements: `achievements`

Examples

```
<job>
  <jobtitle>Eggroll Engineer</jobtitle>
  <employer>Chineese Cuisine Systems, Inc.</employer>
  <period>
    <from>
      <date>
        <month>August</month>
        <year>1993</year>
      </date>
    </from>
    <to>
      <present/>
    </to>
  </period>
  <achievements>
    <achievement>
      Wrote an <emphasis>excellent</emphasis> article on shell crispiness for
      the <citation>European Journal of Egg Food Engineering</citation>,
      published at <url>http://www.ejeps.org/articles/crispiness.html</url>.
    </achievement>
    <achievement>
      Decreased filling greasiness by a <emphasis>whopping</emphasis> 30%!
    </achievement>
  </achievements>
</job>
```

Name

achievements -- Container for one or more job achievements

achievements

Synopsis

Content Model

```
(achievement+)
```

Attributes

Name	Type	Default
targets	CDATA	*None*

Description

`achievements` is a wrapper element that contains one or more `achievement` elements. Achievements are specific things that were accomplished at a job.

Parents

`achievements` may be contained in these elements: `job`

Examples

For examples, see `achievement`.

Name

address -- A postal address

address

Synopsis

Content Model

```
(#PCDATA|street|street2|suburb|ward|city|state|province|county|prefecture|zip|postalCode|country|bre
```

Attributes

Name	Type	Default
format	 Enumeration: Enumeration:	*None*
id	ID	*None*
targets	CDATA	*None*

Description

The `address` element defines a single postal address. It does **not** contain information about a person who resides at the address.

Different countries have different address formatting conventions. Address formatting is controlled by the *`address.format`* parameter.

If an address cannot be specified using `address`'s child elements (`street`, `city`, etc.), it may be specified in untagged format. In this case, the address will be formatted verbatim, with line breaks preserved.

Parents

`address` may be contained in these elements: `header`, `referee`

Attributes

`id` A string used to uniquely identify an address. The address may then be referred to by this identifier.

`format` The address format. This attribute specifies the format of an address, and overrides the global *`address.format`* parameter for each address that it appears in.

See Also

`address.format`.

Examples

Example 1. U.S.-style address

```
<address>
  <street>123 Pickle St.</street>
  <street>Apt. #12</street>
  <city>Sourville</city>
  <state>NX</state>
  <zip>99999-9999</zip>
</address>
```

Formatted as:

```
123 Pickle St. Apt. #12
Sourville, NX 99999-9999
```

Example 2. Italian address

```
<address>
  <street>Via Garibaldi, 23</street>
  <city>Sorrento</city>
  <postalCode>123 456</postalCode>
  <province>NA</province>
  <country>Italy</country>
</address>
```

Example 3. Untagged address

```
<address>Reina #35, apt. 4a, e/ Gervasio y Escobar
Ciudad de La Habana, CP 11900
CUBA</address>
```

Formatted as:

```
Reina #35, apt. 4a, e/ Gervasio y Escobar
Ciudad de La Habana, CP 11900
CUBA
```

> ## Warning
>
> You may have noticed that the XML for this address isn't indented as usual. This is because untagged addresses are formatted verbatim, and so any indentation would be preserved in the formatted address. The address text begins on the same line as the start tag for the same reason (we don't want a leading linebreak in the formatted address).

For additional address examples, look at the XML files in the `examples/addressing` directory of the XML Résumé Library distribution.

Name

annotation -- Additional information about a degree

annotation

Synopsis

Content Model

```
(#PCDATA)
```

Attributes

Name	**Type**	**Default**
targets	CDATA	*None*

Description

The `annotation` element provides additional information about an academic degree. It could be used to describe special honors (“Graduated summa cum laude”) or circumstances (“Completed degree in one year in accelerated program”).

Parents

`annotation` may be contained in these elements: `degree`

Examples

```
<degree>
  <level>BS</level>
  <major>Street Cleaning Technology</major>
  <annotation>
    Graduated with highest honors.
  </annotation>
</degree>
```

```
<degree>
  <level>BFA</level>
  <major>Chalkboard Scratching Composition</major>
  <annotation>
    Received dean's award in 1984.
  </annotation>
</degree>
```

Name

artTitle -- A title of an article or other work

artTitle

Synopsis

Content Model

```
(#PCDATA|link)*
```

Attributes

Name	Type	Default
targets	CDATA	*None*

Description

The `artTitle` element marks the title of a sub-work (such as an article or poem) in a book, journal, magazine, newspaper, anthology, or other compilation.

Parents

`artTitle` may be contained in these elements: `pub`

See Also

`bookTitle`.

Examples

```
<pub>
  <artTitle>Lion Brainwashing During the Late 20th Century</artTitle>
  <bookTitle>Journal of the American Feline Psychiatric Association</bookTitle>
</pub>

<pub>
  <artTitle>Cucumber Pickling Process Management</artTitle>
  <bookTitle>Proceedings of the European Brine Process Engineers' Group</bookTitle>
</pub>
```

Name

author -- An author of a publication

author

Synopsis

Content Model

`(#PCDATA)`

Attributes

Name	Type	Default
name	IDREF	*None*
targets	CDATA	*None*

Description

The `author` element contains the name of an author of a work or publication. It may also reference a previously-defined `name` through its `name` attribute. In this case, the element will be processed and displayed as if it were actually the referenced name.

Parents

`author` may be contained in these elements: `pub`

Attributes

`name` The `name` attribute contains the `id` of a `name` element.

If an `author` element specifies this attribute, it should not have any content. In other words, there should be a single tag with a reference (`<author name="joe.smith"/>`) instead of two tags enclosing content (`<author>Joe Smith</author>`).

See Also

`name`.

Examples

```
<pub>
  <bookTitle>Nuclear Engineering for Dummies</bookTitle>
  <author>Edgar Neutron</author>
</pub>

<name id="shirly.mendoza">
  <firstname>Shirly</firstname>
  <surname>Mendoza</surname>
</name>
<pub>
  <artTitle>Cooking Wire-Core Pretzels</artTitle>
  <author name="shirly.mendoza"/>
  <bookTitle>Journal of Metal-Reinforced Confections</bookTitle>
```

```
</pub>
```

Name

award -- An award or other honor

award

Synopsis

Content Model

```
(title,organization?,(date|period)?,description?)
```

Attributes

Name	Type	Default
targets	CDATA	*None*

Description

The `award` element describes an award or other honor received by a person (e.g. “Nobel Prize”, “MacArthur Fellow”).

Parents

`award` may be contained in these elements: `awards`

Examples

```
<award>
  <title>Dean's List</title>
  <organization>ACME University</organization>
</award>

<award>
  <title>Honor Roll</title>
  <organization>Littleton Middle School</organization>
  <date>
    <year>1995</year>
  </date>
  <description>
    <para>Maintained a 4.0 GPA.</para>
  </description>
</award>
```

Name

awards -- Container for one or more awards

awards

Synopsis

Content Model

```
(title?,award+)
```

Attributes

Name	Type	Default
targets	CDATA	*None*

Description

The `awards` element contains one or more `award`s.

Parents

`awards` may be contained in these elements: `resume`

Examples

```
<awards>
  <title>Awards and Honors</title>
  <award>
    <title>Employee of the Month</title>
    <organization>ACME, Inc.</organization>
    <date><month>May</month><year>2002</year></date>
    <description>
      <para>
        Received for <emphasis>exceptional</emphasis> customer service.
      </para>
    </description>
  </award>
  <award>
    <title>AP Scholar</title>
    <organization>College Board</organization>
    <date><year>2002</year></date>
  </award>
</awards>
```

Name

birth -- Container for information about a person's birth

birth

Synopsis

Content Model

```
(date)
```

Attributes

Name	**Type**	**Default**
targets	CDATA	*None*

Description

The `birth` element contains a single `date` that specifies when a person was born.

Parents

`birth` may be contained in these elements: `header`

Examples

```
<header>
  <name>
    <firstname>Scotty</firstname>
    <surname>O'Reilly</surname>
  </name>
  <birth>
    <date>
      <dayOfMonth>17</dayOfMonth>
      <month>March</month>
      <year>1931</year>
    </date>
  </birth>
</header>
```

```
<header>
  <name>
    <firstname>Josè</firstname>
    <middlenames>Luis Miguel</middlenames>
    <surname>Domingo</surname>
  </name>
  <birth>
    <date>
      <dayOfMonth>23</dayOfMonth>
      <month>Mayo</month>
      <year>1834</year>
    </date>
  </birth>
</header>
```

Name

bookTitle -- A title of a book or similar work

bookTitle

Synopsis

Content Model

```
(#PCDATA|link)*
```

Attributes

Name	Type	Default
targets	CDATA	*None*

Description

The `bookTitle` element marks the title of a book, magazine, journal, anthology, or other similar work. To denote the title of a sub-work (like an article), see `artTitle`.

Parents

`bookTitle` may be contained in these elements: `pub`

See Also

`artTitle`.

Examples

```
<pub>
  <bookTitle>Tropical Siberian Vacations</bookTitle>
  <author>Boris Tzchleikovsy</author>
</pub>

<pub>
  <bookTitle>Children's Bedtime Stories</bookTitle>
  <artTitle>Jack the Axe Murderer</artTitle>
  <author>Caroline Black</author>
</pub>

<pub>
  <artTitle>Global Commodities Market Downturn</artTitle>
  <bookTitle>The Economist</bookTitle>
  <pubDate>
    <month>June</month>
    <year>1992</year>
  </pubdate>
</pub>
```

Name

break -- A linebreak

break

Synopsis

Content Model

`EMPTY`

Attributes

Name	Type	Default
targets	CDATA	*None*

Description

Important

This element is deprecated as of XML Résumé Library version 1.3.3. It should not be used.

`break` was introduced to allow linebreaks in free-form addresses, but the stylesheets have been modified so that they don't need an explicit linebreak marker. Instead, linebreaks in source text are formatted as line-breaks in formatted text.

The `break` element indicates a position in text where a linebreak should occur.

Parents

`break` may be contained in these elements: `address`

Examples

None.

Name

citation -- The name of a work being referenced

citation

Synopsis

Content Model

`(#PCDATA)`

Attributes

Name	Type	Default
targets	CDATA	*None*

Description

The `citation` element marks the title of a book, website, course, or other work.

Parents

`citation` may be contained in these elements: `achievement`, `employer`, `institution`, `organization`, `para`, `project`, `skill`

See Also

`pub`.

Examples

```
<achievement>
  Increased earnings 341%, thereby landing Acme Widgets on <citation>Fortune
  Magazine</citation>'s Fortune 500 list.
</achievement>

<para>
  More information on the AHTM system is published in <citation>Developing
  Today</citation> at <url>http://www.dtoday.bob/tips.html</url>.
</para>
```

Name

city -- The name of a city

city

Synopsis

Content Model

```
(#PCDATA)
```

Attributes

Name	**Type**	**Default**
id	ID	*None*
targets	CDATA	*None*

Description

The `city` element contains a the name of a single city or other similar municipality.

Parents

`city` may be contained in these elements: `address`, `location`

Attributes

`id` A string used to uniquely identify a city. The city may then be referred to by this identifier.

See Also

`county`, `prefecture`, `suburb`, `ward`.

Examples

```
<address>
  <street>3400 N. 5230 S.</street>
  <city>Salt Lake City</city>
  <state>UT</state>
  <zip>84352</zip>
</address>
```

Name

clearance -- A security clearance

clearance

Synopsis

Content Model

```
(level,organization?,(date|period)?,note?)
```

Attributes

Name	**Type**	**Default**
targets	CDATA	*None*

Description

The `clearance` element contains a single security clearance (e.g. “Top Secret”).

Parents

`clearance` may be contained in these elements: `clearances`

Examples

```
<clearance>
  <level>Super-duper it-doesn't-even-exist extra secret</level>
  <organization>NATO</organization>
</clearance>

<clearance>
  <level>Top Secret</level>
  <organization>CIA</organization>
  <date><year>2002</year></date>
  <note>Full polygraph test performed.</note>
</clearance>
```

Name

clearances -- One or more security clearances

clearances

Synopsis

Content Model

```
(title?,clearance+)
```

Attributes

Name	**Type**	**Default**
targets	CDATA	*None*

Description

The `clearances` element contains one or more security clearances. It is a top-level section of a résumé.

