Notice

This is a draft document for public review.

Overview

This document defines the OpenSearch description document, the OpenSearch Query element, the OpenSearch URL template syntax, and the OpenSearch response elements. Collectively these formats may be referred to as "OpenSearch 1.1" or simply "OpenSearch".

Search clients can use OpenSearch description documents to learn about the public interface of a search engine. These description documents contain parameterized URL templates that indicate how the search client should make search requests. Search engines can use the OpenSearch response elements to add search metadata to results in a variety of content formats.

Namespace

The XML Namespaces URI for the XML data formats described in this specification is:

```
http://a9.com/-/spec/opensearch/1.1/
```

OpenSearch description document

Overview

An OpenSearch description document can be used to describe the web interface of a search engine.

Examples

Example of a simple OpenSearch description document:

```
<?xml version="1.0" encoding="UTF-8"?>
<OpenSearchDescription xmlns="http://a9.com/-/spec/opensearch/1.1/">
  <ShortName>Web Search</ShortName>
  <Description>Use Example.com to search the Web.</Description>
  <Tags>example web</Tags>
  <Contact>admin@example.com</Contact>
  <Url type="application/rss+xml"
       template="http://example.com/?q={searchTerms}&amp;pw={startPage?}&amp;format=rss"/>
</OpenSearchDescription>
```

Example of a detailed OpenSearch description document:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<OpenSearchDescription xmlns="http://a9.com/-/spec/opensearch/1.1/">
  <ShortName>Web Search</ShortName>
  <Description>Use Example.com to search the Web.</Description>
  <Tags>example web</Tags>
  <Contact>admin@example.com</Contact>
  <Url type="application/atom+xml"
       template="http://example.com/?q={searchTerms}&amp;pw={startPage?}&amp;format=atom"/>
  <Url type="application/rss+xml"
       template="http://example.com/?q={searchTerms}&amp;pw={startPage?}&amp;format=rss"/>
  <Url type="text/html"
       template="http://example.com/?q={searchTerms}&amp;pw={startPage?}"/>
  <LongName>Example.com Web Search</LongName>
  <Image height="64" width="64" type="image/png">http://example.com/websearch.png</Image>
  <Image height="16" width="16" type="image/vnd.microsoft.icon">http://example.com/websearch.ico</Imag
  <Query role="example" searchTerms="cat" />
  <Developer>Example.com Development Team</Developer>
  <Attribution>
    Search data Copyright 2005, Example.com, Inc., All Rights Reserved
  </Attribution>
  <SyndicationRight>open</SyndicationRight>
  <AdultContent>false</AdultContent>
  <Language>en-us</Language>
  <OutputEncoding>UTF-8</OutputEncoding>
  <InputEncoding>UTF-8</InputEncoding>
</OpenSearchDescription>
```

Type

OpenSearch description documents are referred to via the following type:

```
application/opensearchdescription+xml
```

This type is pending IANA registration.

Extensibility

OpenSearch description documents can be extended with foreign markup provided that all foreign elements and attributes are associated with an explicit XML namespace distinct from that of the core OpenSearch format. When possible, the foreign XML namespace URI should resolve to a document that indicates the intention and format of the extension. Clients that encounter unrecognized foreign markup should continue to process the document as if the markup did not appear.

OpenSearch description elements

The "OpenSearchDescription" element

The root node of the OpenSearch description document.

> Parent: None
> Requirements: The element **must** appear exactly once as the root node of the document.

Example:

```
<OpenSearchDescription xmlns="http://a9.com/-/spec/opensearch/1.1/">
  <!--- ... --->
</OpenSearchDescription>
```

The "ShortName" element

Contains a brief human-readable title that identifies this search engine.

Parent: OpenSearchDescription
Restrictions: The value must contain 16 or fewer characters of plain text. The value must not contain HTML or other markup.
Requirements: This element **must** appear exactly once.

Example:

```
<ShortName>Web Search</ShortName>
```

The "Description" element

Contains a human-readable text description of the search engine.

Parent: OpenSearchDescription
Restrictions: The value must contain 1024 or fewer characters of plain text. The value must not contain HTML or other markup.
Requirements: This element **must** appear exactly once.

Example:

```
<Description>Use Example.com to search the Web.</Description>
```

The "Url" element

Describes an interface by which a client can make requests for an external resource, such as search results, search suggestions, or additional description documents.

Parent: OpenSearchDescription
Attributes:

`template` - The URL template to be processed according to the OpenSearch URL template syntax.

Requirements: This attribute is **required**.

`type` - The MIME type of the resource being described.

Restrictions: The value must be a valid MIME type.
Requirements: This attribute is **required**.

`rel` - The role of the resource being described in relation to the description document.

Restrictions: Contains a space-delimited list of valid rel value tokens. See the Url rel values specification for allowed rel values.
Default: "results"
Requirements: This attribute is optional.

`indexOffset` - The index number of the first search result.

Restrictions: The value must be an integer.
Default: "1"

Requirements: This attribute is optional.

`pageOffset` - The page number of the first set of search results.

Restrictions: The value must be an integer.
Default: "1"
Requirements: This attribute is optional.

Requirements: This element **must** appear one or more times.

Example of a Url element describing a request for a HTML search results page:

