Understanding OpenSearch Standards

What is OpenSearch? Well, the opensearch.org Web site says it best. “OpenSearch is a collection of simple formats for the sharing of search results.” In OpenSearch, there are four basic parts to the standard:

- A Description Document that is used to describe the search engine
- Search client applications
- OpenSearch response elements
- A result set

In this section, we’ll look at the OpenSearch 1.1 standard, focusing on the description file and some of the response elements. In Search Server 2008, the search client is the Keyword Query Web part, and the result set is the page of Web parts that displays the results from one or more indexes. When you build an FLD in your environment or when you export an FLD to a file system, you’ll be working with an OpenSearch-compliant XML file.

Note If you want to read the OpenSearch documents directly, you can do so at OpenSearch.org. The vast majority of this section is taken directly from the OpenSearch 1.1 documentation at the OpenSearch Web site.

OpenSearch Description Documents

The Description Document is used to describe the Web interface of the OpenSearch-compliant search engine. The document is written in XML. The document consists of description elements that form the overall Description Document. The description elements are laid out in the OpenSearch 1.1 standard and are summarized here for your consideration.

Note All XML files start with an XML declaration that specifies, among other things, which version of XML is being used. This is the first line in the XML document and normally looks like the following:

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

The grammar of the URL template used by OpenSearch 1.1 is defined by the Augmented Backus-Naur Form (ABNF) rules from RFC 2234. You can think of a parameter as a variable that will need a value supplied in order for the URL template to work. Anytime there is a parameter in the template, a value must be entered by the search client before the search request can be performed. Parameter names consist 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.

Now, let’s discuss each Description Document element individually in the follow sections.

OpenSearchDescription Element

The *OpenSearchDescription* element is the first entry in the Description Document after the XML declaration statement. Hence, this element is the overall opening and closing tag for the Description Document. The following is what the opening and closing tags would look like:

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

The *OpenSearchDescription* opening and closing tags must appear once in the Description Document and cannot appear more than once. In addition, this tag is a root tag because it has no parents inside which it must operate.

ShortName Element

The *ShortName* element allows you to assign a vanity, or easily remembered, name to the search engine that you are trying to describe. It must appear inside the *OpenSearchDescription* tags and may contain no more than 16 plain-text characters. The name may not include HTML or other markup language. This element must also appear in the Description Document. The entry would look like the following:

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

LongName Element

If 16 characters aren't enough, you can use the *LongName* element to assign a vanity name to the search engine that you are describing in the Description Document. The *LongName* element can contain up to 48 characters and cannot contain any HTML or other markup characters. The XML will look like the following:

```
<LongName>contoso.msft Search Engine</LongName>
```

Description Element

The *Description* element allows you to describe the search engine in the Description Document. While this element has the same title as the overall document, don't confuse the two. The *Description* element is a customized text description of the search engine that the Description Document is describing. The Description Document contains many elements, and a *Description* element is one of them.

You can put up to 1,024 characters in the Description element. As with the *ShortName* element, it cannot contain any HTML or other markup language text. It also must appear within the *OpenSearchDescription* tags. The *Description* element will look something like the following:

```
<Description>Use the <name_of_Web_site> search engine
to retrieve results from this Web.</Description>
```

URL Element

The URL element points to the URL address that clients normally use to enter search queries for the remote index that was generated by the remote search engine. The *URL* element must appear within the *OpenSearchDescription* tags and will have several required and optional attributes. Because OpenSearch supports both index-based and page-based search engines, you'll find that two of the attributes help you define offset numbers for the result sets that these two basic search engine types return. The four attributes are as follows:

- **Template** This is the actual URL template that will be processed by the remote search engine. This attribute is required as part of the URL element.
- **Type** This attribute specifies the Multipurpose Internet Mail Extension (MIME) type of the search result format. This attribute is also required, and the values entered must be valid MIME types.
- **Indexoffset** By default, the content items in the result set are set to start with the first content item or the integer 1. This number can be changed if needed. The offset means that, from the first content item in the result set, you can start with the first one or you can start the result set with another content item in the list. By default, the offset is set to 1, so you don't need to include this attribute unless you want to set the offset to something other than 1.
- **Pageoffset** This attribute defines which page of the result set you want displayed first. By default, the first page of the result set will be displayed. If you need to start the result set on a page other than the first page, set this attribute to a number other than 1. In Search Server 2008, you'll use this attribute most often for the next results URL when creating an FLD.

