The Open Archives Initiative Protocol for Metadata Harvesting

Protocol Version 2.0 of 2002-06-14
Document Version 2015-01-08
http://www.openarchives.org/OAI/2.0/openarchivesprotocol.htm

Previous protocol version: Protocol Version 1.1 of 2001-07-02
Instructions for migrating from Version 1.1 to 2.0
Implementation Guidelines

Editors

The OAI Executive:
Carl Lagoze <lagoze@cs.cornell.edu > -- Cornell University - Computer Science
Herbert Van de Sompel <herbertv@lanl.gov > -- Los Alamos National Laboratory - Research Library

From the OAI Technical Committee:
Michael Nelson <m.l.nelson@larc.nasa.gov > -- NASA - Langley Research Center
Simeon Warner <simeon@cs.cornell.edu > -- Cornell University - Computer Science

Table of Contents

1. Introduction

The Open Archives Initiative Protocol for Metadata Harvesting (referred to as the OAI-PMH in the remainder of this document) provides an application-independent interoperability framework based on *metadata harvesting*. There are two classes of participants in the OAI-PMH framework:

- *Data Providers* administer systems that support the OAI-PMH as a means of exposing metadata; and
- *Service Providers* use metadata harvested via the OAI-PMH as a basis for building value-added services.

In this document the key words "**must**", "**must not**", " **required**", "**shall**", "**shall not**", "**should**", " **should not**", "**recommended**", "**may**", and "**optional** " in **bold face** are to be interpreted as described in RFC 2119 . An implementation is not conformant if it fails to satisfy one or more of the "**must**" or "**required**" level requirements for the protocols it implements.

This document refers in several places to "community-specific" practices to which individual protocol implementations **may** conform. These practices are described in an accompanying Implementation Guidelines document.

2. Definitions and Concepts

2.1 Harvester

A *harvester* is a client application that issues OAI-PMH requests. A harvester is operated by a service provider as a means of collecting metadata from repositories.

2.2 Repository

A *repository* is a network accessible server that can process the 6 OAI-PMH requests in the manner described in this document. A repository is managed by a data provider to expose metadata to harvesters. To allow various repository configurations, the OAI-PMH distinguishes between three distinct entities related to the metadata made accessible by the OAI-PMH.

- *resource* - A resource is the object or "stuff" that metadata is "about". The nature of a resource, whether it is physical or digital, or whether it is stored in the repository or is a constituent of another database, is outside the scope of the OAI-PMH.
- *item* - An item is a constituent of a repository from which metadata about a resource can be disseminated. That metadata may be disseminated on-the-fly from the associated resource, cross-walked from some canonical form, actually stored in the repository, etc.
- *record* - A record is metadata in a specific metadata format. A record is returned as an XML-encoded byte stream in response to a protocol request to disseminate a specific metadata format from a constituent item.

2.3 Item

An *item* is a constituent of a repository from which metadata about a resource can be disseminated. An item is conceptually a container that stores or dynamically generates metadata about a single resource in multiple formats, each of which can be harvested as records via the OAI-PMH. Each item has an identifier that is unique within the scope of the repository of which it is a constituent.

2.4 Unique Identifier

A *unique identifier* unambiguously identifies an item within a repository; the unique identifier is used in OAI-PMH requests for extracting metadata from the item. Items **may** contain metadata in multiple formats. The unique identifier maps to the item, and all possible records available from a single item share the same unique identifier.

The format of the unique identifier **must** correspond to that of the URI (Uniform Resource Identifier) syntax. Individual communities **may** develop community-specific URI schemes for coordinated use across repositories. The scheme component of the unique identifiers **must not** correspond to that of a recognized URI scheme unless the identifiers conform to that scheme. Repositories **may** implement the `oai-identifier` syntax described in the accompanying Implementation Guidelines document.

Unique identifiers play two roles in the protocol:

1. *Response:* Identifiers are returned by both the `ListIdentifiers` and `ListRecords` requests.
2. *Request:* An identifier, in combination with a `metadataPrefix`, is used in the `GetRecord` request as a means of requesting a record in a specific metadata format from an item.

Note that the identifier described here is *not* that of a *resource*. The nature of a resource identifier is outside the scope of the OAI-PMH. To facilitate access to the resource associated with harvested metadata, repositories **should** use an element in metadata records to establish a linkage between the record (and the identifier of its item) and the identifier (URL, URN, DOI, etc.) of the associated resource. The mandatory Dublin Core format provides the `identifier` element that **should** be used for this purpose.

2.5 Record

A record is metadata expressed in a single format. A record is returned in an XML-encoded byte stream in response to an OAI-PMH request for metadata from an item. A record is identified unambiguously by the combination of the unique identifier of the item from which the record is available, the `metadataPrefix` identifying the metadata format of the record, and the datestamp of the record. The XML-encoding of records is organized into the following parts:

- *header* -- contains the unique identifier of the item and properties necessary for selective harvesting. The header consists of the following parts:
  - the unique identifier -- the unique identifier of an item in a repository;
  - the datestamp -- the date of creation, modification or deletion of the record for the purpose of selective harvesting.
  - zero or more setSpec elements -- the set membership of the item for the purpose of selective harvesting.
  - an **optional** `status` attribute with a value of `deleted` indicates the withdrawal of availability of the specified metadata format for the item, dependent on the repository support for deletions.
- *metadata* -- a single manifestation of the metadata from an item. The OAI-PMH supports items with multiple manifestations (formats) of metadata. At a minimum, repositories **must** be able to return records with metadata expressed in the Dublin Core format, without any qualification. Optionally, a repository **may** also disseminate other formats of metadata. The specific metadata format of the record to be disseminated is specified by means of an argument -- the `metadataPrefix` -- in the `GetRecord` or `ListRecords` request that produces the record. The `ListMetadataFormats` request returns the list of all metadata formats available from a repository, or for a specific item (which can be specified as an argument to the `ListMetadataFormats` request).

- *about* -- an **optional** and repeatable container to hold data about the metadata part of the record. The contents of an about container **must** conform to an XML Schema. Individual implementation communities **may** create XML Schema that define specific uses for the contents of about containers. Two common uses of about containers are:
  - *rights statements:* some repositories may find it desirable to attach terms of use to the metadata they make available through the OAI-PMH. No specific set of XML tags for rights expression is defined by OAI-PMH, but the about container is provided to allow for encapsulating community-defined rights tags.
  - *provenance statements:* One suggested use of the about container is to indicate the provenance of a metadata record, e.g. whether it has been harvested itself and if so from which repository, and when. An XML Schema for such a provenance container, as well as some supporting information is available from the accompanying Implementation Guidelines document.