```
<Url type="text/html"
     template="http://example.com/search?q={searchTerms}&amp;pw={startPage?}" />
```

Example of a Url element describing a request for search results over RSS, starting with element index 0:

```
<Url type="application/rss+xml"
     indexOffset="0"
     rel="results"
     template="http://example.com/?q={searchTerms}&amp;start={startIndex?}&amp;format=rss" />
```

Example of a Url element describing a request for search suggestions over JSON:

```
<Url type="application/json"
     rel="suggestions"
     template="http://example.com/suggest?q={searchTerms}" />
```

Example of a Url element describing a request to refresh the OpenSearch description document itself:

```
<Url type="application/opensearchdescription+xml"
     rel="self"
     template="http://example.com/osd.xml" />
```

Url rel values

Rel attribute strings can contain a space-delimited list of one or more semantically meaningful rel value tokens. An empty rel attribute value should be treated by the client as if the rel attribute was not present at all.

If a client does not recognize the semantic meaning of any rel value token, then the containing Url should be ignored by the client.

Rel value tokens may be either fully qualified tokens (e.g., "http://example.com/rel#foo") or unqualified tokens (e.g., "results").

All fully qualified tokens must be a **valid URL**. The semantic meaning of any fully qualified token is outside the scope of this specification, but convention dictates that the URL should resolve to a resource that describes the relationship.

All unqualified tokens must be a lowercase alphanumeric string of the form [a-z][a-z\-]+. Only those tokens listed below have meaning defined in this specification.

Rel values:

> `"results"` (default)
>
> > Represents a request for search results in the specified format.
>
> `"suggestions"`
>
> > Represents a request for search suggestions in the specified format. See the OpenSearch Suggestions extension for further details.
>
> `"self"`
>
> > Represents the canonical URL of this description document.
>
> `"collection"`
>
> > Represents a request for a set of resources.

The "Contact" element

Contains an email address at which the maintainer of the description document can be reached.

> Parent: OpenSearchDescription
> Restrictions: The value must conform to the requirements of Section 3.4.1 "Addr-spec specification" in **RFC 2822**.
> Requirements: This element may appear zero or one time.

Example:

```
<Contact>admin@example.com</Contact>
```

The "Tags" element

Contains a set of words that are used as keywords to identify and categorize this search content. Tags must be a single word and are delimited by the space character (' ').

> Parent: OpenSearchDescription
> Restrictions: The value must contain 256 or fewer characters of plain text. The value must not contain HTML or other markup.
> Requirements: This element may appear zero or one time.

Example:

```
<Tags>example web</Tags>
```

The "LongName" element

Contains an extended human-readable title that identifies this search engine.

Search clients should use the value of the ShortName element if this element is not available.

Parent: OpenSearchDescription
Restrictions: The value must contain 48 or fewer characters of plain text. The value must not contain HTML or other markup.
Requirements: This element may appear zero or one time.

Example:

```
<LongName>Example.com Web Search</LongName>
```

The "Image" element

Contains a URL that identifies the location of an image that can be used in association with this search content.

Image sizes are offered as a hint to the search client. The search client will choose the most appropriate image for the available space and should give preference to those listed first in the OpenSearch description document. Square aspect ratios are recommended. When possible, search engines should offer a 16x16 image of type "image/x-icon" or "image/vnd.microsoft.icon" (the ***Microsoft ICON format****) and a 64x64 image of type "image/jpeg" or "image/png".*

Parent: OpenSearchDescription
Attributes:

`height` – Contains the height, in pixels, of this image.

Restrictions: The value must be a non-negative integer.
Requirements: This attribute is optional.

`width` – Contains the width, in pixels, of this image.

Restrictions: The value must be a non-negative integer.
Requirements: This attribute is optional.

`type` – Contains the the MIME type of this image.

Restrictions: The value must be a valid MIME type.
Requirements: This attribute is optional.

Restrictions: The value must be a URI.
Requirements: This element may appear zero, one, or more times.

Example:

```
<Image height="16" width="16" type="image/x-icon">http://example.com/favicon.ico</Image>

<Image height="64" width="64" type="image/png">http://example.com/websearch.png</Image>
```

The "Query" element

Defines a search query that can be performed by search clients. Please see the OpenSearch Query element specification for more information.

OpenSearch description documents should include at least one Query element of role="example" that is *expected to return search results. Search clients may use this example query to validate that the search engine is working properly.*

Parent: OpenSearchDescription
Requirements: This element may appear zero or more times.

Example:

```
<Query role="example" searchTerms="cat" />
```

The "Developer" element

Contains the human-readable name or identifier of the creator or maintainer of the description document.

The developer is the person or entity that created the description document, and may or may not be the owner, author, or copyright holder of the source of the content itself.

Parent: OpenSearchDescription
Restrictions: The value must contain 64 or fewer characters of plain text. The value must not contain HTML or other markup.
Requirements: This element may appear zero or one time.

Example:

```
<Developer>Example.com Development Team</Developer>
```

The "Attribution" element

Contains a list of all sources or entities that should be credited for the content contained in the search feed.

Parent: OpenSearchDescription
Restrictions: The value must contain 256 or fewer characters of plain text. The value must not contain HTML or other markup.
Requirements: This element may appear zero or one time.

Example:

```
<Attribution>Search data copyright Example.com, Inc.</Attribution>
```

The "SyndicationRight" element

Contains a value that indicates the degree to which the search results provided by this search engine can be queried, displayed, and redistributed.

Parent: OpenSearchDescription
Values: The value must be one of the following strings (case insensitive):

`"open"` –

The search client may request search results.
The search client may display the search results to end users.

The search client may send the search results to other search clients.

`"limited"` –

The search client may request search results.
The search client may display the search results to end users.
The search client may not send the search results to other search clients.

`"private"` –

The search client may request search results.
The search client may not display the search results to end users.
The search client may not send the search results to other search clients.

`"closed"` -

The search client may not request search results.

Default: "open"
Requirements: This element may appear zero or one time.

Example:

```
<SyndicationRight>open</SyndicationRight>
```

The "AdultContent" element

Contains a boolean value that should be set to true if the search results may contain material intended only for adults.

As there are no universally applicable guidelines as to what constitutes "adult" content, the search engine should make a good faith effort to indicate when there is a possibility that search results may contain material inappropriate for all audiences.

Parent: OpenSearchDescription
Values:

The values "false", "FALSE", "0", "no", and "NO" will be considered boolean FALSE; all other strings will be considered boolean TRUE.

Default: "false"
Requirements: This element may appear zero or one time.

Example:

```
<AdultContent>false</AdultContent>
```

The "Language" element

Contains a string that indicates that the search engine supports search results in the specified language.

An OpenSearch description document should include one "Language" element for each language that the search engine supports. If the search engine also supports queries for any arbitrary language then the

OpenSearch description document should include a Language element with a value of "". The "language" template parameter in the OpenSearch URL template can be used to allow the search client to choose among the available languages.*

Parent: OpenSearchDescription
Restrictions: The value must conform to the **XML 1.0 Language Identification**, as specified by **RFC 5646**. In addition, the value of "*" will signify that the search engine does not restrict search results to any particular language.
Default: "*".
Requirements: This element may appear zero, one, or more times.

Example:

```
<Language>en-us</Language>

<Language>*</Language>
```

The "InputEncoding" element

Contains a string that indicates that the search engine supports search requests encoded with the specified character encoding.

An OpenSearch description document should include one "InputEncoding" element for each character encoding that can be used to encode search requests. The "inputEncoding" template parameter in the OpenSearch URL template can be used to require the search client to identify which encoding is being used to encode the current search request.

Parent: OpenSearchDescription
Restrictions: The value must conform to the **XML 1.0 Character Encodings**, as specified by the **IANA Character Set Assignments**.
Default: "UTF-8".
Requirements: This element may appear zero, one, or more times.

Example:

```
<InputEncoding>UTF-8</InputEncoding>
```

The "OutputEncoding" element

Contains a string that indicates that the search engine supports search responses encoded with the specified character encoding.

An OpenSearch description document should include one "OutputEncoding" element for each character encoding that can be used to encode search responses. The "outputEncoding" template parameter in the OpenSearch URL template can be used to allow the search client to choose a character encoding in the search response.

Parent: OpenSearchDescription
Restrictions: The value must conform to the **XML 1.0 Character Encodings**, as specified by the **IANA Character Set Assignments**.
Default: "UTF-8".
Requirements: This element may appear zero, one, or more times.

Example:

```
<OutputEncoding>UTF-8</OutputEncoding>
```

Autodiscovery

Search engines that publish OpenSearch description documents can assist search clients in the discovery of OpenSearch interfaces through the use of "link" elements. Search engines that support OpenSearch should include a reference to the related OpenSearch description document on each page of search results.

Autodiscovery in RSS/Atom

RSS and Atom documents may reference related OpenSearch description documents via the Atom 1.0 "link" element, as specified in Section 4.2.7 of **RFC 4287**.

The "rel" attribute of the link element should contain the value "search" when referring to OpenSearch description documents. This relationship value is pending IANA registration. The reuse of the Atom link element is recommended in the context of other syndication formats that do natively support comparable functionality.

The following restrictions apply:

- The "type" attribute must contain the value "application/opensearchdescription+xml".
- The "rel" attribute must contain the value "search".
- The "href" attribute must contain a URI that resolves to an OpenSearch description document.
- The "title" attribute may contain a human-readable plain text string describing the search engine.

Example of Atom-based search results that include an OpenSearch autodiscovery link element:

```
<?xml version="1.0" encoding="UTF-8"?>
<feed xmlns="http://www.w3.org/2005/Atom"
      xmlns:opensearch="http://a9.com/-/spec/opensearch/1.1/">
  <!-- ... --->
  <link rel="search"
        href="http://example.com/opensearchdescription.xml"
        type="application/opensearchdescription+xml"
        title="Content Search" />
  <!-- ... --->
</feed>
```

Example of RSS-based search results that include an OpenSearch autodiscovery link element:

```
<?xml version="1.0" encoding="UTF-8"?>
<rss version="2.0"
     xmlns:atom="http://www.w3.org/2005/Atom">
  <channel>
    <!--- ... --->
    <atom:link rel="search"
               href="http://example.com/opensearchdescription.xml"
               type="application/opensearchdescription+xml"
               title="Content Search" />
    <!--- ... --->
  </channel>
</rss>
```

Autodiscovery in HTML/XHTML

HTML and XHTML documents may reference related OpenSearch description documents via the **HTML 4.0 <link/> element**.

The following restrictions apply:

- The "type" attribute must contain the value "application/opensearchdescription+xml".
- The "rel" attribute must contain the value "search".
- The "href" attribute must contain a URI that resolves to an OpenSearch description document.
- The "title" attribute may contain a human-readable plain text string describing the search engine.
- The HTML <head/> element should include a "profile" attribute that contains the value "http://a9.com/-/spec/opensearch/1.1/".

Example of an HTML document that includes OpenSearch autodiscovery link elements:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en" dir="ltr">
  <head profile="http://a9.com/-/spec/opensearch/1.1/">
    <!--- ... --->
    <link rel="search"
          type="application/opensearchdescription+xml"
          href="http://example.com/content-search.xml"
          title="Content search" />
    <link rel="search"
          type="application/opensearchdescription+xml"
          href="http://example.com/comment-search.xml"
          title="Comments search" />
    <!--- ... --->
  </head>
  <body>
    <!--- ... --->
  </body>
</html>
```

MIME type application/opensearchdescription+xml

For the purposes of **RFC 4288** section 4.10 this specification contains the registration template for MIME type `application/opensearchdescription+xml`. *This type is pending IANA registration.*

```
Type name:      application

Subtype name:   opensearchdescription+xml

Required parameters:
                There are no required parameters.

Optional parameters:
                charset (defaults to "UTF-8")

Encoding considerations:
                Identical to those of "application/xml" as described
                in RFC 3023; especially "UTF-8" (RFC 3629) and its
                proper subset "US-ASCII" are supposed to work.

                For non-ASCII documents served as "text/xml" the
                "charset" parameter is required; this might be
                relevant when authors are unable to configure the
                server hosting their OSD (OpenSearch Description
                document).
```

```
   Security considerations:
                   All general security and privacy considerations for
                   sending queries to servers specified in an URL are
                   applicable.

                   Where clients support the optional update feature in
                   OSDs it affects the privacy of users.

                   The Ecmascript API AddSearchProvider() typically
                   enforces a "same origin" policy for the OSD; the URL
                   element within the OSD can designate a third party
                   as search provider.

                   An OSD can claim to be a search description for X,
                   but actually do something else.

   Interoperability considerations:
                   OSDs use the <http://a9.com/-/spec/opensearch/1.1/>
                   XML name space, optionally in conjunction with other
                   XML name spaces for extensions or application specific
                   purposes.

   Published specification:
                   <http://www.opensearch.org/Specifications/OpenSearch/1.1>

   Applications that use this media type:
                   Various Web browsers, search engines, and software
                   libraries support OSDs.  The "search" link relation
                   is used on many Web pages with this media type.

                   The Ecmascript API AddSearchProvider() documented for
                   WhatWG HTML uses this media type.

   Additional information:
                   OSDs have no "magic numbers" as defined in RFC 4288.
                   There are no special "common file name extensions" for
                   OSDs, OSDs are XML documents.  If specific extensions
                   are desired the conventional ".osdx" or ".a9.xml" might
                   do the trick.

   Person & email address to contact for further information:
                   <http://groups.google.com/group/opensearch>
                   <http://www.opensearch.org/Community/Guidelines>

   Intended usage: COMMON

   Restrictions on usage:
                   None

   Author:         DeWitt Clinton

   Change controller:
                   <http://www.opensearch.org/>
```

OpenSearch URL template syntax

Overview

The OpenSearch URL template format can be used to represent a parameterized form of the URL by which a search engine is queried.

The search client will process the URL template and attempt to replace each instance of a template parameter, generally represented in the form `{name}`*, with a value determined at query time.*

By default, parameter names are considered part of the OpenSearch 1.1 template namespace, and definitions for a set of core search parameter names are provided in this specification. However, search engines and search clients can adopt new parameter names using an extensibility mechanism based on the XML namespace prefix conventions.

Examples

Example of a search URL template that contains a template parameter:

```
http://example.com/search?q={searchTerms}
```

Example of a search URL template that contains an optional template parameter:

```
http://example.com/feed/{startPage?}
```

Example of a search URL template that contains an optional template parameter in an extended namespace, shown in the context of a Url element:

```
<Url type="application/rss+xml"
     xmlns:example="http://example.com/opensearchextensions/1.0/"
     template="http://example.com?q={searchTerms}&amp;c={example:color?}"/>
```

Context

This specification refers to the use of the OpenSearch URL template syntax specifically within the context of the "Url" element in an OpenSearch description document.