Examples of the URL element query template are as follows:

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

Note When crafting the XML or the FLD file in Search Server 2008, you must capitalize the T in searchTerms. If you enter searchterms, the query template will fail because the OpenSearch standard is case sensitive, as is XML.

If you're unfamiliar with MIME, you can think of MIME types as an Internet standard describing different types of content that can exist on the Internet. Many of these MIME types have subtypes, which are also defined in the RFC documents. To learn more about MIME, see RFC 4288 and RFC 4289 as well as the Internet Assigned Numbers Authority (IANA) Web site about MIME types at . The potential MIME types (in theory) that can be included are audio, video, example, image, message, model, multipart, and text. But for our discussion about building FLDs, you'll likely use the Application MIME type and two of its subtypes—atom+xml and rss+xml—for nearly every FLD that you create.

The Application MIME type is explained in RFC 2046, which outlines the standards for the Application MIME type. According to this RFC, "The application media type is to be used for discrete data which do not fit in any of the other categories, and particularly for data to be processed by some type of application program. This is information which must be processed by an application before it is viewable or usable by a user." The data that is transferred between the FLD file during the execution of a query and the result set that is returned fits this definition well because it is not usable until the query is executed and the result set—returned in XML—is rendered using the built-in XSL in the Federated Results Web part. Therefore, you will want to use the Application MIME type, which is why the XML will read <ULR type=application/>.

You'll also specify the subtype of content that will be returned from the remote index. When you work with Search Server 2008, your choices are atom+xml or rss+xml because Search Server 2008 can't render HTML or XHTML in the Federated Results Web part. You'll enter both the MIME type (usually Application) and its

subtype into the URL element. Both the MIME type and its subtype are required as part of the URL element.

Contact Element

The *Contact* element will specify the e-mail address for the person who maintains the Description Document. The e-mail address must conform to the <alias>@<domain> convention that is specified in RFC 2822. This element is not required in the Description Document, but if it does appear, it can appear only once. The XML tag would look like the following:

```
<Contact>admin@contoso.msft</Contact>
```

Tags Element

The *Tags* element contains one or more words that are used as keywords to identify and categorize the search content. These words must be single words (no phrases are allowed) and are separated by a simple character space. This element is not required in the Description Document, but if it does appear, it can appear only once. The XML tag would look like the following:

```
<Tags>clothing boots</Tags>
```

Image Element

If you want to associate an image with the search content, you'll want to use the Image element in the Description Document. The value entered must be a Uniform Resource Indicator (URI), which is described in RFC 3986.

Note URIs are a standard way of identifying resources, both real and logical, on the Internet. URLs are similar to URIs in that, while they specify the location of the resource, they also include the method of finding that resource. URIs often include the method of finding the resource, but are not required to do so.

The *Image* element is not required as part of the Description Document, and it can appear from zero to n number of times. It must appear within the *OpenSearchDescription* tags.

There are three attributes that accompany the *Image* element, and they are as follows:

- **Height** Specifies the height of the image and must be a non-negative integer.
- **Width** Specifies the width of the image and must be a non-negative integer.
- **Type** Specifies the MIME Image subtype and must be a valid MIME type. Possible subtypes include common formats like jpeg, bmp, png, gif, or tiff. There are also vendor-specific types identified with the beginning vnd suffix, such as vnd.microsoft.icon.