The following example shows an XML-encoding of a record and its components:

- the *header* part with:
  - a unique identifier of the item from which the record was disseminated, equal to oai:arXiv.org:cs/0112017;
  - the datestamp of the record equal to 2002-02-28;
  - two setSpecs, respectively `cs` and `math`, indicating that the item from which the record was disseminated belongs to two sets of the repository;
- the *metadata* part. This consists of a single root tag - in the example the tag `oai_dc:dc` - with the nested tags belonging to the corresponding metadata format - in the example, Dublin Core elements such as `dc:title`. Note that the root tag within the metadata part includes a number of attributes that are common to all XML documents that use namespaces and schema validity:
  - *namespace declarations* -- the declarations of the namespaces used within the metadata part, each of which is prefixed with `xmlns`. Namespace declarations within the metadata part fall into two categories:
    - *metadata format specific namespace(s)* - every metadata part **must** include one or more `xmlns` prefixed attributes that define the correspondence between a metadata format prefix -- e.g. `dc` -- and the namespace URI (as defined by the XML namespace specification) of the respective metadata format. Some metadata formats employ tags from multiple namespaces, requiring multiple `xmlns` prefixed attributes -- in the example, there are declarations for both `oai_dc` and `dc`.
    - *xml schema namespace* - every metadata part **must** include the attribute `xmlns:xsi`, the value of which **must** always be the URI shown in the example, which is the namespace URI for XML schema.
  - `xsi:schemaLocation` -- the value of which is a URI, URL pair; the first is the namespace URI (as defined by the XML namespace specification) of the metadata that follows in this part, and the second is the URL of the XML schema for validation of the metadata that follows.
- one *about* part of the record which uses the `oai_provenance.xsd` schema, described in the accompanying Implementation Guidelines document, as a means to provide information regarding the origins of the metadata part of the record. Note that the root element within each about part has the same structure as the root element in the metadata part.

```
<header>
  <identifier>oai:arXiv:cs/0112017</identifier>
  <datestamp>2002-02-28</datestamp>
  <setSpec>cs</setSpec>
  <setSpec>math</setSpec>
</header>
<metadata>
 <oai_dc:dc
     xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"
     xmlns:dc="http://purl.org/dc/elements/1.1/"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/
     http://www.openarchives.org/OAI/2.0/oai_dc.xsd">
   <dc:title>Using Structural Metadata to Localize Experience of Digital
             Content</dc:title>
   <dc:creator>Dushay, Naomi</dc:creator>
   <dc:subject>Digital Libraries</dc:subject>
```

```
   <dc:description>With the increasing technical sophistication of both
    information consumers and providers, there is increasing demand for
    more meaningful experiences of digital information. We present a
    framework that separates digital object experience, or rendering,
    from digital object storage and manipulation, so the
    rendering can be tailored to particular communities of users.
   </dc:description>
   <dc:description>Comment: 23 pages including 2 appendices,
                   8 figures</dc:description>
   <dc:date>2001-12-14</dc:date>
   <dc:type>e-print</dc:type>
   <dc:identifier>http://arXiv.org/abs/cs/0112017</dc:identifier>
 </oai_dc:dc>
</metadata>
<about>
  <provenance
      xmlns="http://www.openarchives.org/OAI/2.0/provenance"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/provenance
      http://www.openarchives.org/OAI/2.0/provenance.xsd">
    <originDescription harvestDate="2002-02-02T14:10:02Z" altered="true">
      <baseURL>http://the.oa.org</baseURL>
      <identifier>oai:r2:klik001</identifier>
      <datestamp>2002-01-01</datestamp>
      <metadataNamespace>http://www.openarchives.org/OAI/2.0/oai_dc/</metadataNamespace>
    </originDescription>
  </provenance>
</about>
```

2.5.1 Deleted records

If a record is no longer available then it is said to be *deleted*. Repositories **must** declare one of three levels of support for deleted records in the `deletedRecord` element of the Identify response:

- `no` - the repository does not maintain information about deletions. A repository that indicates this level of support **must not** reveal a deleted status in any response.
- `persistent` - the repository maintains information about deletions with no time limit. A repository that indicates this level of support **must** persistently keep track of the full history of deletions and consistently reveal the status of a deleted record over time.
- `transient` - the repository does not guarantee that a list of deletions is maintained persistently or consistently. A repository that indicates this level of support **may** reveal a deleted status for records.

If a repository does not keep track of deletions then such records will simply vanish from responses and there will be no way for a harvester to discover deletions through continued incremental harvesting. If a repository does keep track of deletions then the datestamp of the deleted record **must** be the date and time that it was deleted. Responses to `GetRecord` request for a deleted record **must** then include a `header` with the attribute `status="deleted"`, and **must not** include `metadata` or `about` parts. Similarly, responses to selective harvesting requests with set membership and date range criteria that include deleted records **must** include the headers of these records. Incremental harvesting will thus discover deletions from repositories that keep track of them.

Deleted status is a property of individual records. Like a normal record, a deleted record is identified by a unique identifier, a metadataPrefix and a datestamp. Other records, with different metadataPrefix but the same unique identifier, may remain available for the item.

2.6 Set

A *set* is an optional construct for grouping items for the purpose of selective harvesting. Repositories **may** organize items into sets. Set organization **may** be flat, i.e. a simple list, or hierarchical. Multiple hierarchies with distinct, independent top-level nodes are allowed. Hierarchical organization of sets is expressed in the syntax of the

`setSpec` parameter as described below. When a repository defines a set organization it **must** include set membership information in the headers of items returned in response to the `ListIdentifiers` , `ListRecords` and `GetRecord` requests.

Each node in a set organization of a repository has:

- a `setSpec` -- a colon [`:`] separated list indicating the path from the root of the set hierarchy to the respective node. Each element in the list is a string consisting of any valid URI *unreserved* characters, which **must not** contain any colons [`:`]. Since a `setSpec` forms a unique identifier for the set within the repository, it **must** be unique for each set. Flat set organizations have only sets with `setSpec` that do not contain any colons [`:`].
- a `setName` -- a short human-readable string naming the set.
- a `setDescription` -- an **optional** and repeatable container that **may** hold community-specific XML-encoded data about the set; the accompanying Implementation Guidelines document provides suggestions regarding the usage of this container.

The following is an example of a possible set hierarchy in a repository:

- Institutions
  - Oceanside University of Nebraska
  - Valley View University of Florida
- Subjects
  - Existential Kenesiology
  - Quantum Psychology

The following table shows a possible representation of the above set hierarchy by means of `setName` and respective `setSpec` values.

setName	setSpec
Institutions	institution
Oceanside University of Nebraska	institution:nebraska
Valley View University of Florida	institution:florida
Subjects	subject
Existential Kenesiology	subject:kenesiology
Quantum Psychology	subject:quantum

An item **may** be organized in one set, several sets, or no sets at all. In the example above, it is conceivable that an individual item is organized in both `subject` and `institution:florida`. A harvester **should not** assume that harvesting every set in a repository will retrieve metadata from all items in the repository. Items **may** also be assigned to interior nodes in the set hierarchy.

The actual meaning of a set or of the arrangement of sets in a repository is not defined in the protocol. It is expected that individual communities may formulate well-defined set configurations with perhaps a controlled vocabulary for `setNames` and `setSpec` , and may even develop mechanisms for exposing these to harvesters. For example, a group of cooperating e-print archives in a specific discipline may agree on sets that arrange metadata in their repositories based on a controlled subject classification.