Parents

`clearances` may be contained in these elements: `resume`

Examples

```
<clearances>
  <clearance>
    <level>Top Secret</level>
    <organization>NSA</organization>
  </clearance>
  <clearance>
    <level>Secret</level>
    <organization>US Navy</organization>
  </clearance>
</clearances>
```

Name

company -- The name of a company

company

Synopsis

Content Model

```
(#PCDATA)
```

Attributes

Name	Type	Default
targets	CDATA	*None*

Description

Important

This element is deprecated as of XML Résumé Library version 1.3.3. It should not be used.

Parents

`company` may be contained in these elements: None

See Also

`institution`, `organization`.

Examples

None.

Name

contact -- Container for one or more methods of contacting someone

contact

Synopsis

Content Model

```
(phone|fax|pager|email|url|instantMessage)*
```

Attributes

Name	**Type**	**Default**
targets	CDATA	*None*

Description

The `contact` element is a container that holds elements that specify how to contact a person. The information it contains is similar to what would be printed on a business card.

Parents

`contact` may be contained in these elements: `header`, `referee`

See Also

`address`.

Examples

```
<contact>
  <phone>555-555-1212</phone>
  <email>sam.jones@xyz.bob</email>
  <url>http://www.xyz.bob/~sam.jones/</url>
</contact>

<contact>
  <phone>123-123-1234 x123</phone>
</contact>

<contact>
  <email>kawanza_eloma@hotmail.bob</email>
</contact>
```

Name

copyright -- A copyright notice

copyright

Synopsis

Content Model

```
(year,name?,legalnotice?)
```

Attributes

Name	**Type**	**Default**
targets	CDATA	*None*

Description

The `copyright` element defines a copyright notice. The notice must include the year of copyright; it may also include the name of the copyright owner and a legal notice.

Parents

`copyright` may be contained in these elements: `resume`

Examples

```
<resume>
  <copyright>
    <year>1892</year>
    <name>
      <firstname>Fluisha</firstname>
      <surname>Copenhagen</surname>
    </name>
    <legalnotice>
      <para>
        All rights reserved. This document may not be copied or distributed
        without permission.
      </para>
    </legalnotice>
  </copyright>
</resume>
```

```
<copyright>
  <year>2001</year>
</copyright>
```

Name

country -- A country name

country

Synopsis

Content Model

```
(#PCDATA)
```

Attributes

Name	Type	Default
id	ID	*None*
targets	CDATA	*None*

Description

The `country` element marks the name of a country.

Parents

`country` may be contained in these elements: `address`, `location`

Attributes

`id` A string used to uniquely identify a country. The country may then be referred to by this identifier.

Examples

```
<address>
  <city>Portland</city>
  <state>Oregon</state>
  <country>USA</country>
</address>

<address>
  <street>2621 Green Loop</street>
  <street>Southam</street>
  <city>Wiggleworm</city>
  <postalCode>S521 2GR</postalCode>
  <country>UK</country>
</address>
```

Name

county -- A county name

county

Synopsis

Content Model

```
(#PCDATA)
```

Attributes

Name	Type	Default
id	ID	*None*
targets	CDATA	*None*

Description

The `county` element specifies the name of a county (not to be confused with `country`). In the United States, a county is “an administrative subdivision of a state”. In the UK, it is a “territorial division exercising administrative, judicial, and political functions.” (Source: *The American Heritage Dictionary of the English Language*, 1979 Ed. Houghton Mifflin, Boston.)

Parents

`county` may be contained in these elements: `address`, `location`

Attributes

`id` A string used to uniquely identify a county. The county may then be referred to by this identifier.

See Also

`state`, `province`, `prefecture`.

Examples

```
<address>
  <city>North Willow</city>
  <county>Washington</county>
  <state>New Canada</state>
</address>

<address>
  <street>1 O'Leary Place</street>
  <city>Coby</city>
  <county>Co. Cork</county>
  <country>Ireland</country>
</address>
```

Name

date -- A specific instant in time

date

Synopsis

Content Model

```
(((dayOfMonth)?,month)?,year)
```

Attributes

Name	Type	Default
targets	CDATA	*None*

Description

The `date` element represents a specific instant in time, with at least year-level granularity, and at most day-level granularity.

When contained inside a `pub` element, `date` represents the date that the work was published.

When contained inside a `clearance` element, `date` represents the date that the security clearance was granted.

Note

> The `month` element was made optional in version 1.4.0 of the XML Résumé Library. It was previously required.

To represent a period of time, instead of a specific instant, use `period`.

Parents

`date` may be contained in these elements: `award`, `birth`, `clearance`, `degree`, `from`, `job`, `lastModified`, `membership`, `pub`, `to`

See Also

`period`.

Examples

```
<date>
  <dayOfMonth>27</dayOfMonth>
  <month>November</month>
  <year>1634</year>
</date>
```

```
<date>
  <month>July</month>
  <year>1457</year>
</date>
```

Name

dayOfMonth -- An ordinal day of the month

dayOfMonth

Synopsis

Content Model

```
(#PCDATA)
```

Attributes

Name	Type	Default
targets	CDATA	*None*

Description

Represents a specific day in a month. Its value is numeric; the first day of the month is written `1`, the fifteenth `15`, etc.

Parents

`dayOfMonth` may be contained in these elements: `date`

Examples

For examples, see `date`.

Name

degree -- Container for information about a degree or similar certification

degree

Synopsis

Content Model

```
(level,annotation?,major*,minor*,(date|period)?,(institution,location?)?,gpa?,subjects?,projects?)
```

Attributes

Name	Type	Default
id	ID	*None*
targets	CDATA	*None*

Description

The `degree` element is a container for information about an academic degree (Bachelor of Science, Doctor of Psychology, etc.) or other similar certification.

Parents

`degree` may be contained in these elements: `degrees`

Attributes

`id` A string used to uniquely identify a degree. The degree may then be referred to by this identifier.

Examples

```
<degree>
  <level>Ph.D</level>
  <major>Microbiology</major>
  <annotation>Thesis on effect of hot dogs on canine amino acids</annotation>
</degree>

<degree>
  <level>BA</level>
  <major>Toilet Paper Pattern Design</major>
  <date>
    <month>June</month>
    <year>1745</year>
  </date>
  <institution>British Design Academy</institution>
  <annotation>
    On dean's list for duration of studies.
  </annotation>
</degree>

<degree>
  <level>BS</level>
  <major>Computer Science</major>
  <major>English</major>
  <minor>Graphic Design</minor>
  <minor>Geography</minor>
  <subjects>
    <subject>
      <title>Compilers</title>
      <result>A-</result>
    </subject>
    <subject>
      <title>Data Structures</title>
      <result>B</result>
    </subject>
    <subject>
      <title>Classic Literature</title>
      <result>A</result>
    </subject>
  </subjects>
</degree>
```

Name

degrees -- Container for one or more degrees

degrees

Synopsis

Content Model

```
(degree+)
```

Attributes

Name	Type	Default
targets	CDATA	*None*

Description

The `degrees` element is a container for at least one `degree`.

Parents

`degrees` may be contained in these elements: `academics`

Examples

```
<degrees>
  <degree>
    <level>BA</level>
    <major>Interstellar Basket Weaving</major>
  </degree>
  <degree>
    <level>Ph.D</level>
    <major>Aquatic Tennis Teaching</major>
  </degree>
</degree>
```

Name

description -- An explanation of something

description

Synopsis

Content Model

`(para+)`

Attributes

Name	Type	Default
targets	CDATA	*None*

Description

The `description` element specifies a person's role, their activities, or other descriptive information.

Parents

`description` may be contained in these elements: `award`, `interest`, `job`, `membership`

See Also

`projects`, `achievements`.

Examples

```
<job>
  <jobtitle>Supersonic Vacuum Technician</jobtitle>
  <employer>Joe's Vacuum Shop</employer>
  <period>
    <from><date><month>June</month><year>2344</year></date></from>
    <to><present/></to>
  </period>
  <description>
    Repaired hyperdynamic microbial particle accelerators in Supersonic vacuum
    models XL144 and XP2000.
  </description>
</job>

<membership>
  <title>Treasurer</title>
  <organization>Mars Vacuum Repair Association</organization>
  <description>
    Kept books and collected membership dues.
  </description>
</job>
```

Name

docpath --

docpath

Synopsis

Content Model

```
(head?,node*,tail)
```

Attributes

Name	Type	Default
targets	CDATA	*None*

Description

Important

This element is deprecated as of XML Résumé Library version 1.4.0. It should not be used, and will be removed in a future version.

Parents

`docpath` may be contained in these elements: `resume`

Examples

```
<!-- TODO -->
```

Name

email -- An e-mail address

email

Synopsis

Content Model

```
(#PCDATA)
```

Attributes

Name	Type	Default
targets	CDATA	*None*

Description

The `email` element specifies a single e-mail address.

Parents

`email` may be contained in these elements: `contact`

See Also

`fax`, `instantMessage`, `pager`, `url`.

Examples

```
<contact>
  <email>john@johnny.bob</email>
  <phone>555-555-1212</phone>
</contact>
```

Name

emphasis -- An emphasized block of text

emphasis

Synopsis

Content Model

`(#PCDATA)`

Attributes

Name	**Type**	**Default**
targets	CDATA	*None*

Description

The `emphasis` element is an inline element that indicates that the text it contains should be emphasized.

Presentation

The presentation of text contained in this element varies depending on the output format.

HTML
: Rendered as bold (using the `<strong>` tag) by default. Presentation may be overridden by creating or modifying a CSS stylesheet that formats the `emphasis` class.

Text
: Text is surrounded in *astrisks* by default. May be overridden by specifying the *`text.emphasis.start`* and *`text.emphasis.end`* parameters when processing the résumé with an XSLT processor like Xalan or Saxon.

XSL-FO/PDF
: Rendered as bold text.

Parents

`emphasis` may be contained in these elements: `achievement`, `employer`, `institution`, `organization`, `para`, `project`, `skill`

Examples

```
<para>
  I really, really, <emphasis>really</emphasis> want a job.
</para>

<skill>
  I am <emphasis>extremely</emphasis> at using rocket-powered cheese graters.
</skill>
```

Name

employer -- A name of an employer

employer

Synopsis

Content Model

```
(#PCDATA|emphasis|citation|url|link)*
```

Attributes

Name	**Type**	**Default**
id	ID	*None*
targets	CDATA	*None*

Description

The `employer` element specifies the name of a person, company, or other entity for whom a person works or has worked.

Parents

`employer` may be contained in these elements: `job`

Attributes

`id` A string used to uniquely identify an employer. The employer may then be referred to by this identifier.

See Also

`institution`, `organization`.

Examples

```
<job>
  <jobtitle>Angelfish Fitness Trainer</jobtitle>
  <employer>Fish Fitness Systems, Inc.</employer>
  <period>
    <from><date><month>July</month><year>1998</year></date</from>
    <to><present/></to>
  </period>
  <description>
    <para>
      Helped to ensure optimum fish health by training fish on weights and
      and fin and tail exercises.
    </para>
  </description>
</job>

<job>
  <jobtitle>House Cleaner</jobtitle>
  <employer>Mrs. Shirlock</employer>
  <period>
```

```
    <from><date><month>July</month><year>1954</year></date</from>
    <to><date><month>January</month><year>1956</year></date</to>
  </period>
  <para>Dusted the blinds. That is all.</para>
</job>
```

Name

fax -- A fax telephone number

fax

Synopsis

Content Model

```
(#PCDATA)
```

Attributes

Name	Type	Default
location	Enumeration: Enumeration:	*None*
targets	CDATA	*None*

Description

The `fax` element contains a telephone number that can be used to connect to a facsimile machine. The phone number is not required to be in a specific format. (See `phone` for examples of numbers.)

Parents

`fax` may be contained in these elements: `contact`

Attributes

`location` Indicates the location that the phone number connects to. This attribute is optional.

See Also

`email`, `pager`, `phone`.

Examples

```
<fax>555-555-1212</fax>
```

```
<fax location="work">(123) 456-7890</fax>
```

```
<fax location="home">345.324.2721</fax>
```

Name

firstname -- A person's given name

firstname

Synopsis

Content Model

```
(#PCDATA)
```

Attributes

Name	**Type**	**Default**
targets	CDATA	*None*

Description

The `firstname` element specifies a person's first, or given, name. For example, in the name “Mr. Thomas B. Jones”, the first name is “Thomas”.