A common practice is to use the 16 ×16 size for the image type "image/x-icon" or "image/vnd.microsoft.icon" and a 64 × 64 image for the image type "image/jpeg" or "image/png." Examples of the XML for this attribute are as follows:

```
<Image height="16" width="16" type="image/x-
icon">http://contoso.msft/icon_name.ico</Image>
```

```
<Image height="64" width="64" type="image/png">http://contoso.msft/image_name.png</Image>
```

Query Element

The *Query* element is the template or actual query that will be performed against the remote index. You can specify a specific search request or define a variable to host user-defined keyword query terms.

The *Query* element should also provide at least one 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 re-create the current search.

The following attributes for the Query element are important to understand:

- **role** Identifies how the search client should interpret the search request. This is the only attribute that is required for this element, and the specified role values are as follows:
    - **Request** Represents a search query that retrieves the same set of search results
    - **Example** Represents a search query to demonstrate the search engine
    - **Related** Represents a search query that represents similar, but different, results
    - **Correction** Represents a search query that improves the result set, such as correcting misspelled words
    - **Subset** Represents a search query that narrows an existing set of results
    - **Superset** Represents a search query that expands an existing set of results
- **title** Contains the title or name of the search query that is no longer than 256 characters. This attribute cannot contain HTML or other markup language and is optional, so it is not required as part of the Query element.
- **totalResults** Contains the expected number of content items in the result set that will be returned from the remote index engine. This value is optional and, if configured, must contain a non-negative integer.
- **searchTerms** This attribute is replaced with the keyword values that you want to execute against the remote index. The configuration of this value is optional; however, you'll use this attribute often when you build FLD files in Search Server 2008.
- **count** This attribute is replaced by the number of search results per page that is indicated by the search client. This is an optional attribute.
- **startIndex** This attribute is replaced by the number of the first search results specified by the client's search query. This is an optional attribute.
- **startPage** This attribute is replaced with the page number of the result set that is specified by the client's search query. This is an optional attribute.
- **language** This attribute is replaced by the name of the language in which the client wants to view the search results (see the "Language Element" section later in this chapter for more information). This is an optional attribute.
- **inputEncoding** This attribute is replaced by the specific character encoding that the client specifies in the query (see the "*InputEncoding* Element" section later in this chapter for more information). This is an optional attribute.
- **outputEncoding** This attribute is replaced by the specific character encoding for the result set (see the "*OutputEncoding* Element" section later in this chapter for more information). This is an optional attribute.

An example of the XML would be the following:

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

The *Query* element can be extended with additional attributes if those additional attributes are associated

with a namespace. The extended attribute will have a corresponding template parameter with the same name within the specified namespace.

Developer Element

The *Developer* element is similar to the Contact element in that it is used only to identify the person who created or maintains the Description Document. This element is optional and may contain up to 64 characters that are non-HTML or non-markup text. An example of this element would be following:

```
<Developer>Bill English bill@contoso.msft</Developer>
```

Attribution Element

The *Attribution* element is used to credit the right sources for the result set content. This element can have up to 256 plain-text characters and may appear either zero or one time in the Description Document. The XML would look like the following:

```
<Attribution>Copyright contoso.msft</Attribution>
```

SyndicationRight Element

The *SyndicationRight* element is included to indicate the degree to which the search engine's results or contents can be queried or redistributed. The default value is "open," and this element is not required in the Description Document.

The attributes for this element and their meanings are explained in Table 14-1.

Table 14-1 Explanation of the *SyndicationRight* Element Attributes

	The search client may request search results	The search client may display the results to end-users	The search client may send the results to other search clients
Open	Yes	Yes	Yes
Limited	Yes	Yes	No
Private	Yes	No	No
Closed	No	No	No

The following is what the XML would look like in the Description Document:

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

AdultContent Element

The *AdultContent* element indicates whether the content should be displayed only to adults. There are no industry standards for what constitutes adult material, so the search engine manager must indicate if the material is adult oriented. The value for this element is Boolean, so the values of *FALSE*, *NO*, *0*, and *no* will all be accepted for the Boolean FALSE. All other character strings will be considered TRUE.

