Metadata Frequently Asked Questions

General collection and metadata questions

Q: What metadata planning information will help me decide what to catalog?

A: Become familiar with the NSDL_DC Metadata Guidelines as well as the NSDL Collection Policy and th Resource Quality Checklist, and other metadata quality rubrics. Understand how to contribute to UCARConnect.

Various topics are appropriate for inclusion in the library. In terms of making your resources accessible within UCARConnect, you can use specified collection tools that will assist in the generation of metadata records and in the ability of sending metadata records to UCARConnect. The project will create metadata records that catalog each major product or section of your resource(s) with the following information: title, URL, description, resource creator, education level and subject keywords. Once these metadata records are ingested into UCARConnect, its accessible to users.

Q: We have many pages on our web site, and provide access to many documents. Applying metadata to all of these will mean that we will have to create extensive amounts of metadata records. Isn't that right?

A: Achieving NSDL_DC compliance does not depend on metadata for every single page on a web site. It is quite appropriate to create "higher level" metadata records that describe a collection of resources. Furthermore, records for discrete portions of a site (e.g., individual web pages, .pdf documents, or videos) can be created over time and contributed to the repository gradually.

Q: What do I do if I have existing metadata that is not NSDL_DC metadata?

A: You may choose a metadata standard that works best for your collections and your particular audience. However, for optimal discovery of your resources in UCARconnect, you will want to share NSDL_DC metadata with UCARConnect. At a minimum, UCARConnect can work with oai_dc metadata. You will need to crosswalk or map your metadata to the oai_dc and/or NSDL_DC metadata format. In essence, a crosswalk is a table that maps the relationships and equivalencies between two or more metadata standards. Best practice is to share crosswalked NSDL_DC metadata instead of your native metadata. We can assist you to actually make the crosswalk work and provide advice on the appropriate metadata fields to which your metadata may map.

Q: What technical format does NSDL_DC metadata need to be delivered in for UCARConnect?

A: XML documents that validate to the lar XML schema.

Q: What tools can be used for cataloging?

A: See the NSDL_DC Metadata Guidelines.

Q: We have submitted a collection to NSDL but would like to modify our collection record. What's the easiest way to do this?

A: At present the easiest way is to contact us to discuss the changes you suggest.

Q: How does UCARConnect handle subcollections?

A: Currently, UCARConnect accommodates one level of aggregation. This means that any collections and subcollections of items associated with a particular collection record will be exposed in the OAI provider as ONE set (collection), no matter how many we aggregated at the time of ingest.

Q: We'd like to send a logo/brand image for our collection. How should we supply that to you? What format and size do you prefer?

Send an image as an email attachment to us at UCARConnect. In the body of the message specify the height and width in pixels. See our Brand Image Guidelines for specific information on size and format requirements.

Q: How does collection branding work?

A: Brand images are associated with collection records. The brand displayed with an item in search results is derived from the collection record associated with the item. If you would like specific collections from your OAI provider branded separately, each collection must be a separate OAI set (so that we can harvest those items separately) and each collection will need its own UCARconnect collection record. You will also need to let us know which OAI set goes with which collection record. See more on branding.

Q: What is the process for having new or updated metadata records harvested? How frequently will these occur?

A: Generally, by default, collections are updated monthly unless we are informed otherwise. Collection builders may request a harvest if need be.

Q: The form on your site for submitting collection records only allows 8 keywords. Can we submit more keywords?

A: There is no limit on the number of keywords. If you would like to submit more than 8 keywords, contact us with the name of your collection and what you would like to add.

Q: Our repository makes heavy use of educational grade bands, what NSDL_DC element can be used for this information?

A: In the lar metadata framework, use the element <educationLevel> and one of the vocabulary choices for that field. See the lar framework information page for more information.

Q: Does the audienceRefinement element require a controlled vocabulary?

A: Yes. The choices for this field are "Educator", "Educator and learner" and "Learner". Please refer to the vocabulary choices for the <audienceRefinement> field as well as to the lar framework information page for more information.

Q: We have controlled vocabulary terms for the metadata fields of type and format. Is this a problem?

A. We're using the information in the Format and Type elements to allow users to limit search results, so it's very important to have controlled vocabulary terms that we can recognize in those fields. Please refer to the vocabularies for each field in the lar format: type; format.

Q: What is the correct way to express language in the metadata record?