Parents

`firstname` may be contained in these elements: `name`

See Also

`middlenames`, `surname`.

Examples

```
<name>
  <firstname>Caroline</firstname>
  <surname>Francisca</surname>
</name>
```

Name

from -- The beginning point in a period of time

from

Synopsis

Content Model

```
(date|present)
```

Attributes

Name	**Type**	**Default**
targets	CDATA	*None*

Description

The `from` element specifies the instant at which a given period of time begins. It is always followed by a `to` element in a `period`.

Parents

`from` may be contained in these elements: `period`

See Also

`to`.

Examples

```
<period>
  <from>
    <date>
      <dayOfMonth>26</dayOfMonth>
      <month>December</month>
      <year>1936</year>
    </date>
  </from>
  <to>
    <date>
      <dayOfMonth>14</dayOfMonth>
      <month>January</month>
      <year>2015</year>
    </date>
  </to>
</period>
```

Name

gpa -- Information about a grade point average

gpa

Synopsis

Content Model

```
(score,possible?,note?)
```

Attributes

Name	Type	Default
targets	CDATA	*None*
type	Enumeration: Enumeration:	`overall`

Description

The `gpa` element contains a information about a grade point average, or GPA. A GPA is a number that averages the scores of different subjects (such as mathematics, English, history, etc.).

Parents

`gpa` may be contained in these elements: `degree`

Attributes

`type` Indicates whether the GPA refers to all courses taken toward a degree, or just those in the major field of study. Possible values: `overall` and `major`. Default is `overall`.

Examples

```
<gpa type="major">
  <score>3.78</score>
  <possible>4.00</possible>
  <note>This is on a weighted scale.</note>
</gpa>

<gpa>
  <score>3</score>
</gpa>
```

Name

head --

head

Synopsis

Content Model

```
(label,uri)
```

Attributes

Name	**Type**	**Default**
targets	CDATA	*None*

Description

Important

This element is deprecated as of XML Résumé Library version 1.4.0. It should not be used, and will be removed in a future version.

Parents

`head` may be contained in these elements: `docpath`

Examples

```
<!-- TODO -->
```

Name

header -- Container for information about the person being described in a résumé

header

Synopsis

Content Model

```
(name,address?,birth?,contact?)
```

Attributes

Name	**Type**	**Default**
targets	CDATA	*None*

Description

The `header` element is a container for information typically contained in the header at the top of a résumé. This includes biographic and contact information about the person who the résumé describes.

Formatting

The *`header.format`* parameter influences the formatting of headers in all output formats.

Parents

`header` may be contained in these elements: `resume`

See Also

`header.format`.

Examples

```
<header>
  <name>
    <firstname>Hector</firstname>
    <surname>Gomez</surname>
  </name>
  <address>
    <street>245 San Bernadino</street>
    <city>Los Gatos</city>
    <state>LX</state>
    <zip>94262</zip>
  </address>
  <birth>
    <date>
      <dayOfMonth>30</dayOfMonth>
      <month>April</month>
      <year>1942</year>
    </date>
  </birth>
  <contact>
    <phone>555-1212</phone>
    <email>hector.gomez@xyz.bob</email>
  </contact>
```

```
</header>
```

Name

history -- Container for a person's previous (and possibly current) jobs

history

Synopsis

Content Model

```
(job+)
```

Attributes

Name	Type	Default
targets	CDATA	*None*

Description

The `history` element contains a person's job history (the jobs that they have worked at in the past). It is also common to include their current job in this element.

Parents

`history` may be contained in these elements: `resume`

Examples

```
<history>
  <job>
    <jobtitle>Counselor</jobtitle>
    <employer>Shrink Inc.</employer>
    <period>
      <from><date><month>August</month><year>1490</year></date></from>
      <to><date><month>January</month><year>1497</year></date></to>
    </period>
    <description>
      <para>
        Counseled King Ferdinand and Queen Isabella. Helped to calm their fears
        about their large investment in a man named Christopher Columbus.
      </para>
      <para>
        When Columbus returned with gold, I was "downsized" by beheading.
      </para>
    </description>
  </job>
  <job>
    <jobtitle>Angel</jobtitle>
    <employer>Heaven</employer>
    <period>
      <from><date><month>January</month><year>1497</year></date></from>
      <to><date><month>June</month><year>2344</year></date></to>
    </period>
    <description>
      <para>
        Saved souls.
      </para>
    </description>
  </job>
  <job>
```

```
    <jobtitle>Supersonic Vacuum Technician</jobtitle>
    <employer>Joe's Vacuum Shop</employer>
    <period>
      <from><date><month>June</month><year>2344</year></date></from>
      <to><present/></to>
    </period>
    <description>
      <para>
        Allowed to return to earth to help my great-great-great-(you get the
        picture)-grandson Joe with his business.
      </para>
      <para>
        Repaired hyperdynamic microbial neutron particle accelerators in
        Supersonic vacuum models XL144 and XP2000.
      </para>
    </description>
  </job>
</history>
```

Name

instantMessage -- An instant message username or address

instantMessage

Synopsis

Content Model

```
(#PCDATA)
```

Attributes

Name	**Type**	**Default**
service	CDATA	*None*
targets	CDATA	*None*

Description

The `email` element contains a username or address that can be used to contact a person via an internet instant messaging service, such as AOL Instant Messenger (AIM) or Jabber.

Parents

`instantMessage` may be contained in these elements: `contact`

Attributes

`service` Specifies the service that the username or address is to be used with. Acceptable values are:

`aim`	AOL Instant Messenger
`icq`	ICQ
`irc`	Internet Relay Chat
`jabber`	Jabber
`msn`	MSN Messenger
`yahoo`	Yahoo! Messenger

See Also

`email`.

Examples

```
<instantMessage service="yahoo">joesmith</instantMessage>
```

```
<instantMessage service="aim">janedoe</instantMessage>
```

```
<instantMessage service="jabber">sammybob@jabber.com</instantMessage>
```

Name

institution -- A name of an academic institution

institution

Synopsis

Content Model

```
(#PCDATA|emphasis|citation|url|link)*
```

Attributes

Name	**Type**	**Default**
id	ID	*None*
targets	CDATA	*None*

Description

The `institution` element specifies the name of a degree-awarding institution, such as a university, college, or trade school.

Parents

`institution` may be contained in these elements: `degree`

Attributes

`id` A string used to uniquely identify an institution. The institution may then be referred to by this identifier.

See Also

`organization`.

Examples

```
<degree>
  <level>BA</level>
  <major>Daisy Chain Manufacturing</major>
  <institution>Harvard</institution>
</degree>
```

Name

interest -- Something a person is interested in

interest

Synopsis

Content Model

```
(title,description?)
```

Attributes

Name	Type	Default
targets	CDATA	*None*

Description

The `interest` element describes something that a person is interested in. A common use for `interest` elements is to list hobbies.

Formatting

The formatting of `interest` is more complex than that of many other elements. The title is always displayed first, in standard font. Then, if there is a description, and the *`interest.description.format`* parameter value is `single-line`, a period and a space (“. ”) are displayed. Finally, the description is output, formatted according to the *`interest.description.format`* parameter.

Parents

`interest` may be contained in these elements: `interests`

See Also

`skill`.

Examples

```
<interest>
  <title>Sewing</title>
</interest>

<interest>
  <title>Seattle Mariners</title>
  <description>
    <para>I'm a season ticket holder.</para>
    <para><emphasis>GO MARINERS!</emphasis></para>
  </description>
</interest>
```

For additional examples, see `interests`.

Name

interests -- Container for one or more interests

interests

Synopsis

Content Model

```
(title?,interest+)
```

Attributes

Name	Type	Default
targets	CDATA	*None*

Description

The `interests` element contains one or more `interest` elements.

Formatting

The `interests` element is formatted as a top-level section in a résumé. The ***`interests.word`*** parameter is used as the title text if the `interests` doesn't contain a `title` element.

Child `interest` elements are formatted as a bulleted list in all output formats.

Parents

`interests` may be contained in these elements: `resume`

See Also

`skillarea`.

Examples

```
<interests>
  <title>Hobbies</title>
  <interest>
    <title>Swimming</title>
  </interest>
  <interest>
    <title>Fine foods</title>
    <description>
      <para>I do quite well in pie-eating contests.</para>
    </description>
  </interest>
  <interest>
    <title>Australia</title>
  </interest>
</interests>

<interests>
  <interest>
    <title>My belly button</title>
  </interest>
  <interest>
    <title>Fire ants</title>
  </interest>
  <interest>
    <title>Fertilizer</title>
  </interest>
</interests>
```

Name

job -- A specific employment engagement

job

Synopsis

Content Model

```
(jobtitle,employer,location?,(date|period),description?,projects?,achievements?)
```

Attributes

Name	Type	Default
id	ID	*None*
targets	CDATA	*None*

Description

The `job` element describes a job, contract, or other employment engagement.

Parents

`job` may be contained in these elements: `history`

Attributes

`id` A string used to uniquely identify a job. The job may then be referred to by this identifier.

Examples

```
<job>
  <jobtitle>Sr. VP of Soap Suds Engineering</jobtitle>
  <employer>Proctor and Gamble</employer>
  <period>
    <from><date>
      <month>May</month>
      <year>1982</year>
    </date></from>
    <to><date>
      <month>June</month>
      <year>1988</year>
    </date></to>
  </period>
</job>

<job>
  <jobtitle>Dirt</jobtitle>
  <employer>Earth</employer>
  <period>
    <from><date>
      <month>Januaray</month>
      <year>30,000 B.C.</year>
    </date></from>
    <to><present/></to>
  </period>
  <description>
```

```
    Provided nutrients, absorbed water, and got tracked into houses.
  </description>
  <achievements>
    <achievement>
      Grew a 300-ft. tree.
    </achievement>
  </achievements>
  <projects>
    <project>
      Took part in the construction of the construction of the Empire State
      Building in New York City. (Was excavated from its basement.)
    </project>
  </projects>
</job>
```

Name

jobtitle -- A job title

jobtitle

Synopsis

Content Model

```
(#PCDATA)
```

Attributes

Name	**Type**	**Default**
targets	CDATA	*None*

Description

The `jobtitle` element specifies the offical title or name of the position a person held at a job.

Parents

`jobtitle` may be contained in these elements: `job`

Examples

For examples, see job

Name

keyword -- A keyword to be used for résumé indexing and searching

keyword

Synopsis

Content Model

```
(#PCDATA)
```

Attributes

Name	**Type**	**Default**
targets	CDATA	*None*

Description

The `keyword` element specifies a word or phrase that is to be used to index and/or search a résumé, usually on a computer.

Keywords are often used to specify “buzzwords” that are not (or that are) in the main text of the résumé, but may be used by employers or recruiters to find résumés.

Parents

`keyword` may be contained in these elements: `keywords`

Examples

```
<keywords>
  <keyword>forward-thinking</keyword>
  <keyword>self-starter</keyword>
  <keyword>independent</keyword>
  <keyword>flexible</keyword>
</keywords>
```

Name

keywords -- Container for a list of keywords

keywords

Synopsis

Content Model

```
(keyword+)
```

Attributes

Name	**Type**	**Default**
targets	CDATA	*None*

Description

The `keywords` element contains one or more keywords. These keywords are included in the HTML presentation as `<meta name="keywords" content="...">` tags.

Parents

`keywords` may be contained in these elements: `resume`

Examples

For examples, see `keyword`.

Name

label --

label

Synopsis

Content Model

```
(#PCDATA)
```

Attributes

Name	**Type**	**Default**
targets	CDATA	*None*

Description

Important

This element is deprecated as of XML Résumé Library version 1.4.0. It should not be used, and will be removed in a future version.

Parents

`label` may be contained in these elements: `head`, `node`

Examples

```
<!-- TODO -->
```

Name

lastModified -- Information about when a résumé was last modified

lastModified

Synopsis

Content Model

```
(date)
```

Attributes

Name	Type	Default
targets	CDATA	*None*

Description

The `lastModified` element contains the date that a résumé was last updated.