A repository's set hierarchy is represented in the protocol via `setSpecs`. `ListSets` returns a list indicating the configuration of sets in a repository. Each member of this list **must** include a `setSpec` and a `setName` and **may** include

a `setDescription`. `ListRecords` and `ListIdentifiers` requests **may** include an **optional** `set` argument, the value of which is a `setSpec`, to specify the target set for selective harvesting. In the previous example of a set hierarchy, the `setSpec institution:nebraska` could be used in a request to return only those records that are disseminated from items organized in the set represented by this `setSpec`. Five issues should be noted here:

- If a repository supports sets then it **must** include set membership information in response to `ListIdentifiers`, `ListRecords` and `GetRecord` requests. The list of `setSpec` elements should include only the minimum number of `setSpec` elements required to specify the set membership. Using the previous example of a set hierarchy, the header for an item organized in set `institution:florida` should not include `setSpec institution` since that is implied by the `setSpec institution:florida`.
- An item **may** be organized in more than one set; meaning that different `setSpec` arguments **may** return the same record(s).
- An item need not be organized in any set; meaning that an exhaustive repetition of `ListRecords` requests with all possible `setSpecs` is not guaranteed to return all records in the repository. The only guaranteed methods of harvesting all records or headers are `ListRecords` or `ListIdentifiers` requests with no `setSpec` argument.
- When a `setSpec` is used as an argument, the response **must** include records or headers from all items in the set specified by the `setSpec` , and all records or headers from items in sets that are descendant from the specified set. Using the previous example of a set hierarchy, a `setSpec` of `institution` to the `ListRecords` request will return all records from metadata organized within the set with a `setSpec` value equal to `institution` and within the descendent sets with setSpec values equal to `institution:florida` and `institution:nebraska`.
- The set hierarchy of a repository may include sets that are empty.

2.7 Selective Harvesting

Selective harvesting **allows harvesters to limit harvest requests to portions of the metadata available from a repository.** The OAI-PMH supports selective harvesting with two types of harvesting criteria that **may** be combined in an OAI-PMH request: datestamps and set membership.

2.7.1 Selective Harvesting and Datestamps

Harvesters **may** use *datestamps* to harvest only those records that were created, deleted, or modified within a specified date range. To specify datestamp-based selective harvesting, datestamps are included as values of the **optional** arguments, `from` and `until`, in the `ListRecords` and `ListIdentifiers` requests. Harvesting is restricted to the range specified by the `from` and `until` arguments, extending back to the earliest datestamp if `from` is omitted, and forward to the most recent datestamp if `until` is omitted. Range limits are *inclusive:* `from` specifies a bound that **must** be interpreted as "greater than or equal to", `until` specifies a bound that **must** be interpreted as "less than or equal to". Therefore, the `from` argument **must** be less than or equal to the `until` argument. Otherwise, a repository must issue a `badArgument` error.

Repositories **must** support selective harvesting with the `from` and `until` arguments expressed at day granularity. **Optional** support for seconds granularity is indicated in the response to the `Identify` request. The value of datestamps in both requests and responses **must** comply to the specifications for UTCdatetime in this document. A repository **must** update the datestamp of a record if a change occurs, the result of which would be a change to the metadata part of the XML-encoding of the record. Such changes include, but are not limited to, changes to the metadata of the record, changes to the metadata format of the record, introduction of a new metadata format, termination of support for a metadata format, etc.

Datestamp ranges for selective harvesting are expressed in the `from` and `until` arguments that **may** be submitted in the `ListRecords` and `ListIdentifiers` requests. Repositories **must** use the following rules to create a `ListRecords` response matching the specified datestamp range according to the type of change that occurred within the repository. The response to a `ListIdentifiers` request follows the same rules but is abbreviated to include only headers rather than records.

- *modification* - the response **must** include records, corresponding to the `metadataPrefix` argument, which have changed within the bounds of the `from` and `until` arguments.

- *creation* - the response **must** include records, corresponding to the `metadataPrefix` argument, that have become available from the repository within the bounds of the `from` and `until` arguments.
- *deletion* - depending on the level at which a repository keeps track of deleted records, the response **may** include headers of records, corresponding to the `metadataPrefix` argument, which have been withdrawn from the repository within the bounds of the `from` and `until` arguments. Deleted status is indicated via the status attribute of the header element and no metadata is included.

Every header returned by the `GetRecord`, `ListRecords` or `ListIdentifiers` requests contains a datestamp, which reflects the most recent date and time of the creation, modification, or deletion according to the rules defined above.

2.7.2 Selective Harvesting and Sets

Harvesters may specify set membership as a criteria for selective harvesting. To specify set-based selective harvesting, a `setSpec` is included as the value of the **optional** `set` argument to the `ListRecords` and `ListIdentifiers` requests, thereby specifying selective harvesting of records from items within the respective set.

When a `setSpec` is used as an argument, the response **must** include:

- the records corresponding to the `metadataPrefix` argument, or headers thereof in the case of deleted records, available from those items in the set specified by the `setSpec;`
- the records corresponding to the `metadataPrefix` argument, or headers thereof in the case of deleted records, available from those items in sets that are descendant from the specified set.

3. Protocol Features

3.1 HTTP Embedding of OAI-PMH requests

OAI-PMH requests are expressed as HTTP requests. A typical implementation uses a standard Web server that is configured to dispatch OAI-PMH requests to the software handling these requests. The remainder of this section describes the aspects of the protocol that are specific to the HTTP embedding.

3.1.1 HTTP Request Format

OAI-PMH requests **must** be submitted using either the HTTP `GET` or `POST` methods. `POST` has the advantage of imposing no limitations on the length of arguments. Repositories **must** support both the `GET` and `POST` methods. There is a single base URL for all requests. The base URL specifies the Internet host and port, and **optionally** a path, of an HTTP server acting as a repository. Repositories expose their base URL as the value of the `baseURL` element in the `Identify` response. Note that the composition of any path is determined by the configuration of the repository's HTTP server.

In addition to the base URL, all requests consist of a list of *keyword arguments*, which take the form of `key=value` pairs. Arguments may appear in any order and multiple arguments **must** be separated by ampersands [`&`]. Each OAI-PMH request **must** have at least one `key=value` pair that specifies the OAI-PMH request issued by the harvester:

- `key` is the string `'verb'`;
- `value` is one of the defined OAI-PMH requests.

The number and nature of additional `key=value` pairs depends on the arguments for the individual request.