A. The lar metadata format uses two-letter language codes--for a list, please refer to the vocabulary choices for the field.

Q: Why does the NSDL require a unique identifier for each metadata record? And how is the metadata record id different from the unique id in an identifier field?

A: The value in the identifier field pertains to the resource, not the metadata; we also need a unique identifier for the metadata itself. This is called the OAI identifier of a metadata record and allows harvesters, such as UCARConnect, to match reharvested metadata records with the correct metadata record in our repository. In other words, the unique id for your metadata record makes it possible for harvesters to update your metadata record. In fact, we recommend that collections use the oai-identifier format (http://www.openarchives.org/OAI/2.0/guidelines-oai-identifier.htm) to identify metadata records. Many harvesters, by default, may use the filename as the OAI identifier.

Warning: it is bad practice to have a date or time as part of the unique identifier for a metadata record. Including a date or time in the id makes each iteration of the same metadata record unique. This means harvesters cannot match new versions of metadata records with old ones, and eventually results in harvesters having a large number of duplicate (and stale!) records.

XML questions

Q: What is an XML schema?

A: An XML schema defines the structure of XML documents. They define what XML elements can be used in a document and how. They also specify the data type and number of occurrences for each XML element. For example, the NSDL_DC XML schema defines an element called dc:title that may or may not appear in the document but if it does, the content is text. Because XML schemas define the structure of XML elements in a document, this 'instance' document can be validated by machines to see if it is correctly structured and using the correct data types. For more information, see http://www.w3.org/XML/Schema, and in particular the XML schema primer: http://www.w3.org/TR/xmlschema-0/.

Q: What is an XML entity reference?

A: An XML entity reference is a way to give indirect references to characters for an XML parser. For example, an XML parser given:

<tag>I am explaining the <sample> element </tag>
will view the characters "<sample>" as an element, and complain that there is no closing tag. XML entities allow the above to be expressed in a way that makes sense to XML parsers:
<tag>I am explaining the <sample> element </tag>
The "<" and ">" characters are expressed as the XML entities of "<" and ">" respectively.

Q: Does XML require any special encoding of characters?

A: XML defines five entities that must be used whenever the characters are to be viewed as simply characters (e.g. when they are part of the value of an element): < must be <
> must be >
& must be &
' must be '
" must be "

Note that this is true even when the characters are part of a URL, such as: <dc:identifier>http://fake.example.com?arg1=5&arg2=8</dc:identifier>

Q: What does double encoding mean for XML? How does it happen?

A: Sometimes text is XML encoded by a program in more than one context. Since the ampersand character "&" is used for XML encoding, when XML is doubly encoded, we see things like "&" or ">" If we can, we try to correct these double encodings back to single encodings so that the user interface displays readable text.

Q: What is Unicode?

A: Unicode is a character set which defines virtually all characters from pre-existing character sets. (See http://www.unicode.org/). A character set defines a set of characters regardless of how these characters are represented (in a text file, in a computer). XML is based on Unicode.

Q: What is UTF-8? Are there other encodings for Unicode?

A: An encoding method describes a way to store character set information. UTF-8, UTF-16 and UTF-32 are all encoding schemes for Unicode. UTF-8 encodes 7-bit ASCII characters "as is" in one byte, but is 2-5 bytes for other Unicode characters. UTF-16 uses 2 bytes for most characters, with additional characters encoded using "surrogate blocks." UTF-32 uses 4 bytes for each Unicode character.

Q: Can UTF-8 characters be invalid in XML?

A: From the Xerces documentation (http://xml.apache.org/xerces2-j/faq-common.html#faq-2): "There are many Unicode characters that are not allowed in an XML document, according to the XML spec. Typical disallowed characters are control characters, even if you escape them using the Character Reference form: &#xxxx; . See the XML spec, sections 2.2 and 4.1 for details. If the parser is generating this error, it is very likely that there is a character in the file that you cannot see. You can generally use a UNIX command like "od -hc" to find it."

Q: Are there potential problems with URL encoding?