Parents

`lastModified` may be contained in these elements: `resume`

Examples

```
<lastModified>
  <date>
    <dayOfMonth>4</dayOfMonth>
    <month>July</month>
    <year>1796</year>
  </date>
</lastModified>
```

Name

legalnotice -- A legal statement

legalnotice

Synopsis

Content Model

```
(para+)
```

Attributes

Name	**Type**	**Default**
targets	CDATA	*None*

Description

The `legalnotice` element contains text that is intended to specify the legal use of a résumé.

Parents

`legalnotice` may be contained in these elements: `copyright`

Examples

For examples, see `copyright`

Name

level -- A degree type

level

Synopsis

Content Model

```
(#PCDATA)
```

Attributes

Name	**Type**	**Default**
targets	CDATA	*None*

Description

The semantic of the `title` element vary depending on which element contains it:

`clearance`
: When contained in this element, the `level` element specifies the level of security clearance that a person has been granted, such as “Top Secret” or “Restricted”.

`degree`
: When contained in this element, the `level` element specifies a degree type, such as “BFA” (bachelor of fine arts) or “Ph.D” (doctor of philosophy). It typically contains the abbreviation (“BS”) not the full spelling (“bachelor of science”).

Parents

`level` may be contained in these elements: `clearance`, `degree`

Examples

For examples, see `clearance` and `degree`.

Name

link -- A titled hyperlink

link

Synopsis

Content Model

`(#PCDATA)`

Attributes

Name	**Type**	**Default**
href	CDATA	*None*
targets	CDATA	*None*

Description

The `link` element defines a single hyperlink, much like HTML's `a` (anchor) element.

Formatting

In FO/PDF and plain text output formats, the text contained in the `link` element is formatted as if the surrounding `link` tags were not present. The `href` attribute is ignored.

In the HTML output format, the `link` element is formatted as an `a` (anchor) hyperlink.

Parents

`link` may be contained in these elements: `achievement`, `artTitle`, `bookTitle`, `employer`, `institution`, `organization`, `para`, `project`, `publisher`, `skill`

Attributes

`href` The URL that the link should point to. then be referred to by this identifier.

See Also

`uri`, `url`.

Examples

```
<employer>
  <link href="http://www.acme.bob/">Acme Corporation</link>
</employer>

<para>
  The <link href="http://xmlresume.sf.net/">XML R&eacute;sum&eacute;
  Project</link> supplies tools that easily convert a single XML
  r&eacute;sum&eacute; to multiple output formats.
</employer>
```

Name

location -- A location, with city-level granularity

location

Synopsis

Content Model

```
(city?,(state|province|county)?,country?)
```

Attributes

Name	**Type**	**Default**
targets	CDATA	*None*

Description

The `location` element specifies a level location, such as “London, England”, “Seattle, Washington”, “St. Louis, Missouri, USA”, or “Newfoundland, Canada”.

Parents

`location` may be contained in these elements: `degree`, `job`, `membership`

See Also

`address`.

Examples

```
<location>
  <city>Los Angeles<city>
  <state>California</state>
  <country>USA</country>
</location>
```

```
<location>
  <city>Paris<city>
  <country>France</country>
</location>
```

Name

major -- A main course of study

major

Synopsis

Content Model

```
(#PCDATA)
```

Attributes

Name	**Type**	**Default**
targets	CDATA	*None*

Description

The `major` element names the main (“major”) course of study in a degree. A major is the area that is focused on the most by a student.

Parents

`major` may be contained in these elements: `degree`

See Also

`minor`.

Examples

For examples, see `degree`.

Name

membership -- A professional membership

membership

Synopsis

Content Model

```
(title?,(organization,location?)?,(date|period)?,description?)
```

Attributes

Name	**Type**	**Default**
id	ID	*None*
targets	CDATA	*None*

Description

The `membership` element contains information about a person's membership in a professional organization.

Parents

`membership` may be contained in these elements: `memberships`

Attributes

`id` A string used to uniquely identify a membership. The membership may then be referred to by this identifier.

Examples

For examples, see `memberships`.

Name

memberships -- Container for one or more memberships

memberships

Synopsis

Content Model

```
(title,membership+)
```

Attributes

Name	Type	Default
targets	CDATA	*None*

Description

The `memberships` element contains one or more `memberships`.

Parents

`memberships` may be contained in these elements: `resume`

Examples

```
<memberships>
  <title>Professional Memberships</title>
  <membership>
    <organization>Foobar State Technology Association</organization>
  </membership>
  <membership>
    <title>President</title>
    <organization>Smallville Internet Developers' Group</organization>
    <period>
      <from><date>
        <month>September</month>
        <year>1998</year>
      </date></from>
      <to><present/></to>
    </period>
    <description>
      <para>
        Founded group and guided its growth from 4 to 150 members.
      </para>
    </description>
  </membership>
</memberships>
```

Name

middlenames -- One or more “middle” names of a person

middlenames

Synopsis

Content Model

```
(#PCDATA)
```

Attributes

Name	Type	Default
targets	CDATA	*None*

Description

The `middlenames` element contains all of the names of a person, excluding their first (given) name and surname. If the person has multiple middle names, they should be space-separated.

Parents

`middlenames` may be contained in these elements: `name`

See Also

`firstname`, `surname`.

Examples

```
<middlenames>Samione</middlenames>
```

```
<middlenames>Lu&iacute;s Garcia M&aacute;rquez</middlenames>
```

Name

minor -- A minor course of study

minor

Synopsis

Content Model

```
(#PCDATA)
```

Attributes

Name	Type	Default
targets	CDATA	*None*

Description

The `major` element names the main (“major”) course of study in a degree. A major is the area that is focused on the most by a student.

Parents

`minor` may be contained in these elements: `degree`

See Also

`major`.

Examples

For examples, see `degree`.

Name

misc -- Miscellaneous remarks

misc

Synopsis

Content Model

`(para+)`

Attributes

Name	Type	Default
targets	CDATA	*None*

Description

The `misc` element contains paragraphs of text that don't fit into any other top-level categories.

Parents

`misc` may be contained in these elements: `resume`

Examples

```
<misc>
  <para>What can't be captured on a r&eacute;sum&eacute; is my sheer
    level of <emphasis>enthusiasm</emphasis> for not just cooking
    food, but dealing with people.  I'm told I make nearly every
    kitchen run more smoothly because I know how to manage and make
    people want to <emphasis>do their best</emphasis>.
  </para>
  <para>I'm listed in <citation>Who's Who in American
      Restaurants</citation> as well as <citation>La
      Rubach&ograve;n</citation>.
  </para>
</misc>
```

Name

month -- A month name

month

Synopsis

Content Model

```
(#PCDATA)
```

Attributes

Name	**Type**	**Default**
targets	CDATA	*None*

Description

The `month` element contains the name of a month, such as “March” (English), “marzo” (Spanish), or “mars” (French).

Parents

`month` may be contained in these elements: `date`, `pubDate`

See Also

`dayOfMonth`, `year`.

Examples

```
<month>April</month>
```

Name

name -- A person's name

name

Synopsis

Content Model

```
(title?,firstname,middlenames?,surname,suffix?)
```

Attributes

Name	Type	Default
id	ID	*None*
targets	CDATA	*None*

Description

The `name` element contains a person's name.

Parents

`name` may be contained in these elements: `copyright`, `header`, `referee`

Attributes

`id` A string used to uniquely identify a name. The name may then be referred to by this identifier.

Examples

```
<name>
  <title>Mr.</title>
  <firstname>John</firstname>
  <middlenames>Q.</middlenames>
  <surname>Doe</surname>
</name>

<name>
  <firstname>Joe</firstname>
  <middlenames>Sam George Bob</middlenames>
  <surname>Flamboyzo</surname>
  <suffix>Jr.</suffix>
</name>

<name>
  <firstname>King</firstname>
  <surname>Louis</surname>
  <suffix>VIII</suffix>
</name>
```

Name

node --

node

Synopsis

Content Model

`(label,uri)`

Attributes

Name	**Type**	**Default**
targets	CDATA	*None*

Description

Important

This element is deprecated as of XML Résumé Library version 1.4.0. It should not be used, and will be removed in a future version.

Parents

`node` may be contained in these elements: `docpath`

Examples

```
<!-- TODO -->
```

Name

note -- Additional information

note

Synopsis

Content Model

`(para+)`

Attributes

Name	Type	Default
targets	CDATA	*None*

Description

The `note` element contains additional information about its parent element.

Parents

`note` may be contained in these elements: `academics`, `clearance`, `gpa`

See Also

`annotation`, `legalnotice`.

Examples

```
<academics>
  <degrees>
    ...
  </degrees>
  <note>
    <para>
      I take education very seriously. After all, I'm in 37th grade now!
    </para>
  </note>
</academics>
```

Name

objective -- A person's employment goal

objective

Synopsis

Content Model

`(para+)`

Attributes

Name	Type	Default
id	ID	*None*
targets	CDATA	*None*

Description

The `objective` element contains a statement of a person's goal. It commong specifies the industry or position in which a person would like to work.

Parents

`objective` may be contained in these elements: `resume`

Attributes

`id` A string used to uniquely identify an objective. The objective may then be referred to by this identifier.

Examples

```
<objective>
  <para>
    I seek a high-paying job in the golf playing industry. I would like at
    least 20 weeks of vacation per year.
  </para>
</objective>
```

Name

organization -- A name of an organization

organization

Synopsis

Content Model

```
(#PCDATA|emphasis|citation|url|link)*
```

Attributes

Name	**Type**	**Default**
id	ID	*None*
targets	CDATA	*None*

Description

The `organization` element contains the name of a professional or other type of organization.

Parents

`organization` may be contained in these elements: `award`, `clearance`, `membership`, `referee`

Attributes

`id` A string used to uniquely identify an organization. The organization may then be referred to by this identifier.

See Also

`institution`.
-->

Examples

```
<membership>
  <organization>Croquet Players' Association of the UK</organization>
</membership>
```

Name

pageNums -- One or more page numbers or ranges

pageNums

Synopsis

Content Model

```
(#PCDATA)
```

Attributes

Name	**Type**	**Default**
targets	CDATA	*None*

Description

The `pageNums` element specifies one or more page numbers or ranges of page numbers, and is used in citing a publication.

Parents

`pageNums` may be contained in these elements: `pub`

Examples

```
<pub>
  <artTitle>Ancient Mayan Anthropology</artTitle>
  <bookTitle>Nature</bookTitle>
  <pageNums>13-54</pageNums>
</pub>

<pub>
  <artTitle>The New Fad: Bottle Cap Collecting</artTitle>
  <bookTitle>Time Magazine</bookTitle>
  <pageNums>14</pageNums>
</pub>
```

Name

pager -- A pager telephone number

pager

Synopsis

Content Model

```
(#PCDATA)
```

Attributes

Name	**Type**	**Default**
targets	CDATA	*None*

Description

The `pager` element contains a telephone number that can be used to connect to a pager. The phone number is not required to be in a specific format. (See `phone` for examples of numbers.)

Parents

`pager` may be contained in these elements: `contact`

See Also

`email`, `fax`, `phone`.

Examples

```
<pager>456-145-4468</pager>
```

```
<pager>736.272.4673 x 235</pager>
```

Name

para -- A paragraph of text

para

Synopsis

Content Model

```
(#PCDATA|emphasis|citation|url|link)*
```

Attributes

Name	**Type**	**Default**
targets	CDATA	*None*

Description

The `para` element contains a single paragraph of text. Its contents are not preserved verbatim; instead, it is treated much like HTML. Multiple whitespace characters (including spaces and tabs) are compressed into a single space, and linebreaks are treated as space characters instead of newlines. This allows flexibility in formatting XML source code.

Paragraphs may contain basic markup to indicate emphasis, URLs, and citations.

Parents

`para` may be contained in these elements: `description`, `legalnotice`, `misc`, `note`, `objective`, `pub`

Examples

```
<para>
  This is a boring paragraph.
</para>

<para>
  I like to <emphasis>emphasize</emphasize> my words; in fact, I've written an
  article on the topic. It is published on the <emphasis><citation>Really
  Exciting WRITING!</citation></emphasis> web site at
  <url>http://www.really-exciting-writing.bob/EMPHASIS.html</url>.
</para>
```

Name

period -- A period of time

period

Synopsis

Content Model

```
(from,to)
```

Attributes

Name	Type	Default
targets	CDATA	*None*

Description

The `period` element defines a specific period in time by specifying the period's start time and end time.