3.1.1.1 Encoding an OAI-PMH request in a URL for an HTTP GET

URLs for `GET` requests have keyword arguments appended to the base URL, separated from it by a question mark [`?`]. For example, the URL of a `GetRecord` request to a repository with base URL that is `http://an.oa.org/OAI-script` might be:

```
http://an.oa.org/OAI-script?
verb=GetRecord&identifier=oai:arXiv.org:hep-th/9901001&metadataPrefix=oai_dc
```

However, since special characters in URIs **must** be encoded, the correct form of the above `GET` request URL is:

```
http://an.oa.org/OAI-script?
verb=GetRecord&identifier=oai%3AarXiv.org%3Ahep-th%2F9901001&metadataPrefix=oai_dc
```

3.1.1.2 Encoding an OAI-PMH request in an HTTP POST

Keyword arguments are carried in the message body of the HTTP `POST`. The `Content-Type` of the request **must** be `application/x-www-form-urlencoded`. For example, submitting the same request as above using the `POST` method would use just the base URL as the URL, with the format of the `POST` being:

```
POST http://an.oa.org/OAI-script HTTP/1.0
Content-Length: 82
Content-Type: application/x-www-form-urlencoded

verb=GetRecord&identifier=oai%3AarXiv.org%3Ahep-th%2F9901001&metadataPrefix=oai_dc
```

3.1.1.3 Encoding of special characters in keyword arguments of OAI-PMH requests

The syntax rules for URIs restrict a few characters to special roles in certain contexts, and require that if these characters are used in any other way that they **must** be written as an escape sequence, i.e. a percent sign followed by the character code in hexadecimal. The reserved characters include:

Character	*URI Role*	*Escape Sequence*
/	Path Component Separator	%2F
?	Query Component Separator	%3F
#	Fragment Identifier	%23
=	Name/Value Separator	%3D
&	Argument Separator in Query Component	%26
:	Host Port Separator	%3A
;	Authority Namespace Separator	%3B
	Space Character	%20
%	Escape Indicator	%25
+	Escaped Space	%2B

As a result, these characters **must** be represented by their respective escape sequence if their use does not correspond to their established *URI role*. In case of the OAI-PMH, this means that the reserved characters **must** be encoded when they appear in the `value` part of the `key=value` pairs of the request. This applies for both the `GET` and `POST` encoding of the OAI-PMH requests.

3.1.2 HTTP Response Format

Responses to requests are formatted as HTTP responses, with appropriate HTTP header fields.

3.1.2.1 Content-Type

The `Content-Type` returned for all OAI-PMH requests **must** be `text/xml`.

3.1.2.2 Status-Code

OAI-PMH errors are distinguished from HTTP `Status-Codes`. Since OAI-PMH uses HTTP as a transport layer, servers implementing OAI-PMH **must** conform to HTTP status code definitions and report relevant HTTP transport layer status via those `Status-Codes`. OAI-PMH repositories **may** employ HTTP `Status-Codes` in addition to "`200 OK`". For instance, the following `Status-Codes` may be useful for load balancing in OAI repositories:

- `302` - Allows the repository to temporarily redirect an OAI-PMH request to another repository. The URI of the temporary repository **should** be given by the `Location` field in the HTTP response.
- `503` - Service unavailable, a `Retry-After` period is specified. Harvesters **should** wait this period before attempting another OAI-PMH request.

3.1.3 Response Compression

Response compression is **optional** in OAI-PMH. Compression of responses to OAI-PMH requests is handled at the level of HTTP, with the following restrictions:

- Harvesters **may** include an `Accept-Encoding` header in their OAI-PMH requests to specify response compression preferences.
- Harvesters that do not include an `Accept-Encoding` header in their requests will always receive uncompressed responses.
- When a request includes an `Accept-Encoding` header the list of encodings **must** include the `identity` (no compression) encoding (with a non-zero `qvalue`).
- Repositories **must** support the HTTP `identity` encoding.
- Repositories **should** express the encodings they support in addition to `identity` by including `compression` elements in the `Identify` response.

3.2 XML Response Format

All responses to OAI-PMH requests **must** be well-formed XML instance documents. Encoding of the XML **must** use the UTF-8 representation of Unicode. Character references, rather than entity references, **must** be used. Character references allow XML responses to be treated as stand-alone documents that can be manipulated without dependency on entity declarations external to the document.

The XML data for all responses to OAI-PMH requests **must** validate against the XML Schema shown at the end of this section . As can be seen from that schema, responses to OAI-PMH requests have the following common markup:

1. The first tag output is an XML declaration where the `version` is always `1.0` and the `encoding` is always `UTF-8`, eg: `<?xml version="1.0" encoding="UTF-8" ?>`
2. The remaining content is enclosed in a root element with the name OAI-PMH. This element **must** have three attributes that define the XML namespaces used in the remainder of the response and the location of the validating schema:
    - `xmlns` -- the value of which **must** be the namespace URI of the OAI-PMH (`http://www.openarchives.org/OAI/2.0/`).
    - `xmlns:xsi` -- the value of which **must** be the namespace URI for XML schema (`http://www.w3.org/2001/XMLSchema-instance`).
    - `xsi:schemaLocation` -- is a pair, the first part of which is the namespace URI (as defined by the XML namespace specification) of the OAI-PMH (`http://www.openarchives.org/OAI/2.0/`), and the second part is the URL of the XML schema for validation of the response (`http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd`).
3. For all responses, the first two children of the root element are:
    - `responseDate` -- a UTCdatetime indicating the time and date that the response was sent. This **must** be expressed in UTC.
    - `request` -- indicating the protocol request that generated this response. The rules for generating the `request` element are as follows:
        - The content of the `request` element **must** always be the base URL of the protocol request;
        - The only valid attributes for the `request` element are the `keys` of the `key=value` pairs of protocol request. The attribute values must be the corresponding `values` of those `key=value` pairs;

- In cases where the request that generated this response did not result in an error or exception condition, the attributes and attribute values of the `request` element **must** match the `key=value` pairs of the protocol request;
- In cases where the request that generated this response resulted in `a badVerb` or `badArgument` error condition, the repository **must** return the base URL of the protocol request only. Attributes **must not** be provided in these cases.

4. The third child of the root element is either:
   - an `error` element that **must** be used in case of an error or exception condition;
   - an element with the same name as the verb of the respective OAI-PMH request.

An example of a successful reply to the `GetRecord` request shown above is of the form:

```
<?xml version="1.0" encoding="UTF-8" ?>
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
         http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
 <responseDate>2002-05-01T19:20:30Z</responseDate>
 <request verb="GetRecord" identifier="oai:arXiv.org:hep-th/9901001"
          metadataPrefix="oai_dc">http://an.oa.org/OAI-script</request>
 <GetRecord>
  <record>
      ...
  </record>
 </GetRecord>
</OAI-PMH>
```