Template grammar

The grammar of an OpenSearch URL template is defined by the following set of ABNF rules, as specified in **RFC 2234**.

The grammar rules defined in this document build upon a subset of the rules defined for the Uniform Resource Identifier (URI): Generic Syntax in **RFC 3986**. For brevity, rules already stated in **RFC 3986** are referenced in this document by rule name alone and are not restated here in their entirety.

```
ttemplate      = tscheme ":" thier-part [ "?" tquery ] [ "#" tfragment ]
tscheme        = *( scheme / tparameter )
thier-part     = "//" tauthority ( tpath-abempty / tpath-absolute / tpath-rootless / path-empty )
tauthority     = [ tuserinfo "@" ] thost [ ":" tport ]
tuserinfo      = *( userinfo / tparameter )
thost          = *( host / tparameter )
tport          = *( port / tparameter )
tpath-abempty  = *( "/" tsegment )
tsegment       = *( segment / tparameter )
tpath-absolute = "/" [ tsegment-nz *( "/" tsegment ) ]
tsegment-nz    = *( segment-nz / tparameter )
tpath-rootless = tsegment-nz *( "/" tsegment )
tparameter     = "{" tqname [ tmodifier ] "}"
tqname         = [ tprefix ":" ] tlname
tprefix        = *pchar
tlname         = *pchar
```

```
tmodifier       = "?"
tquery          = *( query / tparameter )
tfragement      = *( fragement / tparameter )
```

Substitution rules

The search client **must** replace every instance of a template parameter with a value before the search request is performed.

If a search engine wishes to indicate that a template parameter is optional and can be replaced with the empty string, then the "?" notation described in the section on optional template parameters should be used.

Parameter names

A parameter name consists of an optional parameter name prefix followed by the local parameter name. If the parameter name prefix is present then it will be separated from the local parameter name with the ":" character. All parameter names are associated with a parameter namespace. In the case of unqualified parameter names, the local parameter name is implicitly associated with the OpenSearch 1.1 namespace. In the case of fully qualified parameter names, the local parameter name is explicitly associated with an external namespace via the parameter name prefix.

Case sensitivity of parameter names

Both the parameter name prefix and the local parameter name are case sensitive.

Parameter name prefix

A parameter name prefix associates a local parameter name with a parameter namespace. All parameter name prefixes must be previously declared as an **XML namespace** prefix on the containing element or ancestor elements.

The choice of prefix is at the discretion of the author of the OpenSearch description document. Search clients should make no assumption as to the meaning of any particular literal prefix string, and should rely exclusively on the mapping of prefix strings to XML namespace declarations when parsing fully qualified parameter names.

Example of two equivalent URL templates that will be processed identically by search clients:

```
<Url type="application/rss+xml"
     xmlns:a="http://example.com/extensions/"
     template="http://example.com?q={a:localname?}"/>

<Url type="application/rss+xml"
     xmlns:b="http://example.com/extensions/"
     template="http://example.com?q={b:localname?}"/>
```

Unqualified parameter names

Unqualified parameter names consist of only a local parameter name and do not include a parameter name prefix. Unqualified parameter names in OpenSearch URL templates are implicitly associated with the OpenSearch 1.1 namespace.

This specification includes an exhaustive list of all unqualified OpenSearch 1.1 parameter names.

Example of an unqualified parameter name:

```
<Url type="application/rss+xml"
     template="http://example.com/?q={searchTerms}"/>
```

Fully qualified parameter names

Fully qualified parameter names consist of a parameter name prefix, followed by the ":" character, followed by the local parameter name. Fully qualified parameter names are associated with the namespace identified by the paramater name prefix, as it appears as an XML namespace declaration on the containing element or ancestor elements.

Example of a fully qualified parameter name:

```
<Url type="application/rss+xml"
     xmlns:example="http://example.com/opensearchextensions/1.0/"
     template="http://example.com?f={example:format?}"/>
```

Required template parameters

Required template parameters are template parameters that do not contain a template parameter modifier. The search client may use the default value if one is known, but may not use the empty string as a value.

Example of a required template parameter:

```
{searchTerms}
```

Optional template parameters

Optional template parameters are template parameters that contain a template parameter modifier equal to "?". The search client may use the empty string as a value if no other value is available.

Example of an optional template parameter:

```
{startPage?}
```

OpenSearch 1.1 parameters

The following local parameter names are identified with the OpenSearch 1.1 namespace. The list is exhaustive; only the local parameter names listed below may appear unqualified in an OpenSearch URL template.

Search clients should be prepared to substitute reasonable values for these parameter names when they appear in an OpenSearch URL template.

The "searchTerms" parameter

Replaced with the keyword or keywords desired by the search client.

Restrictions: The value must be URL-encoded.

The "count" parameter

Replaced with the number of search results per page desired by the search client.

Search clients should anticipate that the value of the "count" parameter may not be honored by the search engine, and should rely exclusively on the contents of the "itemsPerPage" response element in calculating actual page size.

Restrictions: The value must be a non-negative integer.