If you want to specify an instant in time instead of a period, use `date`.

Parents

`period` may be contained in these elements: `award`, `clearance`, `degree`, `job`, `membership`

See Also

`element`.

Examples

```
<period>
  <from><date>
    <dayOfMonth>10</dayOfMonth>
    <month>February</month>
    <year>1246</year>
  </date></from>
  <to><date>
    <dayOfMonth>19</dayOfMonth>
    <month>November</month>
    <year>1351</year>
  </date></to>
</period>

<period>
  <from><date>
    <month>May</month>
    <year>1985</year>
  </date></from>
  <to>
    <present/>
  </to>
</period>
```

Name

phone -- A voice telephone number

phone

Synopsis

Content Model

```
(#PCDATA)
```

Attributes

Name	Type	Default
location	Enumeration: Enumeration:	*None*
targets	CDATA	*None*

Description

The `phone` element contains a telephone number. The number is not required to be in a specific format; these are all valid phone numbers:

- `555-555-1212`
- `555.555.1212`
- `(555) 555-1212`
- `(555) 555-1212 x555`
- `(011) 47 8931-858128`
- `(011) 59-3331-8580`

Parents

`phone` may be contained in these elements: `contact`

Attributes

`location` Indicates the location that the phone number connects to. This attribute is optional.

See Also

`address`, `email`, `fax`, `pager`.

Examples

```
<phone>555.555.1212</phone>
```

```
<phone location="mobile">345-262-4567</phone>
```

```
<phone location="home">(261) 345-1616</phone>
```

For additional examples, see `contact`.

Name

possible -- The highest possible score in a GPA

possible

Synopsis

Content Model

```
(#PCDATA)
```

Attributes

Name	Type	Default
targets	CDATA	*None*

Description

The `possible` element contains a number that indicates the highest possible score that can be attained in a GPA. In other words, a “perfect” score.

Parents

`possible` may be contained in these elements: `gpa`

See Also

`score`.

Examples

For examples, see `gpa`.

Name

postalCode -- A postal code

postalCode

Synopsis

Content Model

```
(#PCDATA)
```

Attributes

Name	**Type**	**Default**
id	ID	*None*
targets	CDATA	*None*

Description

The `postalCode` element contains an alphanumeric string used by postal services to route mail to its destination.

For postal codes in the United States, use `zip`.

Parents

`postalCode` may be contained in these elements: `address`

See Also

`zip`.

Examples

A Brazilian postal code:

```
<postalCode>85070-200</postalCode>
```

A Canadian postal code:

```
<postalCode>V2B 5S8</postalCode>
```

Name

prefecture -- A name of a prefecture or other administrative district

prefecture

Synopsis

Content Model

```
(#PCDATA)
```

Attributes

Name	Type	Default
id	ID	*None*
targets	CDATA	*None*

Description

The `prefecture` element contains the name of a prefecture or other administrative district of a city, province, state, or other area.

This element is currently ignored during formatting.

Parents

`prefecture` may be contained in these elements: `address`

Attributes

`id` A string used to uniquely identify a prefecture. The prefecture may then be referred to by this identifier.

See Also

`city`, `county`, `state`, `suburb`, `ward`.

Examples

```
<prefecture>Basilius</prefecture>
```

Name

present -- Indicates the current time

present

Synopsis

Content Model

`EMPTY`

Attributes

Name	**Type**	**Default**
targets	CDATA	*None*

Description

The `present` element denotes the current time, as opposed to a fixed time in the past or future. It is most commonly used in date ranges, to express things like “From May 1995 to present”.

`present` is a null element, that is, it has no content or attributes. It is always written as `<present/>`.

Parents

`present` may be contained in these elements: `from`, `to`

See Also

`date`.

Examples

```
<period>
  <from><date>
    <month>May</month>
    <year>1995</year>
  </from>
  <to>
    <present/>
  </to>
</period>
```

Name

project -- Container for information about a project

project

Synopsis

Content Model

```
(#PCDATA|emphasis|citation|url|link)*
```

Attributes

Name	**Type**	**Default**
targets	CDATA	*None*
title	CDATA	*None*

Description

The `project` element describes a project that a person participated in at a `job`.

Parents

`project` may be contained in these elements: `projects`

Attributes

`title` The title of the project.

See Also

`achievement`.

Examples

For examples, see `projects`.

Name

projects -- Container for one or more projects

projects

Synopsis

Content Model

```
(project+)
```

Attributes

Name	Type	Default
targets	CDATA	*None*

Description

The `projects` element groups together one or more `project` elements.

Parents

`projects` may be contained in these elements: `degree`, `job`

See Also

`achievements`.

Examples

```
<projects>
  <project>Organization of Paperclips</project>
  <project>Development of rocket booster fuel</project>
  <project>1997 Papercut Convention</project>
</projects>
```

Name

province -- A name or abbreviation of a province

province

Synopsis

Content Model

```
(#PCDATA)
```

Attributes

Name	**Type**	**Default**
id	ID	*None*
targets	CDATA	*None*

Description

The `province` element contains the name or abbreviation of a single province or similar political division.

Parents

`province` may be contained in these elements: `address`, `location`

Attributes

`id` A string used to uniquely identify a province. The province may then be referred to by this identifier.

See Also

`county`, `state`.

Examples

```
<province>Saskatchewan</province>
```

```
<province>BC</province>
```

Name

pub -- A work published by the résumé “owner”

pub

Synopsis

Content Model

```
(para|(artTitle|bookTitle|author|date|pubDate|publisher|pageNums|url))*
```

Attributes

Name	Type	Default
id	ID	*None*
targets	CDATA	*None*

Description

The pub element contains elements that describe a work (like an article, a book, a poem, or a scholarly paper) that was published by the person the résumé describes.

Parents

pub may be contained in these elements: pubs

Attributes

id A string used to uniquely identify a publication. The publication may then be referred to by this identifier.

See Also

citation.

Examples

Example 4. Book with multiple authors

```
<pub>
  <bookTitle>Foundations of Computer Science, C Edition</bookTitle>
  <author>Alfred V. Aho</author>
  <author>Jeffrey D. Ullman</author>
  <pubDate>
    <year>1995</year>
  </pubDate>
  <publisher>Computer Science Press, New York</publisher>
</pub>
```

Example 5. Magazine article

```
<pub>
  <artTitle>Low-Fat Philly Cheese Steaks</bookTitle>
  <bookTitle>Healthy Cooking</bookTitle>
  <author>Joe Sixpack</author>
  <pubDate>
    <month>June</month>
    <year>1999</year>
  </pubDate>
  <publisher>Healthy Publishing Group</publisher>
</pub>
```

Name

pubDate -- The date a work was published

pubDate

Synopsis

Content Model

```
(month?,year)
```

Attributes

Name	Type	Default
targets	CDATA	*None*

Description

> ### Important
>
> This element is deprecated as of XML Résumé Library version 1.4.0. It should not be used.
>
> `date` elements should be used instead of `pubDate` elements.

The `pubDate` element indicates when a literary, scholarly, or other work was published. It must contain a `year`, and may contain a `month`.

Parents

`pubDate` may be contained in these elements: `pub`

See Also

`date`.

Examples

```
<pub>
  <bookTitle>How to Drink Water for Fun and Profit</bookTitle>
  <pubdate>
    <year>1996</year>
  </pubdate>
</pub>

<pub>
  <artTitle>Modern Neo-Classical Baroque Music</bookTitle>
  <bookTitle>Journal of Classical Music</bookTitle>
  <pubdate>
    <month>July</month>
    <year>1931</year>
  </pubdate>
</pub>
```

Name

publisher -- Information about a publisher of a work

publisher

Synopsis

Content Model

```
(#PCDATA|link|url)*
```

Attributes

Name	Type	Default
targets	CDATA	*None*

Description

Important

The use of the `url` element as a child of `publisher` is deprecated as of XML Résumé Library version 1.4.0.

We recommend that you convert all `url`s contained in `publisher`s to `link`s.

The `publisher` element contains the name, and possibly location, of the organization that published a literary, artistic, scholary, or other work.

Parents

`publisher` may be contained in these elements: `pub`

Examples

Example 6. Publisher name only

```
<pub>
  <bookTitle>Deep Thoughts on Stuff</bookTitle>
  <publisher>Totally Groovy Publishing, Ltd.</publisher>
</pub>
```

Example 7. Publisher name and location

```
<pub>
  <bookTitle>A User's Guide to Dirt</bookTitle>
  <publisher>Mother Nature Books, Boston, Massachusetts</publisher>
</pub>
```

Name

pubs -- A group of one or more publications

pubs

Synopsis

Content Model

```
(pub+)
```

Attributes

Name	Type	Default
targets	CDATA	*None*

Description

The `pubs` element groups together one or more publications.

Parents

`pubs` may be contained in these elements: `resume`

Examples

```
<pubs>
  <pub>
    <bookTitle>General Relativity for Idiots</bookTitle>
    <author>Einstein</author>
  </pub>
  <pub>
    <artTitle>String Theory for Dummies</artTitle>
    <bookTitle>Topics in Physics</bookTitle>
    <author>Stephen Hawking</author>
  </pub>
</pubs>
```

Name

referee -- Someone who can provide additional information about the person the résumé describes

referee

Synopsis

Content Model

`(name,title?,organization?,address?,contact?)`

Attributes

Name	Type	Default
targets	CDATA	*None*

Description

The `referee` element contains the name and contact information information of a person who knows the job seeker (the person that the résumé describes). Referees (also called references) can provide additional information about the job seeker. They often supply background information, confirm facts, and describe the job seeker's character.

Formatting

The display of `referee` is influenced by the *`referees.display`* parameter.

Parents

`referee` may be contained in these elements: `referees`

Examples

```
<referee>
  <name>Joe</name>
  <address>
    <street>123 Main St.</street>
    <city>Anytown</city>
    <state>ST</state>
    <zip>12345</zip>
    <country>USA</country>
  </address>
  <contact>
    <phone>123.456.7890</phone>
    <email>joe@hotmail.bob</email>
  </contact>
</referee>
```

For additional examples, see `referees`.

Name

referees -- Contains one or more referees

referees

Synopsis

Content Model

`(referee+)`

Attributes

Name	Type	Default
targets	CDATA	*None*

Description

The `referees` element groups together one or more `referee` elements.

Parents

`referees` may be contained in these elements: `resume`

Examples

```
<referees>
  <referee>
    <name>Mr. Samuel J. Black</name>
    <contact>
      <phone>555.555.1212</phone>
    </contact>
  </referee>
  <referee>
    <name>Mom</name>
    <contact>
      <email>my_mommy@moms.bob</email>
    </contact>
  </referee>
  <referee>
    <name>Dad</name>
  </referee>
</referees>
```

Name

result -- An outcome of a subject

result

Synopsis

Content Model

```
(#PCDATA)
```

Attributes

Name	**Type**	**Default**
targets	CDATA	*None*

Description

The `result` element describes the outcome or result of a `subject`. `result` often contains the letter or numeric grade awarded for an academic class.

Parents

`result` may be contained in these elements: `subject`

Examples

```
<subjects>
  <subject>
    <title>Math</title>
    <result>A-</result>
  </subject>
  <subject>
    <title>English</title>
    <result>C</result>
  </subject>
  <subject>
    <title>Science</title>
    <result>B+</result>
  </subject>
  <subject>
    <title>Spanish</title>
    <result>B</result>
  </subject>
</subjects>
```

Name

resume -- A résumé or curriculum vitae

resume

Synopsis

Content Model

```
(docpath?,header?,((objective|history|academics|skillareas|skillarea|pubs|misc|referees|keywords|mem
```

Attributes

Name	Type	Default
id	ID	*None*
targets	CDATA	*None*
xmlns	CDATA	`http://xmlresume.sourceforge.net/resume/0.0`
xmlns:xsi	CDATA	`http://www.w3.org/2001/XMLSchema-instance`
xsi:schemaLocation	CDATA	*None*

Description

The `resume` element represents a single résumé or curriculum vitae (CV). It is a summary of a person's experience that is revelant to employment.

Parents

`resume` may be contained in these elements: `resumes`

Attributes

`id` A string used to uniquely identify a résumé. The résumé may then be referred to by this identifier.