A: Strictly speaking, a URL (http://www.w3.org/Addressing/URL/Overview.html) must be a valid URI (http://www.ietf.org/rfc/rfc2396.txt and http://www.ietf.org/rfc/rfc2732.txt). URIs require that non-ASCII characters and some ASCII characters be escaped. In practice, this may require special treatment of certain characters, such as spaces, in URLs. URL special characters are escaped using "%xx" where x is a hex digit. For example, a space would be represented as "%20".<dc:identifier>http://fake.example.com?arg1=NSDL%20rules</dc:identifier>
We have also discovered that our XML schema validator, Xerces, requires a stricter adherence to the URI spec, such that underscores in domain names are not valid. Oddly, many web browsers and domain name services do allow the underscore and view it as a distinct character, even though that contradicts the URI spec.

Q: How do you define a "well-formed XML file"?

A: A well-formed XML file must have a single outer tag that contains all other XML tags (which can be nested) in the file. The nesting must be matched throughout the document. That is, each open tag must have a corresponding ending tag.

Q: Can the XML element tags have new lines before and after the data they contain?

A: Most XML programs do not consider whitespace BETWEEN tags significant, while whitespace WITHIN tags is considered significant. In general, getting rid of extra whitespace WITHIN your values is good practice. For ingest, NSDL rids extra whitespace in your values for *our* version of your metadata; we leave the values exactly "as is" for your *native* metadata as served out by us.

Q: What is an XML namespace, and why do I need one?

A: "One of the original intentions of the namespace specification was to solve the problem of name collision." (*XML and Java*, Second Edition, by Maruyama, et al., p.20) For example, in a business application's XML documents, the expectation is that the element <stock> will contain a tickertape symbol from the NYSE, while in recipe XML documents, the expectation is that the element <stock> will contain "chicken," "beef" or "vegetable" -- the base ingredient of a cooking broth. The namespace provides context, or scoping for elements.

Namespace declarations indicate the unique identifier for the namespace that disambiguates elements. Thus, the important requirements for a namespace URI are 1) uniqueness, and 2) it must be a valid URI. There is no requirement that a namespace URI be resolvable.

Q: Is there an easy way to ensure that my namespace URI is valid and unique?

A: If you use a URL for a domain you have control over as a namespace URI, then the URI will be valid and unique (provided the URL you choose is unique within your domain). For example, we might use "http://nsdl.org/zing/" as a namespace URI ... but we need to make sure we don't use that URL for anything else.

Q: How do I indicate a namespace in an XML document? What is a "namespace declaration?" What is a "namespace prefix?"

XML namespaces are expressed in XML with namespace declarations. A namespace declaration also indicates a prefix to be used in the XML document when scoping elements.

Here's an example:<oai_dc:dc xmlns:oai_dc=http://www.openarchives.org/OAI/2.0/oai_dc/
xmlns:dc="http://purl.org/dc/elements/1.1/" >
<dc:title>UCARConnect Metadata Guidelines</dc:title>

</oai_dc:dc>
There are two elements in the example. The outer element's qualified name is "oai_dc:dc" and the inner element's qualified name is "dc:title." The characters before the colon are the "namespace prefix"; the characters after the colon are the "local name" for the element. So the outer element has local name "dc" and is scoped to the namespace indicated by the namespace prefix "oai_dc", while the inner element has local name "title" and is scoped to the namespace indicated by the namespace prefix "dc." The two namespace prefixes are declared with the "xmlns:oai_dc" and "xmlns:dc" attributes of the outer element.

"xmlns" indicates an XML namespace declaration. The characters after "xmlns:" are the namespace prefix to be used in the XML document. The value of the attribute is the namespace URI. Note that the oai_dc namespace URI is a URL, but does not resolve to any HTML if it is entered into a web browser. However, the oai_dc namespace URI is a URL in the domain of the Open Archives Initiative organization, which provides the definition for the XML elements in the namespace. The namespace URI for the "dc" namespace prefix is similarly defined with the "xmlns:dc" attribute.

In the following example, the namespace of the outer element is the default namespace. The default namespace URI is defined with the "xmlns" attribute. The important points are that there is a namespace defined for the outer element, and it uses a null namespace prefix.<dc xmlns=http://www.openarchives.org/OAI/2.0/oai_dc/
xmlns:fred="http://purl.org/dc/elements/1.1/" >
<fred:title>UCARConnect Metadata Guidelines</dc:title>

</dc>
Note that the two examples above are semantically the same: they have the same elements with the same contents in the same namespaces. The fact that the namespace prefixes used are different is **NOT**considered to be a semantic difference in XML. Note that there is no XML schema indicated for either of the two examples above. For more information on XML namespaces, see http://www.w3.org/TR/REC-xml-names/or http://www.jclark.com/xml/xmlns.htm or any number of web sites or books that have information on this topic.