The "startIndex" parameter

Replaced with the index of the first search result desired by the search client.

Restrictions: The value must be an integer.
Default: The value specified by the "indexOffset" attribute of the containing Url element.

The "startPage" parameter

Replaced with the page number of the set of search results desired by the search client.

Restrictions: The value must be an integer.
Default: The value specified by the "pageOffset" attribute of the containing Url element.

The "language" parameter

Replaced with a string that indicates that the search client desires search results in the specified language.

An OpenSearch description document should include one "Language" element for each language that the search engine supports. If the search engine also supports queries for any arbitrary language then the OpenSearch description document should include a Language element with a value of "". The "language" template parameter in the OpenSearch URL template can be used to allow the search client to choose among the available languages.*

Restrictions: The value must conform to the **XML 1.0 Language Identification**, as specified by **RFC 5646**. In addition, a value of "*" will signify that the search client desires search results in any language.
Default: "*"

The "inputEncoding" parameter

Replaced with a string that indicates that the search client is performing the search request encoded with the specified character encoding.

An OpenSearch description document should include one "InputEncoding" element for each character encoding that can be used to encode search requests. The "inputEncoding" template parameter in the OpenSearch URL template can be used to require the search client to identify which encoding is being used to encode the current search request.

Restrictions: The value must conform to the **XML 1.0 Character Encodings**, as specified by the **IANA Character Set Assignments**.

Default: "UTF-8"

The "outputEncoding" parameter

Replaced with a string that indicates that the search client desires a search response encoding with the specified character encoding.

An OpenSearch description document should include one "OutputEncoding" element for each character encoding that can be used to encode search responses. The "outputEncoding" template parameter in the OpenSearch URL template can be used to allow the search client to choose a character encoding in the search response.

Restrictions: The value must conform to the **XML 1.0 Character Encodings**, as specified by the **IANA Character Set Assignments**.
Default: "UTF-8"

OpenSearch Query element

Overview

The OpenSearch Query element can be used to define a specific search request that can be performed by a search client.

The Query element attributes correspond to the search parameters in a URL template. The core set of search parameters are explicitly defined as Query attributes, and custom parameters can be added via namespaces as needed.

Authors should provide at least one Query element of role="example" in each OpenSearch description document so that search clients can test the search engine. Search engines should include a Query element of role="request" in each search response so that search clients can recreate the current search.

Examples

Example of a Query element used in an OpenSearch description document to provide an example search request for search clients:

```
<Query role="example" searchTerms="cat" />
```

Example of a Query element used in an OpenSearch response to echo back the original search request:

```
<Query role="request" searchTerms="cat" startPage="1" />
```

Example of a Query element used in an OpenSearch response to correct the spelling of "OpenSurch":

```
<Query role="correction" searchTerms="OpenSearch" totalResults="854000" title="Spelling correction" />
```

Example of a Query element using an extended parameter:

```
<Query xmlns:custom="http://example.com/opensearchextensions/1.0/"
       role="example"
       searchTerms="cat"
       custom:color="blue"
       title="Sample search" />
```

Example of a Query element using a extended role:

```
<Query xmlns:custom="http://example.com/opensearchextensions/1.0/"
       role="custom:synonym"
       title="Synonym of 'cat'"
       searchTerms="feline" />
```

Detailed example of a set of Query elements used in the context of an Atom-based OpenSearch response:

```
<?xml version="1.0" encoding="UTF-8"?>
<feed xmlns="http://www.w3.org/2005/Atom"
      xmlns:opensearch="http://a9.com/-/spec/opensearch/1.1/">
  <!--- ... --->
  <opensearch:Query role="request" searchTerms="General Motors annual report" />
  <opensearch:Query role="related" searchTerms="GM" title="General Motors stock symbol" />
  <opensearch:Query role="related" searchTerms="automotive industry revenue" />
  <opensearch:Query role="subset" searchTerms="General Motors annual report 2005"
  <opensearch:Query role="superset" searchTerms="General Motors" />
  <!-- ... -->
</feed>
```

The "Query" element

Describes a specific search request that can be made by the search client.

Attributes:

`role` - Contains a string identifying how the search client should interpret the search request defined by this Query element.

Restrictions: See the role values specification for allowed role values.
Requirements: This attribute is **required**.

`title` - Contains a human-readable plain text string describing the search request.

Restrictions: The value must contain 256 or fewer characters of plain text. The value must not contain HTML or other markup.
Requirements: This attribute is optional.

`totalResults` - Contains the expected number of results to be found if the search request were made.

Restrictions: The value is a non-negative integer.
Requirements: This attribute is optional.

`searchTerms` - Contains the value representing the "searchTerms" as an OpenSearch 1.1 parameter.

Restrictions: See the "searchTerms" parameter.
Requirements: This attribute is optional.

`count` - Contains the value representing the "count" as a OpenSearch 1.1 parameter.

Restrictions: See the "count" parameter.
Requirements: This attribute is optional.

`startIndex` - Contains the value representing the "startIndex" as an OpenSearch 1.1 parameter.

Restrictions: See the "startIndex" parameter.
Requirements: This attribute is optional.