3.2.1 XML Schema for Validating Responses to OAI-PMH Requests

```
<schema targetNamespace="http://www.openarchives.org/OAI/2.0/"
        xmlns="http://www.w3.org/2001/XMLSchema"
        xmlns:oai="http://www.openarchives.org/OAI/2.0/"
        elementFormDefault="qualified"
        attributeFormDefault="unqualified">

  <annotation>
    <documentation>
    XML Schema which can be used to validate replies to all OAI-PMH
    v2.0 requests. Herbert Van de Sompel, 2002-05-13.
    Validated with XML Spy v.4.3 on 2002-05-13.
    Validated with XSV 1.203.2.45/1.106.2.22 on 2002-05-13.
    Added definition of protocolVersionType instead of using anonymous
    type. No change of function. Simeon Warner, 2004-03-29.
    Tightened definition of UTCdatetimeType to enforce the restriction
    to UTC Z notation. Simeon Warner, 2004-09-14.
    Corrected pattern matches for setSpecType and metadataPrefixType
    to agree with protocol specification. Simeon Warner, 2004-10-12.
    Spelling correction. Simeon Warner, 2008-12-07.
    $Date: 2008/12/07 20:58:40 $
    </documentation>
  </annotation>

  <element name="OAI-PMH" type="oai:OAI-PMHtype"/>

  <complexType name="OAI-PMHtype">
    <sequence>
      <element name="responseDate" type="dateTime"/>
      <element name="request" type="oai:requestType"/>
      <choice>
        <element name="error" type="oai:OAI-PMHerrorType" maxOccurs="unbounded"/>
        <element name="Identify" type="oai:IdentifyType"/>
```

```
      <element name="ListMetadataFormats" type="oai:ListMetadataFormatsType"/>
      <element name="ListSets" type="oai:ListSetsType"/>
      <element name="GetRecord" type="oai:GetRecordType"/>
      <element name="ListIdentifiers" type="oai:ListIdentifiersType"/>
      <element name="ListRecords" type="oai:ListRecordsType"/>
    </choice>
  </sequence>
</complexType>

<complexType name="requestType">
  <annotation>
    <documentation>Define requestType, indicating the protocol request that
    led to the response. Element content is BASE-URL, attributes are arguments
    of protocol request, attribute-values are values of arguments of protocol
    request</documentation>
  </annotation>
  <simpleContent>
    <extension base="anyURI">
      <attribute name="verb" type="oai:verbType" use="optional"/>
      <attribute name="identifier" type="oai:identifierType" use="optional"/>
      <attribute name="metadataPrefix" type="oai:metadataPrefixType" use="optional"/>
      <attribute name="from" type="oai:UTCdatetimeType" use="optional"/>
      <attribute name="until" type="oai:UTCdatetimeType" use="optional"/>
      <attribute name="set" type="oai:setSpecType" use="optional"/>
      <attribute name="resumptionToken" type="string" use="optional"/>
    </extension>
  </simpleContent>
</complexType>

<simpleType name="verbType">
  <restriction base="string">
    <enumeration value="Identify"/>
    <enumeration value="ListMetadataFormats"/>
    <enumeration value="ListSets"/>
    <enumeration value="GetRecord"/>
    <enumeration value="ListIdentifiers"/>
    <enumeration value="ListRecords"/>
  </restriction>
</simpleType>

<!-- define OAI-PMH error conditions -->
<!-- ============================== -->

<complexType name="OAI-PMHerrorType">
  <simpleContent>
    <extension base="string">
      <attribute name="code" type="oai:OAI-PMHerrorcodeType" use="required"/>
    </extension>
  </simpleContent>
</complexType>

<simpleType name="OAI-PMHerrorcodeType">
  <restriction base="string">
    <enumeration value="cannotDisseminateFormat"/>
    <enumeration value="idDoesNotExist"/>
    <enumeration value="badArgument"/>
    <enumeration value="badVerb"/>
    <enumeration value="noMetadataFormats"/>
    <enumeration value="noRecordsMatch"/>
    <enumeration value="badResumptionToken"/>
    <enumeration value="noSetHierarchy"/>
  </restriction>
</simpleType>

<!-- define OAI-PMH verb containers -->
<!-- ============================== -->

<complexType name="IdentifyType">
```

```
    <sequence>
      <element name="repositoryName" type="string"/>
      <element name="baseURL" type="anyURI"/>
      <element name="protocolVersion" type="oai:protocolVersionType"/>
      <element name="adminEmail" type="oai:emailType" maxOccurs="unbounded"/>
      <element name="earliestDatestamp" type="oai:UTCdatetimeType"/>
      <element name="deletedRecord" type="oai:deletedRecordType"/>
      <element name="granularity" type="oai:granularityType"/>
      <element name="compression" type="string" minOccurs="0" maxOccurs="unbounded"/>
      <element name="description" type="oai:descriptionType"
               minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
  </complexType>

  <complexType name="ListMetadataFormatsType">
    <sequence>
      <element name="metadataFormat" type="oai:metadataFormatType" maxOccurs="unbounded"/>
    </sequence>
  </complexType>

  <complexType name="ListSetsType">
    <sequence>
      <element name="set" type="oai:setType" maxOccurs="unbounded"/>
      <element name="resumptionToken" type="oai:resumptionTokenType" minOccurs="0"/>
    </sequence>
  </complexType>

  <complexType name="GetRecordType">
    <sequence>
      <element name="record" type="oai:recordType"/>
    </sequence>
  </complexType>

  <complexType name="ListRecordsType">
    <sequence>
      <element name="record" type="oai:recordType" maxOccurs="unbounded"/>
      <element name="resumptionToken" type="oai:resumptionTokenType" minOccurs="0"/>
    </sequence>
  </complexType>

  <complexType name="ListIdentifiersType">
    <sequence>
      <element name="header" type="oai:headerType" maxOccurs="unbounded"/>
      <element name="resumptionToken" type="oai:resumptionTokenType" minOccurs="0"/>
    </sequence>
  </complexType>

  <!-- define basic types used in replies to
       GetRecord, ListRecords, ListIdentifiers -->
  <!-- ====================================== -->

  <complexType name="recordType">
    <annotation>
      <documentation>A record has a header, a metadata part, and
        an optional about container</documentation>
    </annotation>
    <sequence>
      <element name="header" type="oai:headerType"/>
      <element name="metadata" type="oai:metadataType" minOccurs="0"/>
      <element name="about" type="oai:aboutType" minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
  </complexType>

  <complexType name="headerType">
    <annotation>
      <documentation>A header has a unique identifier, a datestamp,
        and setSpec(s) in case the item from which
        the record is disseminated belongs to set(s).
```

```
      the header can carry a deleted status indicating
      that the record is deleted.</documentation>
  </annotation>
  <sequence>
    <element name="identifier" type="oai:identifierType"/>
    <element name="datestamp" type="oai:UTCdatetimeType"/>
    <element name="setSpec" type="oai:setSpecType" minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
  <attribute name="status" type="oai:statusType" use="optional"/>
</complexType>

<simpleType name="identifierType">
  <restriction base="anyURI"/>
</simpleType>

<simpleType name="statusType">
  <restriction base="string">
    <enumeration value="deleted"/>
  </restriction>
</simpleType>

<complexType name="metadataType">
  <annotation>
    <documentation>Metadata must be expressed in XML that complies
     with another XML Schema (namespace=#other). Metadata must be
     explicitly qualified in the response.</documentation>
  </annotation>
  <sequence>
    <any namespace="##other" processContents="strict"/>
  </sequence>
</complexType>

<complexType name="aboutType">
  <annotation>
    <documentation>Data "about" the record must be expressed in XML
    that is compliant with an XML Schema defined by a community.</documentation>
  </annotation>
  <sequence>
    <any namespace="##other" processContents="strict"/>
  </sequence>
</complexType>

<complexType name="resumptionTokenType">
  <annotation>
    <documentation>A resumptionToken may have 3 optional attributes
     and can be used in ListSets, ListIdentifiers, ListRecords
     responses.</documentation>
  </annotation>
  <simpleContent>
    <extension base="string">
      <attribute name="expirationDate" type="dateTime" use="optional"/>
      <attribute name="completeListSize" type="positiveInteger" use="optional"/>
      <attribute name="cursor" type="nonNegativeInteger" use="optional"/>
    </extension>
  </simpleContent>
</complexType>

<complexType name="descriptionType">
  <annotation>
    <documentation>The descriptionType is used for the description
    element in Identify and for setDescription element in ListSets.
    Content must be compliant with an XML Schema defined by a
    community.</documentation>
  </annotation>
  <sequence>
    <any namespace="##other" processContents="strict"/>
  </sequence>
</complexType>
```

```
<simpleType name="UTCdatetimeType">
  <annotation>
    <documentation>Datestamps are to either day (type date)
    or to seconds granularity (type oai:UTCdateTimeZType)</documentation>
  </annotation>
  <union memberTypes="date oai:UTCdateTimeZType"/>
</simpleType>

<simpleType name="UTCdateTimeZType">
  <restriction base="dateTime">
    <pattern value=".*Z"/>
  </restriction>
</simpleType>

<!-- define types used for Identify verb only -->
<!-- ======================================== -->

<simpleType name="protocolVersionType">
  <restriction base="string">
    <enumeration value="2.0"/>
  </restriction>
</simpleType>

<simpleType name="emailType">
  <restriction base="string">
    <pattern value="\S+@(\S+\.)+\S+"/>
  </restriction>
</simpleType>

<simpleType name="deletedRecordType">
  <restriction base="string">
    <enumeration value="no"/>
    <enumeration value="persistent"/>
    <enumeration value="transient"/>
  </restriction>
</simpleType>

<simpleType name="granularityType">
  <restriction base="string">
    <enumeration value="YYYY-MM-DD"/>
    <enumeration value="YYYY-MM-DDThh:mm:ssZ"/>
  </restriction>
</simpleType>

<!-- define types used for ListMetadataFormats verb only -->
<!-- ================================================== -->

<complexType name="metadataFormatType">
  <sequence>
    <element name="metadataPrefix" type="oai:metadataPrefixType"/>
    <element name="schema" type="anyURI"/>
    <element name="metadataNamespace" type="anyURI"/>
  </sequence>
</complexType>

<simpleType name="metadataPrefixType">
  <restriction base="string">
    <pattern value="[A-Za-z0-9\-_\.!~\*'\(\)]+"/>
  </restriction>
</simpleType>

<!-- define types used for ListSets verb -->
<!-- ================================== -->

<complexType name="setType">
  <sequence>
    <element name="setSpec" type="oai:setSpecType"/>
```

```
      <element name="setName" type="string"/>
      <element name="setDescription" type="oai:descriptionType"
               minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
  </complexType>

  <simpleType name="setSpecType">
    <restriction base="string">
      <pattern value="([A-Za-z0-9\-_\.!~\*'\(\)])+(:[A-Za-z0-9\-_\.!~\*'\(\)]+)*"/>
    </restriction>
  </simpleType>

</schema>
```