This element is not required in the Description Document, but if it does appear, it should appear only once. The XML for this element would look like the following:

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

Language Element

The *Language* element indicates which languages the search engine supports. A value of "*" indicates that the search engine does not restrict results to any particular language. This is the default setting. The language tags must conform to the specifications for RFC 3066. The following are two examples of the XML code:

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

<Language>*</Language>
```

InputEncoding Element

The *InputEncoding* element indicates the character set that will be used for entering queries into the search engine. This element is also considered an attribute of the *Query* element, described earlier. The input of this element must conform to the XML 1.0 Character Encodings that are specified by the IANA Character Set Assignments. The default is UTF-8. This element is not required in the Description Document, but may appear from zero to *n* times.

An example of the XML would be the following:

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

More Info The sources for the XML 1.0 Character Encodings can be found at . You can find the IANA specifications for the Character Set Assignments at .

OutputEncoding Element

The *OutputEncoding* element specifies the character set with which the result set will be encoded. The default is UTF-8. The character sets must conform to the XML 1.0 Character Encodings that are specified by the IANA Character Set Assignments. This element is not required in the Description Document, but may appear from zero to n number of times.

An example of the XML would be the following:

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

AutoDiscovery of RSS/Atom

RSS and Atom are the only supported formats in Search Server 2008 for result sets. If the remote search index

isn’t capable of returning the results in either RSS or Atom, then you’ll need to write a custom connector page that will convert the HTML/XHTML results into RSS or Atom in order to display the results in the Federated Results Web part. Some restrictions will apply when you use this element in the Description Document.

First, the "type" attribute must contain the value "application/opensearchdescription+xml." Second, the "rel" attribute must contain the value "search." Third, the "href" attribute must contain a URI that resolves to an OpenSearch Description Document. The "title" attribute may contain a plain-text string describing the search engine, but this is not required.

An example of the XML would look like the following:

```
<link rel="search"
         href="http://example.com/opensearchdescription.xml"
         type="application/opensearchdescription+xml"
         title="Content Search" />
```

OpenSearch Response Elements

In addition to the description elements, the OpenSearch standard also specifies some response elements, which we will briefly discuss. There are fewer response elements than description elements, and it is less likely that you’ll work with them directly. In addition, you’ll notice that the response elements are really just that: responses to the query elements found in the Description Document. We’ve pulled an example of some response XML directly from the OpenSearch 1.1 documentation and have represented it here for your consideration:

```
<?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/opensearchdescription.xml"/>
     <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>
```

As you read through this XML, you can see the responses to query elements that were entered into the Description Document (not illustrated here), such as *totalResults*, *startIndex*, *Query role*, *searchTerms*, and *title*.

Other response elements are part of the OpenSearch standard. The following sections outline them briefly.

totalResults Element

The *totalResults* element indicates the total number of results that will come back from the search engine. Interestingly, the standard indicates that if this element doesn't appear on the result page, then the user should consider that page to be the last page in the result set. The value returned must be a non-negative integer. The default number will equal the offset index number of the last content item on the current page. This element is not required on the result set page, but based on the standard itself, you may find its lack of requirement a bit confusing. The XML for this element would look like the following:

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

startIndex Element

The *startIndex* element is the number of the first content item in the result set. If this element doesn't appear, then the OpenSearch standard indicates that the current page should be considered as the first page in the result set. The value must be an integer, and its default value equals the *indexOffset* value in the Description Document. This element is not required in the result set. The XML would look like the following:

```
<startIndex>1</startIndex>
```

itemsPerPage Element

This element indicates the number of content items returned on each page of the result set. If this value is not set as one of the response elements, then the number of items that are returned on the first page of the result set will be considered the default number. If the value is set, the number must be a non-negative integer. The XML would look like the following:

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

< Back Next >