Examples

```
<resume id="a8df262">

  <header>
    <name id="harry.potter">
      <firstname>Harry</firstname>
      <surname>Potter</surname>
    </name>
    <address>
      <street>4 Privet Drive</street>
      <city>Little Whinging</city>
      <county>Surrey</county>
      <postalCode>RO51 5NF</postalCode>
      <country>UK</country>
    </address>
  </header>

  <objective>
```

```
    <para>To defeat Lord Voldemort once and for all, then to become an Auror
    for the Ministry of Magic.</para>
  </objective>

  <skillarea>
    <title>Magical Skills</title>
    <skillset>
      <title>Charms</title>
      <skill>Levitation</skill>
      <skill>Cheering</skill>
      <skill>Summoning</skill>
      <skill>Banishing</skill>
    </skillset>
    <skillset>
      <title>Spells and Curses</title>
      <skill>Unlocking Spell</skill>
      <skill>Patronus Spell</skill>
      <skill>Stunning Spell</skill>
      <skill>Disarming Spell</skill>
      <skill>Reductor Curse</skill>
      <skill>Impediment Curse</skill>
      <skill>Imperious Curse Evasion</skill>
    </skillset>
    <skillset>
      <title>Potions</title>
      <skill>Polyjuice</skill>
      <skill>Enlarging</skill>
      <skill>Antidotes</skill>
    </skillset>
    <skillset>
      <title>Flying</title>
      <skill>High-speed</skill>
      <skill>Wronski Feint</skill>
      <skill>Steep dives</skill>
      <skill>Bludger Evasion</skill>
      <skill>No-hands</skill>
      <skill>High-altitude</skill>
    </skillset>
    <skillset>
      <title>Languages</title>
      <skill>English</skill>
      <skill>Parsel Tounge</skill>
      <skill>Troll (point and grunt)</skill>
    </skillset>
  </skillarea>
  <skillarea>
    <title>Muggle Skills</title>
    <skillset>
      <title>Daily Life</title>
      <skill>Telephone</skill>
      <skill>Mailbox</skill>
      <skill>Riding in cars</skill>
      <skill>Electric lights</skill>
      <skill>Hammering</skill>
    </skillset>
    <skillset>
      <title>Camping</title>
      <skill>Setting up tents</skill>
      <skill>Lighting Matches</skill>
    </skillset>
  </skillarea>

  <history>
    <job>
      <jobtitle>Student</jobtitle>
      <employer>Hogwarts</employer>
      <period>
        <from>
          <date>
            <month>September</month>
            <year>ca. 1995</year>
          </date>
        </from>
        <to>
          <present/>
        </to>
      </period>
      <achievements>
        <achievement>
          On Gryffindor House Quidditch team four years running
        </achievement>
        <achievement>
          Quidditch Cup
        </achievement>
        <achievement>Youngest Seeker in a century</achievement>
        <achievement>
          Faced Lord Voldemort four times and lived.
        </achievement>
      </achievements>
    </job>
    <job>
      <jobtitle>De-gnomer</jobtitle>
      <employer>Mrs. Weasley</employer>
      <period>
        <from>
          <date>
            <month>August</month>
            <year>ca. 1995</year>
          </date>
        </from>
        <to>
          <date>
            <month>August</month>
            <year>ca. 1995</year>
          </date>
        </to>
      </period>
      <description>
        <para>Removed gnomes from the Weasleys' garden.</para>
      </description>
    </job>
  </history>

  <academics>
    <degrees>
      <degree>
        <level>Fourth Year</level>
        <major>Wizarding</major>
        <institution>Hogwards School of Witchcraft and Wizardry</institution>
        <subjects>
          <subject>
            <title>Transfiguration</title>
            <result>B</result>
          </subject>
          <subject>
            <title>Charms</title>
            <result>A-</result>
          </subject>
          <subject>
            <title>Defense Against the Dark Arts</title>
            <result>A</result>
          </subject>
          <subject>
            <title>Potions</title>
            <result>D</result>
          </subject>
          <subject>
            <title>Care of Magical Creature</title>
            <result>A+</result>
          </subject>
          <subject>
            <title>Divination</title>
            <result>B</result>
          </subject>
        </subjects>
      </degree>
    </degrees>
  </academics>

  <memberships>
    <title>School Clubs</title>
    <membership>
      <organization>Dueling Club</organization>
      <description><para>Trained in wizard dueling, with focus on
      disarming.</para></description>
    </membership>
    <membership>
      <title>Secretary</title>
      <organization>Society for the Protection of Elfish Workers
      (S.P.E.W.)</organization>
    </membership>
  </memberships>

  <interests>
    <interest><title>Cho Chang</title></interest>
  </interests>

  <referees>
    <referee>
      <name>
        <firstname>Minerva</firstname>
        <surname>McGonagall</surname>
      </name>
      <address>Hogwarts School of Witchcraft and Wizardry
Via Owl Post</address>
    </referee>
    <referee>
      <name>
        <firstname>Albus</firstname>
        <surname>Dumbledore</surname>
      </name>
      <address>Hogwarts School of Witchcraft and Wizardry
Via Owl Post</address>
    </referee>
  </referees>

</resume>
```

For additional examples, look at the sample résumés in the `examples` directory of the XML Résumé Library distribution.

Name

resumes -- A collection of one or more résumés

resumes

Synopsis

Content Model

```
(resume*)
```

Attributes

Name	Type	Default
id	ID	*None*
targets	CDATA	*None*
xmlns	CDATA	`http://xmlresume.sourceforge.net/resume/0.0`
xmlns:xsi	CDATA	`http://www.w3.org/2001/XMLSchema-instance`
xsi:schemaLocation	CDATA	*None*

Description

The `resumes` element contains one or more résumés. It could be used, for example, to distribute the résumés of all applicants for a job as a single file.

The formatting of this element is currently undefined.

Parents

`resumes` may be contained in these elements: None

Attributes

`id` A string used to uniquely identify a group of résumés. The résumés may then be referred to by this identifier.

Examples

```
<resumes id="resumes.tech.titans">
  <resume id="resume.gates">
    <header>
      <name>
        <firstname>Bill</firstname>
        <surname>Gates</surname>
      </name>
    </header>
  </resume>
  <resume id="resume.jobs">
    <header>
      <name>
        <firstname>Steve</firstname>
        <surname>Jobs</surname>
      </name>
    </header>
  </resume>
</resumes>
```

Name

score -- The score earned in a GPA

score

Synopsis

Content Model

```
(#PCDATA)
```

Attributes

Name	**Type**	**Default**
targets	CDATA	*None*

Description

The `score` element contains the GPA score that a person has earned.

Parents

`score` may be contained in these elements: `gpa`

See Also

`possible`.

Examples

For examples, see `gpa`.

Name

skill -- A name and/or description of a skill

skill

Synopsis

Content Model

`(#PCDATA|emphasis|citation|url|link)*`

Attributes

Name	Type	Default
id	ID	*None*
level	CDATA	*None*
targets	CDATA	*None*

Description

The `skill` element describes something that a person is skilled at.

Parents

`skill` may be contained in these elements: `skills`, `skillset`

Attributes

`level` The `level` attribute indicates a person's level of proficiency at a skill. “Level of proficiency” may be expressed in any manner, such as “3 years” (three years of experience with a particular skill), “expert”, or “7/10”. The *`skills.level.display`* parameter determines whether or not skill levels are displayed in formatted output.

See Also

`achievement`, *`skills.format`*, *`skills.level.display`*.

Examples

```
<skill>Singing</skill>
```

```
<skill level="3 years">Java Programming</skill>
```

```
<skill level="certified glutton">Eating</skill>
```

```
<skill level="beginner">
  Throwing <emphasis>large</emphasis> objects
</skill>
```

Name

skillarea -- A group of broadly related skill sets

skillarea

Synopsis

Content Model

```
(title,skillset+)
```

Attributes

Name	Type	Default
id	ID	*None*
targets	CDATA	*None*

Description

The `skillarea` element contains one or more `skillsets`. Usually the skill sets it contains are related to each other, at least in a general sense.

For example, a “Computer Skills” skill area may contain skillsets titled “Programming”, “Office Programs”, and “Hardware”.

It is quite common for a résumé to contain only one skill area.

Formatting

The text of skill area's `title` element is displayed as a top-level heading in all output formats. Then all contained skill sets are displayed sequentially. See `skillset` for information about skill set formatting.

Parents

`skillarea` may be contained in these elements: `resume`, `skillareas`

Attributes

`id` A string used to uniquely identify a skill area. The skill area may then be referred to by this identifier.

Examples

Example 8. Broad skill area

```
<skillarea>
  <title>Skills</title>

  <skillset>
    <title>Computers</title>
    <skill>Typing (<emphasis>150 WPM!!</emphasis>)</skill>
    <skill>Dropping little food crumbs in keyboards</skill>
  </skillset>

  <skillset>
    <title>Writing</title>
    <skill>Editing</skill>
    <skill>Cursive</skill>
  </skillset>
</skillarea>
```

Example 9. More focused skill area

```
<skillarea>
  <title>Financial Skills</title>

  <skillset>
    <title>Investing</title>
    <skill>Stocks</skill>
    <skill>Bonds</skill>
    <skill>Money market accounts</skill>
    <skill>Retirement accounts</skill>
  </skillset>

  <skillset>
    <title>Bookkeeping</title>
    <skill>Amortization</skill>
    <skill>Loan schedules</skill>
    <skill>Book balancing</skill>
  </skillset>

  <skillset>
    <title>Advising</title>
    <skill>Retirement investment</skill>
    <skill>College investment</skill>
    <skill>Philanthropy</skill>
  </skillset>
</skillarea>
```

Name

skillareas -- Group of one or more skill areas

skillareas

Synopsis

Content Model

```
(skillarea+)
```

Attributes

Name	Type	Default
targets	CDATA	*None*

Description

Important

This element is deprecated as of XML Résumé Library version 1.4.0. It should not be used.

`skillarea` elements are now allowed as direct children of `resume`, so `skillareas` is no longer necessary. `skillareas` is still allowed in this release of XML Résumé Library, but may be removed in future versions. It is recommended that you remove all `skillareas` elements from XML résumés.

The `skillareas` element contains one or more `skillarea`s.

This element is a top-level section of a résumé.

Parents

`skillareas` may be contained in these elements: `resume`

Examples

```
<skillareas>
  <skillarea>
    <title>Aviation</title>
    <skillset>
      <title>Piloting</title>
      <skill>Biplanes</skill>
      <skill>Jumbo Jets</skill>
      <skill>Space Shuttle</skill>
    </skillset>
    <skillset>
      <title>Navigation</title>
      <skill>Map and charts</skill>
      <skill>Instruments</skill>
      <skill>Stars</skill>
    </skillset>
  </skillarea>

  <skillarea>
    <title>Cleaning</title>
    <skillset>
      <title>Tools</title>
      <skill>Rag</skill>
      <skill>Sponge</skill>
      <skill>Towel</skill>
    </skillset>
    <skillset>
      <title>Chemicals</title>
      <skill>Bleach</skill>
      <skill>Water</skill>
      <skill>Soap</skill>
    </skillset>
  </skillarea>
</skillareas>
```

Name

skills -- Group of one or more skills

skills

Synopsis

Content Model

```
(skill+)
```

Attributes

Name	Type	Default
id	ID	*None*
targets	CDATA	*None*

Description

Important

This element is deprecated as of XML Résumé Library version 1.4.0. It should not be used.

`skill` elements are now allowed as direct children of `skillset`, so `skills` is no longer necessary. `skills` is still allowed in this release of XML Résumé Library, but may be removed in future versions. It is recommended that you remove all `skills` elements from XML résumés.

The `skills` element contains one or more (usually related) skills.

Parents

`skills` may be contained in these elements: `skillset`

Attributes

`id` A string used to uniquely identify a skills group. The group may then be referred to by this identifier.

See Also

`skills.format`.

Examples

```
<skills>
  <skill>Leaping over tall buildings in a single bound</skill>
  <skill>Flying</skill>
  <skill>Quickly changing clothes</skill>
</skill>
```

Name

skillset -- A titled group of one or more related skills

skillset

Synopsis

Content Model

`(title?,(skill+|skills))`

Attributes

Name	Type	Default
id	ID	*None*
targets	CDATA	*None*

Description

The `skillset` element contains a titled set of (usually closely) related skills.

