Outline Processor Markup Language

Spec

Mail List

Directory

Editor

RSS

XML-RPC

XML

Dave

OPML 2.0

11/22/07 DW -- This is a published document, please deploy, the OPML 2.0 format is frozen.

Goals of the OPML format

This document describes a format for storing outlines in XML 1.0 called *Outline Processor Markup Language* or OPML.

The purpose of this format is to provide a way to exchange information between outliners and Internet services that can be browsed or controlled through an outliner.

OPML is also the file format for an outliner application, which is why OPML files may contain information about the size, position and expansion state of the window the outline is displayed in.

OPML has also become popular as a format for exchanging subscription lists between feed readers and aggregators.

The design goal is to have a transparently simple, self-documenting, extensible and human readable format that's capable of representing a wide variety of data that's easily browsed and edited. It should be possible for a reasonably technical person to understand the format with a quick read of a few web pages.

It's an open format, meaning that other outliner and service developers are welcome to use the format to be compatible with Radio UserLand, the OPML Editor, or for any other purpose.

What is an outline?

An outline is a tree, where each node contains a set of named attributes with string values.

What is an <opml>?

<opml> is an XML element, with a single required attribute, version; a <head> element and a <body> element, both of which are required.

The version attribute is a version string, of the form, x.y, where x and y are both numeric strings.

The value of version may be 1.0, if it conforms to the eariler version of this specification, published in 2000; or 2.0 if it conforms to this specification.

If you see a version 1.1 file, treat it as if it were a version 1.0 file.

What is a <head>?

A <head> contains zero or more optional elements, described below.

<title> is the title of the document.

<dateCreated> is a date-time, indicating when the document was created.

<dateModified> is a date-time, indicating when the document was last modified.

<ownerName> is a string, the owner of the document.

<ownerEmail> is a string, the email address of the owner of the document.

<ownerId> is the http address of a web page that contains information that allows a human reader to communicate with the author of the document via email or other means. It also may be used to identify the author. No two authors have the same ownerId.

<docs> is the http address of documentation for the format used in the OPML file. It's probably a pointer to this page for people who might stumble across the file on a web server 25 years from now and wonder what it is.

<expansionState> is a comma-separated list of line numbers that are expanded. The line numbers in the list tell you which headlines to expand. The order is important. For each element in the list, X, starting at the first summit, navigate flatdown X times and expand. Repeat for each element in the list.

<vertScrollState> is a number, saying which line of the outline is displayed on the top line of the window. This number is calculated with the expansion state already applied.

<windowTop> is a number, the pixel location of the top edge of the window.

<windowLeft> is a number, the pixel location of the left edge of the window.

<windowBottom> is a number, the pixel location of the bottom edge of the window.

<windowRight> is a number, the pixel location of the right edge of the window.

What is a <body>?

A <body> contains one or more <outline> elements.

What is an <outline>?

An <outline> is an XML element containing at least one required attribute, text, and zero or more additional attributes. An <outline> may contain zero or more <outline> sub-elements. No attribute may be repeated within the same <outline> element.

Text attribute

Every outline element must have at least a *text* attribute, which is what is displayed when an outliner opens the OPML file. To omit the text attribute would render the outline useless in an outliner. This is what the user would see -- clearly an unacceptable user experience. Part of the purpose of producing OPML is to give users the power to accumulate and organize related information in an outliner. This is as important a use for OPML as data interchange.

A missing text attribute in any outline element is an error.

Text attributes may contain encoded HTML markup.

Other special attributes

type is a string, it says how the other attributes of the <outline> are interpreted.

isComment is a string, either "true" or "false", indicating whether the outline is commented

or not. By convention if an outline is commented, all subordinate outlines are considered to also be commented. If it's not present, the value is false.

isBreakpoint is a string, either "true" or "false", indicating whether a breakpoint is set on this outline. This attribute is mainly necessary for outlines used to edit scripts. If it's not present, the value is false.

created is the date-time that the outline node was created.

category is a string of comma-separated slash-delimited category strings, in the format defined by the RSS 2.0 category element. To represent a "tag," the category string should contain no slashes. Examples: 1. category="/Boston/Weather". 2. category="/Harvard/Berkman,/Politics".

Subscription lists

A subscription list is a possibly multiple-level list of subscriptions to feeds. Each sub-element of the body of the OPML document is a node of type *rss* or an outline element that contains nodes of type *rss*.

Today, most subscription lists are a flat sequence of *rss* nodes, but some aggregators allow categorized subscription lists that are arbitrarily structured. A validator may flag these files, warning that some processors may not understand and preserve the structure.