You must have a namespace declaration for each namespace prefix you use in your XML, including the default namespace prefix:<outer xmlns="fake.ns" xmlns:dc="http://purl.org/dc/elements/1.1/">
<Record>
<dc:title>Sample title.</dc:title>
</Record>

</outer>**Q: Are XML namespaces always required?**

A: It is good practice to use XML namespaces in nearly all cases, as they disambiguate local names. Namespaces are required in the Open Archives Initiative Protocol for Metadata Harvesting (http://www.openarchives.org/OAI/openarchivesprotocol.html).

Q: What are the correct namespace URIs for NSDL_DC?

A: Several namespace URIs are in an NSDL_DC metadata record. They are:

- ***nsdl_dc*** - http://ns.nsdl.org/nsdl_dc_v1.02/ for the entire NSDL_DC metadata record
- ***dc*** - http://purl.org/dc/elements/1.1/ for the 15 elements in Simple Dublin Core
- ***dct*** - http://purl.org/dc/terms/ for the Qualified Dublin Core elements
- ***ieee*** - http://www.ieee.org/xsd/LOMv1p0 for the IEEE-LOM elements
- ***xsi*** - http://www.w3.org/2001/XMLSchema-instance for the XML schema itself

See the sample XML record for the namespaces.

Q: What XML schemas does NSDL_DC use?

A: UCARconnect develops and maintains its own XML schemas for the NSDL_DC metadata framework. The NSDL_DC schemas can be thought of as a flavor of Qualified Dublin Core metadata with additional elements specific to UCARConnect needs. The NSDL_DC metadata format is currently at version 1.02.020. The XML schema that supports this version is: http://ns.nsdl.org/schemas/nsdl_dc/nsdl_dc_v1.02.xsd. You will see that this NSDL_DC XML schema is really a cascade of XML schemas. This overarching schema essentially calls (think XML schema 'include' or 'import' statements) all other schemas that define various parts of the NSDL_DC metadata record. An example NSDL_DC metadata record is linked at http://ns.nsdl.org/schemas/nsdl_dc/nsdl_dc1_v1.02.xml.

Q: Is there a single XML schema for a single namespace?

No, there can be any number of XML schemas for a single namespace. Schema designers are encouraged to use a namespace that is associated with a domain they control. This encourages the organized development, reuse, and maintenance of XML schemas.

Q: Why does UCARConnect favor XML schemas over DTDs?

A: UCARConnect uses XML schemas because the Open Archives Initiative Protocol for Metadata Harvesting (http://www.openarchives.org/OAI/openarchivesprotocol.html) requires served metadata to be valid according to XML schemas.

Q: How do I indicate which XML schema my XML document should use to validate?

A: The location for an XML schema is indicated with the "schemaLocation" attribute, which resides in the schema instance namespace. See http://www.w3.org/TR/xmlschema-0/#ref40. It is conventional to use "xsi" as the namespace prefix for the schema instance namespace, so the qualified name of the attribute is "xsi:schemaLocation". Don't forget to properly declare the namespace for the "xsi" prefix with a namespace declaration: xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance".

XML schemas are actual documents, not abstract concepts (such as namespace URLs), so XML schema locations are indicated with URLs which resolve to an actual XML schema document.

Therefore, the "xsi:schemaLocation" attribute must be provided and its value should be the namespace URI followed by a blank, followed by the URL for the appropriate XML schema for the indicated namespace. So the location of the XML schema for the namespace "http://ucarconnect.edu" would be indicated by an attribute xsi:schemaLocation="http://ucarconnect.edu/ http://ucarconnect.edu/schemas/zing.xsd"

Q: What XML schema validation tools **exist?**

A: Many tools exist. For a list, see http://www.w3.org/XML/Schema#Tools.

Q: What XML schema validation tools does UCARConnect recommend?

UCARConnect does not specifically recommend a particular validation tool because each project has their own needs and issues.

Q: What XML schema validation tools does UCARConnect use?

UCARConnect uses Xerces from the Apache Software Foundation http://xml.apache.org/xerces2-j/index.html. All contributed metadata is validated using Xerces.

Q: What if I want to include additional terms specific to my collection in my metadata?