This Schema is available at http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd

3.3 UTCdatetime

Dates and times are uniformly encoded using ISO8601 and are expressed in UTC throughout the protocol. When time is included, the special UTC designator ("Z") **must** be used. UTC is implied for dates although no timezone designator is specified. For example, `1957-03-20T20:30:00Z` is UTC 8:30:00 PM on March 20th 1957. UTCdatetime is used in both protocol requests and protocol replies, in the way described in the following sections.

3.3.1 UTCdatetime in Protocol Requests

Datestamps used as values of the **optional** arguments `from` and `until` in the `ListIdentifiers` and `ListRecords` requests are encoded using ISO8601 and are expressed in UTC. These arguments are used to specify datestamp-based selective harvesting. These arguments support the "Complete date" and the "Complete date plus hours, minutes and seconds" granularities defined in ISO8601. The legitimate formats are `YYYY-MM-DD` and `YYYY-MM-DDThh:mm:ssZ`. Both arguments **must** have the same granularity. All repositories **must** support `YYYY-MM-DD`. A repository that supports `YYYY-MM-DDThh:mm:ssZ` **should** indicate so in the `Identify` response. A request by a harvester with finer granularity than that supported by a repository **must** produce an error.

3.3.2 UTCdatetime in Protocol Responses

Datestamps appear in the headers of records that are returned in response to `ListIdentifiers` , `GetRecord` and `ListRecords` requests. These datestamps are encoded using ISO8601 and are expressed in UTC; they **must** be expressed in the finest granularity supported by the repository. The value of the datestamp must correspond to the rules for datestamp-based selective harvesting.

Each protocol response includes a `responseDate` element, which **must** be the time and date of the response in UTC. This is encoded using the "Complete date plus hours, minutes, and seconds" variant of ISO8601. This format is `YYYY-MM-DDThh:mm:ssZ`.

A `resumptionToken` in a protocol reply **may** include an **optional** argument `expirationDate`, which is expressed in UTC. This is encoded using the "Complete date plus hours, minutes, and seconds" variant of ISO8601. This format is `YYYY-MM-DDThh:mm:ssZ`.

3.4 metadataPrefix and Metadata Schema

OAI-PMH supports the dissemination of records in multiple metadata formats from a repository. The `ListMetadataFormats` request returns the list of all metadata formats available from a repository, each of which has the following properties:

- The `metadataPrefix` - a string to specify the metadata format in OAI-PMH requests issued to the repository. `metadataPrefix` consists of any valid URI *unreserved* characters. `metadataPrefix` arguments are used in

ListRecords, ListIdentifiers, and GetRecord requests to retrieve records, or the headers of records that include metadata in the format specified by the metadataPrefix;
- The *metadata schema* URL - the URL of an XML schema to test validity of metadata expressed according to the format;
- The XML namespace URI that is a global identifier of the metadata format.

The metadata in each record returned by ListRecords and GetRecord **must** comply with the conventions of the XML namespace specification. This means that the root element of the metadata part **must** contain an xmlns attribute, the value of which is the XML namespace URI of the metadata format. The root element **must** also contain an xsi:schemaLocation attribute that has a value that includes the URL of the XML schema for validation of the metadata. This URL **must** match the URL of the metadata schema for the metadataPrefix included as an argument to the ListRecords or GetRecord request (the mapping from metadataPrefix to metadata schema is defined by the repository's response to the ListMetadataFormats request).

For purposes of interoperability, repositories **must** disseminate Dublin Core, without any qualification. Therefore, the protocol reserves the metadataPrefix `oai_dc', and the URL of a metadata schema for unqualified Dublin Core, which is http://www.openarchives.org/OAI/2.0/oai_dc.xsd. The corresponding XML namespace URI is http://www.openarchives.org/OAI/2.0/oai_dc/.

The metadataPrefix `all' is reserved for future use. Implementations **should not** use this metadataPrefix.