`startPage` - Contains the value representing the "startPage" as an OpenSearch 1.1 parameter.

Restrictions: See the "startPage" parameter.
Requirements: This attribute is optional.

`language` - Contains the value representing the "language" as an OpenSearch 1.1 parameter.

Restrictions: See the "language" parameter.
Requirements: This attribute is optional.

`inputEncoding` - Contains the value representing the "inputEncoding" as an OpenSearch 1.1 parameter.

Restrictions: See the "inputEncoding" parameter.
Requirements: This attribute is optional.

`outputEncoding` - Contains the value representing the "outputEncoding" as an OpenSearch 1.1 parameter.

Restrictions: See the "outputEncoding" parameter.
Requirements: This attribute is optional.

Example:

```
<Query role="example" searchTerms="cat" />
```

Query element extensibility

The Query element may contain additional attributes if the extended attributes are associated with a namespace. Search clients should interpret extended attributes to represent the corresponding template parameter by the same name in the specified namespace.

Example of a Query element representing a search request that contains an extended attribute that corresponds to an extended search parameter:

```
<OpenSearchDescription xmlns="http://a9.com/-/spec/opensearch/1.1/"
                       xmlns:custom="http://example.com/opensearchextensions/1.0/">
  <Url type="text/html"
       template="http://example.com/search?color={custom:color?}" />
  <Query role="example"  custom:color="blue" />
  <!-- ... -->
</OpenSearchDescription>
```

Role values

A role value consists of an optional prefix followed by the local role value. If the prefix is present it will be separated from the local role value with the ":" character. All role values are associated with a namespace, either implicitly in the case of local role values, or explicitly via a prefix in the case of fully qualified role values.

Role extensibility

The role attribute may take on values beyond those specified in this document provided they are fully qualified with a prefix and associated with a declared namespace. Clients that encounter unrecognized role values should continue to process the document as if the Query element containing the unrecognized role value did not appear.

Role prefix

A role prefix associates a local role name with a namespace. All prefixes must be previously declared as an XML namespace prefix on the containing Query element or ancestor elements.

Local role values

Local role values are not preceded by a prefix. Local role values are associated with the OpenSearch 1.1 namespace.

The following role values are identified with the OpenSearch 1.1 namespace. The list is exhaustive; only the role values listed below may appear in the OpenSearch 1.1 namespace.

Role values:

`"request"`

Represents the search query that can be performed to retrieve the same set of search results.

`"example"`

Represents a search query that can be performed to demonstrate the search engine.

`"related"`

Represents a search query that can be performed to retrieve similar but different search results.

`"correction"`

Represents a search query that can be performed to improve the result set, such as with a spelling correction.

`"subset"`

Represents a search query that will narrow the current set of search results.

`"superset"`

Represents a search query that will broaden the current set of search results.

Example of a local role value:

```
<Query role="related"
       title="A related search"
       searchTerms="tiger" />
```

Fully qualified role values

Fully qualified role values are preceded by a prefix. Fully qualified role values are associated with the namespace identified by the prefix on the containing Query element or ancestor elements.

Example of a fully qualified role value:

```
<Query xmlns:custom="http://example.com/opensearchextensions/1.0/"
       role="custom:synonym"
       title="Synonyms of 'cat'"
       searchTerms="feline" />
```

OpenSearch response elements

The OpenSearch response elements can be used by search engines to augment existing XML formats with search-related metadata.

OpenSearch response elements are typically found augmenting search results returned in list-based XML syndication formats, such as RSS 2.0 and Atom 1.0, but may be used in other contexts without restriction.

Examples of OpenSearch responses

Example of OpenSearch response elements in RSS 2.0

Example of a page of search results in the RSS 2.0 format:

```
<?xml version="1.0" encoding="UTF-8"?>
<rss version="2.0"
     xmlns:opensearch="http://a9.com/-/spec/opensearch/1.1/"
     xmlns:atom="http://www.w3.org/2005/Atom">
  <channel>
    <title>Example.com Search: New York history</title>
    <link>http://example.com/New+York+history</link>
    <description>Search results for "New York history" at Example.com</description>
    <opensearch:totalResults>4230000</opensearch:totalResults>
    <opensearch:startIndex>21</opensearch:startIndex>
    <opensearch:itemsPerPage>10</opensearch:itemsPerPage>
    <atom:link rel="search" type="application/opensearchdescription+xml" href="http://example.com/open
    <opensearch:Query role="request" searchTerms="New York History" startPage="1" />
    <item>
      <title>New York History</title>
      <link>http://www.columbia.edu/cu/lweb/eguids/amerihist/nyc.html</link>
      <description>
        ... Harlem.NYC - A virtual tour and information on
        businesses ...  with historic photos of Columbia's own New York
        neighborhood ... Internet Resources for the City's History. ...
      </description>
    </item>
  </channel>
</rss>
```