A: You may do so but for OAI-PMH to work, you will need to provide an XML schema that defines these additional terms or elements that are specific to your metadata. It is a good idea to put such additional terms or elements in a new namespace under the domain of your organization. For example, you could have:<mf:outer xmlns:mf="http://my.org/specialnamespace"> ... </mf:outer> and it would allow you to declare your own terms. See the W3C documentation on XML Namespaces for additional information: http://www.w3.org/TR/REC-xml-names/.

Q: Can we include our subjects exactly as they are stored in our database? If so, do we need to create our own version of your XML schema, or can we just use NSDL_DC as is?

A: To include your subject or keyword terms, use a separate dc:subject element for each word or concept. If you do this, you do not need your own schema; the NSDL_DC schema works.

Q.: Is there any provision in the NSDL_DC schema for providing a unique identifier for use in future UCARConnect imports of metadata?

A: The NSDL_DC XML schemas defines the elements and metadata fields for describing a resources, not for controlling UCARConnect imports or OAI harvests of metadata. UCARConnect requires a unique identifier, permanent id associated with the METADATA record so that we know to which record we must apply a harvested update, for example. The whole notion of metadata records and their identifiers is dealt with in the OAI Protocol for Metadata Harvesting: http://www.openarchives.org/OAI/openarchivesprotocol.html.

Q: I'm not sure what you mean by "all" the records being schema valid - in my mind they will either all be valid or all invalid.

A: Each metadata record you provide needs to be valid according to its XML schema. If a particular record has an element not allowed by the schema, then that particular record is not valid. If a particular record has a bad value according to the schema, then that record is not valid.

OAI questions

Q: What is "OAI"?

A: OAI stands for the Open Archives Initiative. From their FAQ, "the Open Archives Initiative develops and promotes interoperability standards ... to facilitate the efficient dissemination of content." Actually, this question is best answered by reading the answers to relevant question in their FAQ,http://www.openarchives.org/documents/FAQ.html

Q: What is "OAI-PMH"?

A: The Open Archives Initiative Protocol for Metadata Harvestinghttp://www.openarchives.org/OAI/openarchivesprotocol.html is a protocol that allows metadata to be shared in an interoperable manner.

Q: Are there any technologies I should be familiar with in order to understand OAI-PMH?

A: Yes. You should be conversant with the basics of HTTP, XML, XML namespaces, and XML schemas.

- ***HTTP***: understand how to send an xml response over HTTP, and the basics of GET and POST requests.
- ***XML***: understand the difference between "well-formed" and "valid," and the difference between HTML and XML.
- ***XML namespaces***: understand what they are, the difference between the namespace URI and the namespace prefix; understand the "default" namespace. Understand the difference between a namespace URI and a namespace URL.
- ***XML schemas***: understand what they are, how they can be used, how to indicate a particular schema for a particular namespace in your XML, and how to find out which schema is being used for a particular namespace in any XML.

Q: Are there any online OAI tutorials or other online OAI resources?

A: Yes. A good place to start might be http://www.oaforum.org/tutorial/ - OAI for Beginners - The Open Archives Forum online tutorial. Another good reference is Best Practices for OAI Data Provider Implementations and Shareable Metadata http://webservices.itcs.umich.edu/mediawiki/oaibp/index.php/Main_Page

Q: What metadata is allowed in the oai_dc format?

A: Metadata in the oai_dc format *must* adhere to the provided schema (a thttp://www.openarchives.org/OAI/2.0/oai_dc.xsd), which only allows the 15 Simple Dublin Core elements.

Q: When using OAI, are there official schemas I should use for Simple Dublin Core, that is oai_dc? What about other metadata formats?

A: OAI-PMH version 2.0 requires, at a minimum, that metadata be provided using their schema for Simple Dublin Core. The schema is at http://www.openarchives.org/OAI/2.0/oai_dc.xsd. The metadata format is called 'oai_dc'. OAI requires that you expose oai_dc as a base-level format for broad interoperability. The OAI-PMH protocol document is at: http://www.openarchives.org/OAI/openarchivesprotocol.html.

For any other metadata format besides oai_dc, the OAI-PMH protocol allows it to be provided as long as the metadata validates to its indicated XML schema. This means one can have their own metadata format defined by their own XML schemas and as long as the metadata validates against the schema, the metadata can be provided using OAI-PMH.