Communities **should** adopt guidelines for sharing of metadataPrefixes, metadata schema and XML namespace URI's of metadata formats. Such guidelines are outside of the scope of the OAI-PMH. The accompanying Implementation Guidelines document provides some sample XML Schema and instance documents for common metadata formats such as MARC and RFC 1807.

3.5 Flow Control

A number of OAI-PMH requests return a *list* of discrete entities: ListRecords returns a list of records, ListIdentifiers returns a list of headers, and ListSets returns a list of sets. Collectively these requests are called *list requests*. In some cases, these lists may be large and it may be practical to partition them among a series of requests and responses. This partitioning is accomplished as follows:

- A repository replies to a request with an *incomplete list* and a resumptionToken;
- In order to make the response a *complete list*, the harvester will need to issue one or more requests with resumptionTokens as arguments. The complete list then consists of the concatenation of the *incomplete lists* from the sequence of requests, known as a *list request sequence*.

Details of flow control and the resumptionToken are as follows:

- The only defined use of resumptionToken is as follows:
  - a repository **must** include a resumptionToken element as part of each response that includes an incomplete list;
  - in order to retrieve the next portion of the complete list, the next request **must** use the value of that resumptionToken element as the value of the resumptionToken argument of the request;
  - the response containing the incomplete list that completes the list **must** include an empty resumptionToken element;

  All other uses of resumptionToken by a harvester are illegal and **must** return an error.
- In all cases when a resumptionToken is issued, the incomplete list **must** consist of complete entities; e.g., all individual records returned in an incomplete record list from a ListRecords request **must** be intact.
- The format of the resumptionToken is not defined by the OAI-PMH and **should** be considered opaque by the harvester.
- The protocol does not define the semantics of incompleteness. Therefore, a harvester **should not** assume that the members in an *incomplete list* conform to some selection criteria (e.g., date ordering).
- Before including a resumptionToken in the URL of a subsequent request, a harvester **must** encode any special characters in it.

The following **optional** attributes **may** be included as part of the `resumptionToken` element along with the `resumptionToken` itself:

- `expirationDate` -- a UTCdatetime indicating when the `resumptionToken` ceases to be valid.
- `completeListSize` -- an integer indicating the cardinality of the complete list (i.e., the sum of the cardinalities of the incomplete lists). Because there may be changes in a repository during a list request sequence, as described under Idempotency of resumptionTokens, the value of `completeListSize` may be only an estimate of the actual cardinality of the complete list and may be revised during the list request sequence.
- `cursor` -- a count of the number of elements of the complete list thus far returned (i.e. `cursor` starts at 0).

The following example is a series of ListRecords requests where the complete list consists of 175 records and the repository only returns 100 records per response.

- The harvester issues a `ListRecords` request.
- The repository responds with an incomplete list of 100 records. The repository marks this list as incomplete by including in the response a non-empty `resumptionToken` element, with two attributes: a `completeListSize` of 175, and a `cursor` of 0.
- The harvester issues a subsequent `ListRecords` request that includes the `resumptionToken` that it received in the previous response.
- The repository responds with an incomplete list of 75 records. The repository marks this list as the final incomplete list by including in the response an empty `resumptionToken` element with two attributes: a `completeListSize` of 175, and a `cursor` of 100.

This flow control mechanism, in combination with HTTP transport layer facilities, provides some basic tools with which a repository can enforce an *acceptable use policy* for its harvesting interface. Communities implementing the OAI-PMH may need more extensive tools to enforce acceptable use policies for either the harvesting interface of their repositories or for the metadata harvested from those repositories. The enforcement of such additional policies is outside of the scope of the OAI-PMH.

3.5.1 Idempotency of resumptionTokens

Repositories that implement `resumptionTokens` **must** do so in a manner that allows harvesters to resume a sequence of requests for incomplete lists by re-issuing a list request with the most recent `resumptionToken`. The purpose of this is to allow harvesters to recover from network or other errors that would otherwise mean that the list request sequence would have to be started again. A re-issue of a list request with a `resumptionToken` occurs in two contexts:

1. *When there are no changes in the repository*. There are no changes to the complete list returned by the list request sequence. In this case, the repository **must** return the same incomplete list when the most recent list request, i.e. the one with the most recent non-expired `resumptionToken`, is re-issued.
2. *When there are changes in the repository*. There may be changes to the complete list returned by the list request sequence. These changes occur when the records disseminated in the list move in or out of the datestamp range of the request because of changes, modifications, or deletions in the repository. In this case, strict idempotency of the incomplete-list requests using `resumptionToken` values is not required. Instead, the incomplete list returned in response to a re-issued request **must** include all records with unchanged `datestamps` within the range of the initial list request. The incomplete list returned in response to a re-issued request **may** contain records with datestamps that either moved into or out of the range of the initial request. In cases where there are substantial changes to the repository, it **may** be appropriate for a repository to return a `badResumptionToken` error, signaling that the harvester should restart the list request sequence.

3.6 Error and Exception Conditions

In event of an error or exception condition, repositories **must** indicate OAI-PMH errors, distinguished from HTTP `Status-Codes`, by including one or more `error` elements in the response. While one `error` element is sufficient to indicate the presence of the error or exception condition, repositories **should** report all errors or exceptions that arise from processing the request. Each `error` element **must** have a `code` attribute that **must** be from the following table;

each `error` element **may** also have a free text string value to provide information about the error that is useful to a human reader. These strings are not defined by the OAI-PMH.

Error Codes	*Description*	*Applicable Verbs*
`badArgument`	The request includes illegal arguments, is missing required arguments, includes a repeated argument, or values for arguments have an illegal syntax.	*all verbs*
`badResumptionToken`	The value of the `resumptionToken` argument is invalid or expired.	`ListIdentifiers` `ListRecords` `ListSets`
`badVerb`	Value of the `verb` argument is not a legal OAI-PMH verb, the verb argument is missing, or the `verb` argument is repeated.	*N/A*
`cannotDisseminateFormat`	The metadata format identified by the value given for the `metadataPrefix` argument is not supported by the item or by the repository.	`GetRecord` `ListIdentifiers` `ListRecords`
`idDoesNotExist`	The value of the `identifier` argument is unknown or illegal in this repository.	`GetRecord` `ListMetadataFormats`
`noRecordsMatch`	The combination of the values of the `from`, `until`, `set` and `metadataPrefix` arguments results in an empty list.	`ListIdentifiers` `ListRecords`
`noMetadataFormats`	There are no metadata formats available for the specified item.	`ListMetadataFormats`
`noSetHierarchy`	The repository does not support sets.	`ListSets` `ListIdentifiers` `ListRecords`

The following example demonstrates error handling in the case of an illegal verb argument. All request URLs shown from now on will be wrapped to make them more readable.

Request

```
http://arXiv.org/oai2?
       verb=nastyVerb
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
         http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
  <responseDate>2002-05-01T09:18:29Z</responseDate>
  <request>http://arXiv.org/oai2</request>
  <error code="badVerb">Illegal OAI verb</error>
</OAI-PMH>
```

The following example demonstrates error handling in the case of a `ListSets` request to a repository that does not handle sets.