Example of OpenSearch response elements in Atom 1.0

Example of a page of search results in the Atom 1.0 format:

```
<?xml version="1.0" encoding="UTF-8"?>
<feed xmlns="http://www.w3.org/2005/Atom"
      xmlns:opensearch="http://a9.com/-/spec/opensearch/1.1/">
  <title>Example.com Search: New York history</title>
  <link href="http://example.com/New+York+history"/>
  <updated>2003-12-13T18:30:02Z</updated>
  <author>
    <name>Example.com, Inc.</name>
  </author>
  <id>urn:uuid:60a76c80-d399-11d9-b93C-0003939e0af6</id>
  <opensearch:totalResults>4230000</opensearch:totalResults>
  <opensearch:startIndex>21</opensearch:startIndex>
  <opensearch:itemsPerPage>10</opensearch:itemsPerPage>
  <opensearch:Query role="request" searchTerms="New York History" startPage="1" />
  <link rel="alternate" href="http://example.com/New+York+History?pw=3" type="text/html"/>
  <link rel="self" href="http://example.com/New+York+History?pw=3&amp;format=atom" type="application/a
  <link rel="first" href="http://example.com/New+York+History?pw=1&amp;format=atom" type="application/
  <link rel="previous" href="http://example.com/New+York+History?pw=2&amp;format=atom" type="applicati
  <link rel="next" href="http://example.com/New+York+History?pw=4&amp;format=atom" type="application/a
  <link rel="last" href="http://example.com/New+York+History?pw=42299&amp;format=atom" type="applicati
  <link rel="search" type="application/opensearchdescription+xml" href="http://example.com/opensearchd
  <entry>
    <title>New York History</title>
    <link href="http://www.columbia.edu/cu/lweb/eguids/amerihist/nyc.html"/>
    <id>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344efa6a</id>
    <updated>2003-12-13T18:30:02Z</updated>
    <content type="text">
      ... Harlem.NYC - A virtual tour and information on
      businesses ...  with historic photos of Columbia's own New York
      neighborhood ... Internet Resources for the City's History. ...
    </content>
  </entry>
</feed>
```

Elements

The "totalResults" element

The number of search results available for the current search.

If the totalResults element does not appear on the page then the search client should consider the current page to be the last page of search results.

> Restrictions: The value must be a non-negative integer.
> Default: The default value is equal to the offset index of the last search result on the current page.
> Requirements: The element may appear zero or one time.

Example:

```
<totalResults>492420</totalResults>
```

The "startIndex" element

The index of the first search result in the current set of search results.

If the startIndex element does not appear on the page then the search client should consider the current *page to be the first page of search results.*

> Restrictions: The value must an integer.
> Default: The default value is equal to the value of the "indexOffset" attribute of the "Url" element in the OpenSearch description document.
> Requirements: The element may appear zero or one time.

Example:

```
<startIndex>11</startIndex>
```

The "itemsPerPage" element

The number of search results returned per page.

If the itemsPerPage element does not appear on the page then the search client should use the number of items of the current page as the default page size.

> Restrictions: The value must a non-negative integer.
> Default: The default value is equal to the number of search results on the current page.
> Requirements: The element may appear zero or one time.

Example:

```
<itemsPerPage>10</itemsPerPage>
```

The "Query" element

Defines a search query that can be performed by search clients. Please see the OpenSearch Query element specification for more information.

Search results should include a Query element of type="request" that can be used to recreate the search request that generate the current search response.

> Requirements: The element may appear zero or more times.

Example:

```
<Query role="request" searchTerms="cat" />
```

Response metadata in HTML/XHTML

OpenSearch response metadata may be included in well-formed HTML/XHTML via the **HTML 4.0.1 "meta" element**.

The following meta element "name" attribute values are recognized under the profile associated with the OpenSearch 1.1 namespace:

> `"totalResults"` - Corresponds to value of the "totalResults" element.
> `"startIndex"` - Corresponds to value of the "startIndex" element.
> `"itemsPerPage"` - Corresponds to value of the "itemsPerPage" element.

Example of a page of search results in the XHTML 1.0 format:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
  <head profile="http://a9.com/-/spec/opensearch/1.1/" >
    <title>Example.com Search: New York history</title>
    <link rel="search"
          type="application/opensearchdescription+xml"
          href="http://example.com/opensearchdescription.xml"
          title="Example.com Web Search" />
    <meta name="totalResults" content="4230000"/>
    <meta name="startIndex" content"1"/>
    <meta name="itemsPerPage" content="10"/>
  </head>
  <body>
     <ul>
       <li>
         <a href="http://www.columbia.edu/cu/lweb/eguids/amerihist/nyc.html">
           New York History
         </a>
         <div>
           ... Harlem.NYC - A virtual tour and information on
           businesses ...  with historic photos of Columbia's own New York
           neighborhood ... Internet Resources for the City's History. ...
         </div>
       </li>
       <!-- ... -->
     </ul>
   </body>
</html>
```

Author

DeWitt Clinton <dewitt@opensearch.org>

Contributors

Joel Tesler <tesler@a9.com>, Michael Fagan <mifa@a9.com>, Joe Gregorio <joe@bitworking.org>, Aaron Sauve <aaronsa@microsoft.com>, James Snell <jasnell@us.ibm.com>

License

Category: Specification