Q: May I provide metadata in a format other than oai_dc?

A: OAI-PMH allows metadata to be served in additional formats, as long as you correctly provide XML schema information for each metadata format and your metadata validates to its indicated schema.

Q: Does OAI version 2.0 strictly limit itself to the fifteen traditional elements?

A: See the answers to the two previous questions.

Q: If I want my OAI repository to serve a metadata format other than oai_dc, what do I need to do?

A: OAI-PMH requires schemas for all metadata served, so you must indicate an XML schema for the metadata format, and each of your metadata records must validate to its indicated schema. This means the xsi:schemaLocation attribute must be provided for served metadata. OAI-PMH requires that this attribute be present on the "metadata" elements. We reiterate: your metadata *must* validate to an XML schema to adhere to OAI-PMH.

Q: Does there need to be an XML schema for every metadata format I serve?

A: Yes, for every metadata format served by an OAI repository, there must be a namespace URI and an XML schema. These are exposed via ListMetdataFormats. Your metadata must validate to an XML schema to be compliant with the OAI-PMH protocol.

Q: Does every record I serve need to validate to an XML schema?

A: Every OAI response, and hence every OAI metadata record, *must* validate to an XML schema. Each metadata record you provide needs to be valid according to its XML schema. If a particular record has an element not allowed by the XML schema for its metadata, then that particular record is not valid. If a particular record has a bad value according to the XML schema for its metadata, then that record is not valid.

Note that multiple XML schemas are used in a single OAI response: OAI-PMH schema for the OAI response elements, a schema for the metadata part of a record, and schemas for any "abouts" associated with the record.

Q: Is there a distinction between a "namespace prefix" in XML and a "metadataPrefix" in OAI-PMH?

A: Yes. A "namespace prefix" in XML hooks a qualified name with the appropriate namespace declaration and therefore the URI for the namespace scoping the name. For example, <oai_dc:dc> has "oai_dc" as a namespace prefix, which indicates "dc" is scoped to the namespace URI indicated in the namespace declaration "xmlns:oai_dc=..." on the nearest ancestor element.

A "metadataPrefix" in OAI-PMH is the string used to uniquely identify a particular metadata format for an OAI repository. "metadataPrefix" is a required argument for ListRecords, GetRecord and ListIdentifiers requests. An OAI repository's mappings from metadataPrefixes to metadata namespace URIs and their XML schemas are exposed via ListMetadataFormats. See http://www.openarchives.org/OAI/openarchivesprotocol.html for more information on OAI-PMH metadataPrefixes. While the OAI-PMH reserves "oai_dc" as a *metadataPrefix*, no XML namespace prefixes are dictated in OAI-PMH. In fact, the OAI-PMH *metadataPrefix* and the XML*namespace prefix* in the OAI response may differ or be the same.

Q: Does my OAI server need to accommodate both HTTP "GET" and "POST" requests?

A: The OAI protocol states that "Repositories must support both the GET and POST methods." Any web server usually accommodates both types of requests. OAI servers (the programs you run on your web server that respond to OAI requests) must have the same responses to both GET and POST requests.

To test "POST" requests, you just need an HTML form that does one. See http://services.nsdl.org:8080/nsdloai/listRecords.html and look at the HTML source. Create a copy of the HTML source, change the <form action="blah"> so that "blah" is your OAI server's baseURL, and you'll be able to test the POST request for ListRecords. As for implementing POST, it depends on the flavor of your web server, the language of your program and how it ties to your web server.

Q: What is the baseURL of my OAI server?

A: The baseURL is the URL for an OAI server, without any arguments. The institution hosting the OAI server decides the baseURL. The same baseURL is used for all OAI requests to a particular server; only the arguments (after the "?") change for different OAI requests.

The baseURL is indicated in the <baseURL> element in the OAI Identify response, and it is also indicated as the value of the <request> element in all OAI responses. The indicated baseURL in either of these places should not include any arguments -- it should be up to but not including the question mark.

Q: What should my OAI server put in the <request> element in an OAI-PMH response?

A: The value of the <request> element must be the baseURL of your OAI server. There should also be an attribute-value pair for each argument in the (valid) OAI-PMH request.

Q: We want to use OAI identifiers - do we need a registered domain name?