Required attributes: type, text, xmlUrl. For outline elements whose type is *rss*, the *text* attribute should initially be the top-level title element in the feed being pointed to, however since it is user-editable, processors should not depend on it always containing the title of the feed. *xmlUrl* is the http address of the feed.

Optional attributes: description, htmlUrl, language, title, version. These attributes are useful when presenting a list of subscriptions to a user, except for version, they are all derived from information in the feed itself.

description is the top-level description element from the feed. *htmlUrl* is the top-level link element. *language* is the value of the top-level language element. *title* is probably the same as *text*, it should not be omitted. *title* contains the top-level title element from the feed.

version varies depending on the version of RSS that's being supplied. It was invented at a time when we thought there might be some processors that only handled certain versions, but that hasn't turned out to be a major issue. The values it can have are: RSS1 for RSS 1.0; RSS for 0.91, 0.92 or 2.0; scriptingNews for scriptingNews format. There are no known values for Atom feeds, but they certainly could be provided.

Inclusion

An outline element whose type is *link* must have a *url* attribute whose value is an http address. The *text* element is, as usual, what's displayed in the outliner; it's also what is displayed in an HTML rendering.

When a *link* element is expanded in an outliner, if the address ends with ".opml", the outline expands in place. This is called inclusion.

If the address does not end with ".opml" the *link* is assumed to point to something that can be displayed in a web browser.

In OPML 2.0 a new type is introduced. An outline element whose type is *include* must have a *url* attribute that points to the OPML file to be included. The *text* attribute is, as usual, what's displayed in the outliner, and it's also what is displayed in an HTML rendering.

The difference between link and include is that link may point to something that is displayed in a web browser, and include always points to an OPML file.

Directories

A directory may contain an arbitrary structure of outline elements with type *include, link* or *rss,* and possibly other types. A wide variety of software can be used to display directories, including outliners such as the OPML Editor.

Extending OPML

An OPML file may contain elements and attributes not described on this page, only if those elements are defined in a namespace, as specified by the W3C.

OPML can also be extended by the addition of new values for the type attribute. When specifying such an extension, following the example of this specification, say which attributes are required and which are optional, and explain the roles each of the attributes plays, how they relate to each other, and what rules they must conform to. There is a mechanism in the OPML Editor that is based on this form of extension.

Developers should, whenever possible, use capabilities that are already in use by others, or included in this spec, or recommendations or guidelines.

Examples

http://hosting.opml.org/dave/spec/subscriptionList.opml

http://hosting.opml.org/dave/spec/states.opml

http://hosting.opml.org/dave/spec/simpleScript.opml

http://hosting.opml.org/dave/spec/placesLived.opml

http://hosting.opml.org/dave/spec/directory.opml

http://hosting.opml.org/dave/spec/category.opml

Notes

1. All date-times conform to the Date and Time Specification of RFC 822, with the exception that the year may be expressed with two characters or four characters (four preferred).

2. The page in <ownerId> may contain link elements pointing to other documents containing information about the owner. For example, you may have a link element pointing to a FOAF document describing the owner and his or her network of friends; or an RSS feed with news from the owner, possibly even related via the RSS 2.0 category element to parts of the OPML document. In other words, all the extension mechanisms of HTML can come into play.

3. The value of type attributes are not case-sensitive, that is type="LINK" has the same meaning as type="link".

4. Outline attributes generally do not contain encoded HTML markup, unless their are specifically said to include markup.

5. Processors should ignore any attributes they do not understand.

6. There are no documented limits to the number of attributes an <outline> element can have,

or the number of <outline> elements it can contain or the size of any attribute.

7. Each sub-element of <head> may appear once or not at all. No sub-element of <head> may be repeated.

8. If an HTML page is generated using an OPML document, you may use an HTML link element to provide for "auto-discovery" of the OPML. The rel attribute value is "outline", the type "text/x-opml", and of course the href attribute contains the address of the OPML document.

9. You may include elements of OPML 2.0 in other XML-based formats. The URI for the namespace is http://opml.org/spec2. The namespace declaration should look like this: xmlns:opml="http://opml.org/spec2". However, for backward compatibility, the core elements (those defined by this spec) in an OPML 2.0 document are *not* in a namespace. Here's an example of an RSS 2.0 file that contains an outline in one of its items.

Roadmap

Version 2.0 is the last version of OPML. Any further development will take place in namespaces, new outline types, per the Extending OPML section of this specification; or in formats derived from OPML with different names.

Copyright and disclaimer

UserLand Software, Inc. and Scripting News, Inc. are refererred to in the following as "the Companies."

Posted: 2/16/06; 3:53:39 PM.


This is a
Manila
Site