Formatting

The *`skills.format`* parameter controls the formatting of skill sets.

Parents

`skillset` may be contained in these elements: `skillarea`

Attributes

`id` A string used to uniquely identify a skill set. The skill set may then be referred to by this identifier.

Examples

```
<skillset>
  <title>Eating Utinsels</title>
  <skill>Knife</skill>
  <skill>Fork</skill>
  <skill>Spoon</skill>
  <skill>Soup Spoon</skill>
</skillset>
```

Name

state -- A name or abbreviation of a state

state

Synopsis

Content Model

```
(#PCDATA)
```

Attributes

Name	Type	Default
id	ID	*None*
targets	CDATA	*None*

Description

The `state` element contains the name or abbreviation for the name of a state or other similar political entity.

The `state` element represents a division of a country. It should not be confused with the `country` element, which represents a nation-state.

Parents

`state` may be contained in these elements: `address`, `location`

Attributes

`id` A string used to uniquely identify a state name. The name may then be referred to by this identifier.

See Also

`province`, `county`, `prefecture`.

Examples

Example 10. State name

```
<state>New York</state>
```

Example 11. State abbreviation

```
<state>NY</state>
```

Name

street -- A street name, number, and other related information

street

Synopsis

Content Model

```
(#PCDATA)
```

Attributes

Name	Type	Default
id	ID	*None*
targets	CDATA	*None*

Description

The `street` element contains a street name and number, a suite or apartment number, or other similar information.

Multiple `street` elements are allowed. It is preferred to use multiple `street` elements instead of the deprecated `street2` element.

Parents

`street` may be contained in these elements: `address`

Attributes

`id` A string used to uniquely identify a street. The street may then be referred to by this identifier.

Examples

For examples, see `address`.

Name

street2 -- A second line of a street address

street2

Synopsis

Content Model

```
(#PCDATA)
```

Attributes

Name	**Type**	**Default**
id	ID	*None*
targets	CDATA	*None*

Description

Important

This element is deprecated as of XML Résumé Library version 1.3.3. It should not be used.

Because multiple `street` elements are now allowed, `street2` is unneeded. `street2` is still supported in this release of the XML Résumé Library, but may be removed in future versions. It is recommended that all `street2` elements be converted to `street` elements.

The `street2` element is similar in semantics to the `street` element. It often contains more specialized routing information than `street`, such as a suite or apartment number.

Parents

`street2` may be contained in these elements: `address`

Attributes

`id` A string used to uniquely identify the second line of a street address. The line may then be referred to by this identifier.

See Also

`street`.

Examples

None.

Name

subject -- A name of a class or topic of study

subject

Synopsis

Content Model

```
(title,result)
```

Attributes

Name	**Type**	**Default**
targets	CDATA	*None*

Description

The `subject` element contains the `title` of a class or other course of study, as well as its `result`.

A subject's title can be as general as “mathematics” (a general field of study), or as specific as “ECEn 224 Electrostatics Fall 1994” (a specific university class, along with the date the class was taken).

Parents

`subject` may be contained in these elements: `subjects`

Examples

```
<subject>
  <title>English</title>
  <result>A-</result>
</subject>

<subject>
  <title>PE 194 Crabwalking</title>
  <result>F</result>
</subject>
```

Name

subjects -- A group of one or more subjects

subjects

Synopsis

Content Model

```
(subject+)
```

Attributes

Name	Type	Default
targets	CDATA	*None*

Description

The `subjects` element contains one or more `subjects` that were studied as part of the `degree` represented by the parent of the `subjects`.

Parents

`subjects` may be contained in these elements: `degree`

Examples

```
<subjects>
  <subject>
    <title>Chemistry</title>
    <result>C</result>
  </subject>
  <subject>
    <title>Physics</title>
    <result>B-</result>
  </subject>
</subjects>
```

Name

suburb -- A name of a suburb

suburb

Synopsis

Content Model

```
(#PCDATA)
```

Attributes

Name	**Type**	**Default**
id	ID	*None*
targets	CDATA	*None*

Description

The `suburb` element contains the name of a suburb, as used by a postal service to deliver mail.

Parents

`suburb` may be contained in these elements: `address`

Attributes

`id` A string used to uniquely identify a suburb. The suburb may then be referred to by this identifier.

See Also

`city`, `prefecture`, `ward`.
-->

Examples

Example 12. A New Zealand Address

```
<address>
  <street>236 Majoribanks Street</street>
  <suburb>Mount Victoria</suburb>
  <city>Wellington</city>
  <postalCode>6001</postalCode>
  <country>New Zealand</country>
</address>
```

Name

suffix -- A suffix of a name, usually specifying lineage

suffix

Synopsis

Content Model

```
(#PCDATA)
```

Attributes

Name	**Type**	**Default**
targets	CDATA	*None*

Description

The `suffix` element contains a suffix or suffix abbreviation that follows a name, such as “Jr.”, “Senior”, or “III”.

Parents

`suffix` may be contained in these elements: `name`

Examples

```
<name>
  <firstname>Leonard</firstname>
  <middlenames>Scott Quink</middlenames>
  <surname>Waggenblast</surname>
  <suffix>IV</suffix>
</name>

<name>
  <firstname>Dolores</firstname>
  <surname>Sanchez</surname>
  <suffix>Jr.</suffix>
</name>
```

Name

surname -- A family name

surname

Synopsis

Content Model

```
(#PCDATA)
```

Attributes

Name	**Type**	**Default**
targets	CDATA	*None*

Description

The `surname` element contains a person's family, or last name.

Parents

`surname` may be contained in these elements: `name`

Examples

For examples, see `name`.

Name

tail --

tail

Synopsis

Content Model

```
(#PCDATA)
```

Attributes

Name	**Type**	**Default**
targets	CDATA	*None*

Description

Important

This element is deprecated as of XML Résumé Library version 1.4.0. It should not be used, and will be removed in a future version.

Parents

`tail` may be contained in these elements: `docpath`

Examples

```
<!-- TODO -->
```

Name

title -- A title or heading

title

Synopsis

Content Model

```
(#PCDATA)
```

Attributes

Name	Type	Default
targets	CDATA	*None*

Description

The `title` element contains a title or heading. Its semantics vary depending on the element that contains it:

`name`
When contained in this element, `title` contains a person's formal title, such as “Dr.”, “Ms.”, or “Professor”.

`interest`
When contained in this element, `title` contains the name of an interest, such as “Cooking”, or “Politics”.

`membership`, `referee`
When contained in these elements, `title` contains the name of the person's role or capacity withing an organization. Examples: “Treasurer”, “Member”, “Chief Officer of Toothpaste Policy”, “Lead Programmer”.

`awards`, `interests`, `memberships`, `skillarea`
When contained in these elements, `title` contains a heading for that section of a résumé. It is formatted as a top-level heading.

Note

The `title` element is optional in `awards` and `interests`. If it is not present, the *`awards.word`* and *`interests.word`* parameters will be used as the heading text in formatted résumés, respectively.

`skillset`
When contained in this element, `title` contains the name of a skill category. Examples: “Teaching”, “Computers”, “Woodworking”, “Soap Carving”.

`subject`
When contained in this element, `title` contains the name of a subject. It can be as general as “English” (a very broad field of study), or as specific as “Eng 115 Intro to Writing Fall 2001” (a specific university course name, number, and date of study).

Parents

`title` may be contained in these elements: `award`, `awards`, `clearances`, `interest`, `interests`, `membership`, `memberships`, `name`, `referee`, `skillarea`, `skillset`, `subject`

See Also

`artTitle`, `bookTitle`.

Examples

Example 13. `interest` title

```
<interest>
  <title>Fly fishing for sharks</title>
  <description><para>
    I once caught a 300-kg. Great White off the coase of Florida using a green
    and gold-colored dragonfly that I tied.
  </para></description>
</interest>
```

Example 14. `membership` and `memberships` titles

```
<memberships>
  <title>Professional Memberships</title>
  <membership>
    <organization>American Society of Blower-Uppers</organization>
  </membership>
  <membership>
    <title>Spokesperson</title>
    <organization>Smallville Demolitioneers</organization>
  </membership>
  <membership>
    <title>President</title>
    <organization>Willow County Pyrotechnicans' Guild</organization>
  </membership>
</membership>
```

Example 15. `skillarea` and `skillset` titles

```
<skillarea>
  <title>Computer Skills</title>
  <skillset>
    <title>Word Processing</title>
    <skill>Microsoft Word</skill>
    <skill>Corel WordPerfect</skill>
  </skillset>
  <skillset>
    <title>Programming</title>
    <skill>Perl</skill>
    <skill>Python</skill>
    <skill>XML</skill>
    <skill>C++</skill>
  </skillset>
</skillarea>
```

Example 16. `subject` title

```
<subject>
  <title>Intro to Sociology</title>
  <result>A-</result>
</subject>
```

Name

to -- The ending point in a period of time

to

Synopsis

Content Model

```
(date|present)
```

Attributes

Name	**Type**	**Default**
targets	CDATA	*None*

Description

The `to` element specifies the instant at which a given period of time ends. It is always preceeded by a `from` element in a `period`.

Parents

`to` may be contained in these elements: `period`

See Also

`from`.

Examples

For examples, see `from` and `period`.

Name

uri -- A Uniform Resource Indicator

uri

Synopsis

Content Model

```
(#PCDATA)
```

Attributes

Name	Type	Default
targets	CDATA	*None*

Description

Important

This element is deprecated as of XML Résumé Library version 1.4.0. It should not be used, and will be removed in a future version.

Parents

`uri` may be contained in these elements: `head`, `node`

Examples

```
<!-- TODO -->
```

Name

url -- A Uniform Resource Locator

url

Synopsis

Content Model

```
(#PCDATA)
```

Attributes

Name	Type	Default
targets	CDATA	*None*

Description

Important

The use of the `url` element as a child of `publisher` is deprecated as of XML Résumé Library version 1.4.0.

We recommend that you convert all `url`s contained in `publisher`s to `link`s.

The `url` element contains a single Uniform Resource Locator, as specified by RFC 1738 [http://www.w3.org/Addressing/rfc1738.txt] and RFC 1808 [http://www.w3.org/Addressing/rfc1808.txt].

Formatting

There are two contexts in which a URL is formatted. The first is as an inline (e.g. when contained in a `para`). The second is as contact element (contained in a `contact`).

HTML When an inline, displayed as a hyperlink in a fixed-width font (enclosed in `code` tags). The “hot” text is the value of the element.

When a contact element, displayed the same as when an inline, except not enclosed in `code` tags.

FO/PDF When an inline, displayed as non-hyperlinked text, using the font specified by the *`url.font.family`* parameter. (Default: `monospace`

When a contact element, displayed as non-hyperlinked text, using the normal font.

Plain text Formatted as the value of the element.

Parents

`url` may be contained in these elements: `achievement`, `contact`, `employer`, `institution`, `organization`, `para`, `project`, `pub`, `publisher`, `skill`

See Also

`link`, `uri`.

Examples

```
<para>
  The Apache XML Project web page is located at
  <url>http://xml.apache.org/</url>.
</para>

<skill>
  Python (see <url>http://www.python.org/</url>)
</skill>
```

Name

ward -- A name of a division of a city, town, or county

ward

Synopsis

Content Model

```
(#PCDATA)
```

Attributes

Name	Type	Default
id	ID	*None*
targets	CDATA	*None*

Description

The `ward` element contains the name of a ward, as needed to deliver mail to an address. A ward is an administrative division a city, and of some English, Scottish, and American counties.

Parents

`ward` may be contained in these elements: `address`

Attributes

`id` A string used to uniquely identify a ward. The ward may then be referred to by this identifier.

See Also

`county`, `prefecture`, `suburb`.

Examples

```
<address>
  <street>Rua Afonso Camargo, 805</street>
  <ward>Santana</ward>
  <city>Guarapuava</city>
  <state>PR</state>
  <postalCode>85070-200</postalCode>
  <country>Brazil</country>
</address>
```

Name

year -- A year

year

Synopsis

Content Model

```
(#PCDATA)
```

Attributes

Name	Type	Default
targets	CDATA	*None*

Description

The `year` element contains a year number, along with a possible string that identifies the date system, such as “B.C.” or “A.D.”.