A: Quoting from the official document at http://www.openarchives.org/OAI/2.0/guidelines-oai-identifier.htm: "Organizations must choose namespace-identifier values which correspond to a domain-name that they have registered, and are committed to maintaining. Note that since the oai-identifier is case-sensitive, a particular capitalization style must be selected and used consistently. A single domain name should not be used with variant capitalization. Domain name registration is used to avoid the need for any additional registration service for oai-identifiers. Domain name based identifiers guarantee global uniqueness without the need for OAI registration."

Q: What is the definition of an invalid OAI identifier, and how should my OAI server respond to invalid identifiers when they are passed as arguments?

A: OAI-PMH says that identifiers must be URIs. Thus, your OAI server should indicate an invalid argument (badArgument) error when an identifier value is not a valid URI. For example, an OAI request with an identifier argument containing a quotation (") should respond with an invalid argument (badArgument) error because the quotation character is not allowed in a URI.

Q: If a resumptionToken is given, then should it be the ONLY argument to appear with a verb? And for that matter, do you even need the verb?

A: As OAI-PMH indicates, resumptionToken is an exclusive argument -- it should be the only argument, besides verb. The OAI verb argument is always required; verbs that allow resumptionTokens can either have the exclusive resumptionToken argument or other arguments allowed for the verb. For example, it is incorrect for an OAI server to respond to a "ListRecords" verb that has both a "resumptionToken" and a "metadataPrefix" argument.

Note that OAI-PMH specifies that if resumptionTokens are used, the last response must contain an empty resumptionToken element.

Q: What is a reasonable number of records to serve in a single ListRecords, ListIdentifiers, or ListSets response?

A: Ideally, your responses will be sized to work smoothly with your server and as HTTP responses. The ideal number to serve is also dependent on the size of your records. Very large responses tend to get bogged down in the network or may slow up your server; very small responses require many HTTP requests/responses for a complete harvest. Generally speaking, a response size between 1/2 and 2 megabytes is reasonable. Usually this is somewhere between 100 and 1500 metadata records. This number could be larger for ListIdentifiers and ListSets, as "headers" and "sets" tend to be smaller than "records."

Q: What can go in OAI record headers? Where can we put additional information about our metadata?

A: OAI-PMH specifies that there must be an <identifier> element and a <datestamp> element, and that there may also be one or more <setSpec> elements. No other elements are allowed in the header.

If you would like to serve out additional information about the metadata (not the resource), then it probably belongs in an OAI about, "an optional and repeatable container to hold data about the metadata part of the record." (See http://www.openarchives.org/OAI/openarchivesprotocol.html#Record.) For example, rights information about the resource should be indicated in a <dc:rights> element in the metadata, but rights information about the metadata itself could be expressed in a <dc:rights> element in an OAI <about>. Note further that <about> information is "about the metadata" rather than "about your organization." The latter could be made available at your website or as a <description> specified in your OAI Identify response.

Q: How does the UCARConnect handle OAI sets?

A: UCARConnect can handle sets in terms of harvesting and providing. When UCARConnect uses OAI to harvest a collection, as many sets as necessary can be used to specify a collection. That is, a collection can be created from multiple sets if need be. When UCARconnect uses OAI to provide collections, we represent each collection as a separate set, no matter how many sets we harvested initially to create the collection.

Q: Is there a way to show where a metadata record came from in OAI, apart from analyzing OAI identifiers?

A: There is a "provenance" container that can be included in an OAI "about" which will clearly indicate from where the metadata record came. See http://www.openarchives.org/OAI/2.0/guidelines-provenance.htm. OAI identifiers, especially those that follow the recommended OAI-identifer format (http://www.openarchives.org/OAI/2.0/guidelines-oai-identifier.htm) may be analyzable, but they may not be analyzable, as OAI identifiers do not need to adhere to this format.

Q: Are there contexts where an OAI harvester would use a GetRecord request? Why is that request useful, since there is a ListRecords request?

A: OAI servers must correctly implement all six of the OAI-PMH verbs. An OAI harvester may be performing some spot checking or other testing if it uses a GetRecord request, or it might determine record updates with the ListIdentifiers response and use GetRecord to retrieve a subset of records. There is no requirement (other than common sense) that OAI harvesters must work in a particular way - correct use of OAI-PMH is all that is required.

Q: How can I check my OAI server for validation?

A: See the NSDL_DC Metadata Guidelines.