Parents

`year` may be contained in these elements: `copyright`, `date`, `pubDate`

Examples

```
<year>1995</year>
```

```
<year>240 A.D.</year>
```

Name

zip -- A zip code

zip

Synopsis

Content Model

```
(#PCDATA)
```

Attributes

Name	Type	Default
id	ID	*None*
targets	CDATA	*None*

Description

The `zip` element contains a single zip code (United States postal code). It may contain any format, such as standard five-digit (“34525”), ZIP+4 (“34525-1625”), or any other valid format.

For non-U.S. postal codes, use `postalCode`.

Parents

`zip` may be contained in these elements: `address`

Attributes

`id` A string used to uniquely identify a zip code. The zip code may then be referred to by this identifier.

See Also

`postalCode`.

Examples

For examples, see `address`.

XSL Parameter Reference

Parameters are settings that affect all files that are generated by the XSL templates. They are analogous to attributes that affect everything instead of just one element.

This section is a partial listing of user-configurable parameters. There are many more tunable parameters in `params.xsl`. Take a look at the file in a text editor if you're interested.

Name

address.format -- controls default formatting of the address element

address.format

Applies To

This parameter controls the formatting of the address element in all output formats.

Possible Values

Possible values include standard (the default), european, and italian.

Note

In the examples below, <street> means the value of XML element <street>, if defined. <suburb | ward> means the value of XML element <suburb>, if defined, otherwise the value of XML element <ward>, otherwise blank.

standard

Formats addresses in North American layout:

```
<street>
<street2>
<suburb | ward>
<city>, <state | province | county > <zip | postalCode>
<country>
```

european

Formats addresses in European layout:

```
<street>
<street2>
<suburb | ward>
<zip | postalCode> <city>, <state | province | county > <country>
```

italian

Formats addresses in Italian layout:

```
<street>
<street2>
<postalCode> <city> (<province>)
<country>
```

If none of these formats suit your needs, there is the option of entering your address as a single text block, in which case it will be formatted with line breaks intact. For example:

```
<address>Sr. H&eacute;ctor Garc&iacute;a Mariz&oacute;
Reina #35, apt. 4a, e/ Gervasio y Escobar
Ciudad de La Habana, CP 11900
CUBA</address>
```

Name

css.href -- indicates the CSS stylesheet to use to format HTML output

css.href

Applies To

This parameter applies to only the HTML output format.

Possible Values

The value of *`css.href`* is a URL that points to a Cascading Style Sheet file. The HTML version of the résumé will link to this file.

For information on the CSS classes that are available for formatting, see Chapter 3.

Name

header.format -- controls formatting of the `header` element

header.format

Applies To

This parameter controls the formatting of the `header` element in HTML and text output formats. The FO/PDF formatter is not affected by this parameter; it produces output similar to the `centered` option, but with the text left-justified within a centered block.

Possible Values

Possible values include `standard` (the default) and `centered`.

`standard`

Produces a centered line with the name and the word “Résumé”, followed by a left-justified contact information block:

```
Jo Doe - Résumé

Contact Information:
     Jo Doe
     123 Elm #456
     Garbonzoville, NX 99999-9999
     Phone: 555.555.5555
     Email: doe@doe.doe
     URL: http://doe.com/~doe/
```

This value is the default for *`header.format`*.

`centered`

Produces a single centered block with the name and contact information.

```
Jo Doe
     123 Elm #456
     Garbonzoville, NX 99999-9999
     Phone: 555.555.5555
     Email: doe@doe.doe
     URL: http://doe.com/~doe/
```

Name

interest.description.format -- controls formatting of interest descriptions

interest.description.format

Applies To

This parameter controls the formatting of the `description` element when it is contained in an `interest`. It affects all output formats.

Possible Values

Possible values include `single-line` and `block`. `single-line` is the default.

The examples below demonstrate the formatting of this XML fragment:

```
<interests>
  <interest>
    <title>Scuba diving</title>
  </interest>
  <interest>
    <title>Flying</title>
    <description>
      <para>I have my pilot's license, and have logged over 1000 in-flight
      hours.</para>
      <para>I have also constructed my own airplane from a kit.</para>
    </description>
  </interest>
</interests>
```

`single-line`

This value is the default. Formats all of the `para` elements in the description on the same logical line as the interest title. The title is separated from the description by a period, and the description paragraphs are separated from each other by *`description.para.separator.text`*, which defaults to an em-dash (“—”).

For example, the XML above would be formatted similar to the following:

Interests

- Scuba diving
- Flying. I have my pilot's license, and have logged over 1000 in-flight hours. — I have also constructed my own airplane from a kit.

`block`

Formats each of the `para` elements in the description as a separate block below the interest title.

For example, the XML above would be formatted similar to the following:

Interests

- Scuba diving
- Flying

  I have my pilot's license, and have logged over 1000 in-flight hours.

  I have also constructed my own airplane from a kit.

Name

referees.display -- Determines whether or not referees are visible in formatted output

referees.display

Applies To

This parameter controls the formatting of the `referee` element in all output formats.

Possible Values

Possible values include `1` (display referees) and `0` (display alternate text).

`1` (true)
Referees are displayed in formatted output.

`0` (false)
The text of the *`referees.hidden.phrase`* parameter (default: `Available upon request.`) is displayed in place of referees.

Name

skills.format -- controls formatting of the `skillset` element

skills.format

Applies To

This parameter controls the formatting of the `skillset` element in all output formats.

Possible Values

Possible values include `bullet` (the default) and `comma`.

The examples below show how a `<skillset>` like this would be rendered:

```
<skillarea>
  <title>Technical Skills</title>
  <skillset>
    <title>Programming Languages</title>
    <skill>Java</skill>
    <skill>C++</skill>
    <skill>C</skill>
    <skill>perl</skill>
  </skillset>
</skillarea>
```

`bullet` Formats skills as a bulleted list, one skill per line:

Technical Skills

Programming Languages

- Java
- C++
- C
- Perl

`comma` Produces a comma-separated list on a single line:

Technical Skills

Programming Languages: Java, C++, C, Perl

Name

skills.level.display -- Determines whether or not skill level attributes are visible in formatted output

skills.level.display

Applies To

This parameter controls the formatting of the `level` attribute of `skill` elements in all output formats.

Possible Values

Possible values include `1` (display the attribute) or `0` (suppress the attribute).

`1` (true)
: `level` attributes of `skill`s are displayed after all of the rest of the content of the skill. The level is displayed surrounded by *`skills.level.start`* and *`skills.level.end`* (parenthesis, by default).

`0` (false)
: `level` attributes contained in `skill`s are suppressed; they are treated as if the didn't exist for formatting purposes.

Name

subjects.format -- controls formatting of the `subjects` element

subjects.format

Applies To

This parameter controls the formatting of the `subjects` element in all output formats.

Possible Values

Possible values include `comma` (the default) and `table`.

The examples below show how a `<subjects>` section like this would be rendered:

```
<subjects>
  <subject>
    <title>English</title>
    <result>C</result>
  </subject>
  <subject>
    <title>Science</title>
    <result>A</result>
  </subject>
  <subject>
    <title>Math</title>
    <result>B-</result>
  </subject>
</subjects>
```

`comma` Produces a comma-separated list on a single line:

Subjects: English (C), Science (A), Math (B-).

`table` Produces a table, with subject name in the first column and result in the second column:

Subjects

English	C
Science	A
Math	B-

Chapter 3. CSS Class Reference

Table of Contents

This section of the manual describes the Cascading Stylesheets classes that are available for formatting HTML output. For more information on CSS, visit the W3C web site [http://www.w3c.org/Style/CSS/].

Overview

Table 3.1. Table of CSS Classes

Class	Where Used
`achievement`	On the `li` for a single achievement.
`address`	On the `p` that contains a postal address.
`award`	On the `li` for a single award.
`awardTitle`	On the `span` of an award title.
`bookTitle`	On the `cite` used to refer to a title of a book in one of your publications.
`citation`	On all HTML `cite`s generated from all XML `citation`s.
`copyright`	On the `address` for copyright information.
`degree`	On the `li` for a single degree.
`degrees`	On the `ul` for the list of degrees you hold.
`degreeTitle`	On the `span` surrounding a degree level and major, such as “B.S. in Political Science”.
`description`	Around descriptions of items.
`emphasis`	On all HTML `strong`s generated from all XML `emphasis`s.
`employer`	Around all employer names.
`gpaPreamble`	On the `span` around the preamble to a GPA, such as “Overall GPA”.
`heading`	On the `h2` of résumé section headings, such as “Professional Objective” or “Employment History”.
`headingText`	On the `span` that contains heading text. This element is contained directly within the `h2` with the `heading` class, and allows style to be applied to just the heading text, instead of the whole heading line.
`jobTitle`	On the `span` enclosing the title you held at job.
`lastModified`	On the `p` for the “last modified” notice.
`level`	On the `acronym` for the level attained in a degree.
`linkA`	On the `a` of a hyperlink generated from a `link` element in a résumé.

Class	Where Used
`membershipTitle`	On the position title in a membership.
`nameHeading`	Around your name at the top of the résumé.
`note`	On the `span` of a note.
`organization`	On the name of the organization of which you're a member.
`para`	On all HTML `p`s generated from all XML `para`s.
`project`	On the `li` for a single project.
`pub`	On a single publication.
`pubs`	On the `ul` for the list of publications.
`referee`	On the `div` of a referee.
`refereeName`	On the `div` of a referee name.
`refereeContact`	On the `div` of a referee contact.
`resume`	On the `body` of the entire HTML output.
`skill`	On the `li` for a single skill.
`skills`	If *`skills.format`* is `bullet`, on the `ul` for a list of skills. If *`skills.format`* is `comma`, on the `span` for a list of skills.
`skillsetTitle`	On the `h3` for a title of a set of skills.
`urlA`	On the `a` of a hyperlink generated from a `url` element in a résumé.

Glossary

attribute

Attributes provide additional information about element that they appear in. They take the form of name-value pairs in the element. The **emphasized** parts of this example are attributes:

```
<author name="a.kernighan">
<resume id="r25a4">
```

element

An element of an XML document that defines an “object”, such as `<resume>` or `<firstname>`. HTML tags are examples of elements. Additionally, elements can have attributes. Elements have a start (or "open") tag (e.g., `<firstname>`) and an end (or "close") tag (e.g., `</firstname>`).

Each element (except for the root `<resume>`) has exactly one parent element and 0 or more child elements.

parent element

The element whose start tags and end tags surround a given element.

child element

Any element positioned between the start tags and end tags of a given element.

Part III. Appendixes

Appendix A. GNU Free Documentation License

Table of Contents

Version 1.1, March 2000

> Copyright (C) 2000 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

PREAMBLE

The purpose of this License is to make a manual, textbook, or other written document “free” in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The “Document”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “you”.

A “Modified Version” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “Secondary Section” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (For example, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “Invariant Sections” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License.

The “Cover Texts” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License.

A “Transparent” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup has been designed to thwart or discourage subsequent modification by readers is not Transparent. A copy that is not “Transparent” is called “Opaque”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML designed for human modification. Opaque formats include PostScript, PDF, proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML produced by some word processors for output purposes only.

The “Title Page” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

COPYING IN QUANTITY

If you publish printed copies of the Document numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a publicly-accessible computer-network location containing a complete Transparent copy of the Document, free of added material, which the general network-using public has access to download anonymously at no charge using public-standard network protocols. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has less than five).

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section entitled “History”, and its title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section entitled “History” in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the “History” section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. In any section entitled “Acknowledgements” or “Dedications”, preserve the section's title, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section entitled “Endorsements”. Such a section may not be included in the Modified Version.

N. Do not retitle any existing section as “Endorsements” or to conflict in title with any Invariant Section.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections entitled “History” in the various original documents, forming one section entitled “History”; likewise combine any sections entitled “Acknowledgements”, and any sections entitled “Dedications”. You must delete all sections entitled “Endorsements.”

COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an “aggregate”, and this License does not apply to the other self-contained works thus compiled with the Document, on account of their being thus compiled, if they are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one quarter of the entire aggregate, the Document's Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around the whole aggregate.

TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

> Copyright (c) YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have no Invariant Sections, write "with no Invariant Sections" instead of saying which ones are invariant. If you have no Front-Cover Texts, write “no Front-Cover Texts” instead of “Front-Cover Texts being LIST”; likewise for Back-Cover Texts.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.