Request

```
http://arXiv.org/oai2?
       verb=ListSets
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
         http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
  <responseDate>2002-05-01T09:18:29Z</responseDate>
  <request verb="ListSets">http://arXiv.org/oai2</request>
  <error code="noSetHierarchy">This repository does not
    support sets</error>
</OAI-PMH>
```

4. Protocol Requests and Responses

This section lists the requests, or `verbs`, defined in the OAI-PMH. The documentation for each request is organized as follows:

- A section title corresponding to the token used to specify the request as the required `verb` argument to an HTTP request.
- A brief summary of the meaning of the verb and notes on its usage.
- The list of additional arguments for the request. Arguments are of three types:
  - *required*, the argument **must** be included with the request (the `verb` argument is always *required*, as described in HTTP Request Format).
  - *optional*, the argument **may** be included with the request.
  - *exclusive*, the argument **may** be included with request, but **must** be the only argument (in addition to the `verb` argument).
- Error and exception conditions specific to the protocol request.
- One or more example requests and corresponding responses, with explanatory notes if appropriate.

An XML Schema defines the format of valid replies to all OAI-PMH requests.

4.1 GetRecord

Summary and Usage Notes

This verb is used to retrieve an individual metadata record from a repository. Required arguments specify the identifier of the item from which the record is requested and the format of the metadata that should be included in the record. Depending on the level at which a repository tracks deletions, a header with a "deleted" value for the `status` attribute **may** be returned, in case the metadata format specified by the `metadataPrefix` is no longer available from the repository or from the specified item.

Arguments

- **`identifier`** a *required* argument that specifies the unique identifier of the item in the repository from which the record must be disseminated.
- **`metadataPrefix`** a *required* argument that specifies the `metadataPrefix` of the format that should be included in the metadata part of the returned record . A record should only be returned if the format specified by the `metadataPrefix` can be disseminated from the item identified by the value of the identifier argument. The metadata formats supported by a repository and for a particular record can be retrieved using the `ListMetadataFormats` request.

Error and Exception Conditions

- **`badArgument`** - The request includes illegal arguments or is missing required arguments.
- **`cannotDisseminateFormat`** - The value of the `metadataPrefix` argument is not supported by the item identified by the value of the `identifier` argument.
- **`idDoesNotExist`** - The value of the `identifier` argument is unknown or illegal in this repository.

Examples

Request

Request a record in the Dublin Core metadata format [URL shown without encoding to be more readable].

```
http://arXiv.org/oai2?
       verb=GetRecord&identifier=oai:arXiv.org:cs/0112017&metadataPrefix=oai_dc
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
         http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
  <responseDate>2002-02-08T08:55:46Z</responseDate>
  <request verb="GetRecord" identifier="oai:arXiv.org:cs/0112017"
           metadataPrefix="oai_dc">http://arXiv.org/oai2</request>
  <GetRecord>
   <record>
    <header>
      <identifier>oai:arXiv.org:cs/0112017</identifier>
      <datestamp>2001-12-14</datestamp>
      <setSpec>cs</setSpec>
      <setSpec>math</setSpec>
    </header>
    <metadata>
      <oai_dc:dc
         xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"
         xmlns:dc="http://purl.org/dc/elements/1.1/"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/
         http://www.openarchives.org/OAI/2.0/oai_dc.xsd">
        <dc:title>Using Structural Metadata to Localize Experience of
                  Digital Content</dc:title>
        <dc:creator>Dushay, Naomi</dc:creator>
        <dc:subject>Digital Libraries</dc:subject>
        <dc:description>With the increasing technical sophistication of
            both information consumers and providers, there is
            increasing demand for more meaningful experiences of digital
            information. We present a framework that separates digital
            object experience, or rendering, from digital object storage
            and manipulation, so the rendering can be tailored to
            particular communities of users.
        </dc:description>
        <dc:description>Comment: 23 pages including 2 appendices,
            8 figures</dc:description>
        <dc:date>2001-12-14</dc:date>
      </oai_dc:dc>
    </metadata>
  </record>
 </GetRecord>
</OAI-PMH>
```

Request

Request a record in the Dublin Core metadata format. The requested record, however, can not be returned because the identifier does not exist. Therefore, the response does not contain a `record` container. It does have an `error` element with a `code` attribute that has the value `idDoesNotExist`. [URL shown without encoding for better readability].

```
http://arXiv.org/oai2?
       verb=GetRecord&identifier=oai:arXiv.org:quant-ph/02131001&metadataPrefix=oai_dc
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
         http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
  <responseDate>2002-02-08T08:55:46Z</responseDate>
  <request verb="GetRecord" identifier="oai:arXiv.org:quant-ph/0213001"
           metadataPrefix="oai_dc">http://arXiv.org/oai2</request>
  <error code="idDoesNotExist">No matching identifier in arXiv</error>
</OAI-PMH>
```

Request

Request a record in the oai_marc metadata format. However, the requested metadata format can not be disseminated for this identifier. Therefore, the response contains no record. It does contain an `error` element with a `code` attribute that has the value `cannotDisseminateFormat`. [URL shown without encoding for better readability].

```
http://arXiv.org/oai2?
       verb=GetRecord&identifier=oai:arXiv.org:quant-ph/9901001&metadataPrefix=oai_marc
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
         http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
  <responseDate>2002-02-08T08:55:46Z</responseDate>
  <request verb="GetRecord" identifier="oai:arXiv.org:quant-ph/9901001"
           metadataPrefix="oai_marc">http://arXiv.org/oai1</request>
  <error code="cannotDisseminateFormat"/>
</OAI-PMH>
```

4.2 Identify

Summary and Usage Notes

This verb is used to retrieve information about a repository. Some of the information returned is required as part of the OAI-PMH. Repositories **may** also employ the Identify verb to return additional descriptive information.

Arguments

None

Error and Exception Conditions

- **`badArgument`** - The request includes illegal arguments.

Response Format

The response **must** include one instance of the following elements:

- `repositoryName` : a human readable name for the repository;
- `baseURL` : the base URL of the repository;
- `protocolVersion` : the version of the OAI-PMH supported by the repository;
- `earliestDatestamp` : a UTCdatetime that is the guaranteed lower limit of all datestamps recording changes, modifications, or deletions in the repository. A repository **must not** use datestamps lower than the one specified by the content of the `earliestDatestamp` element. `earliestDatestamp` must be expressed at the finest granularity supported by the repository.
- `deletedRecord` : the manner in which the repository supports the notion of deleted records. Legitimate values are `no` ; `transient` ; `persistent` with meanings defined in the section on deletion.
- `granularity`: the finest harvesting granularity supported by the repository. The legitimate values are `YYYY-MM-DD` and `YYYY-MM-DDThh:mm:ssZ` with meanings as defined in ISO8601.

The response **must** include one or more instances of the following element:

- `adminEmail` : the e-mail address of an administrator of the repository.

The response **may** include multiple instances of the following **optional** elements:

- `compression` : a compression encoding supported by the repository. The **recommended** values are those defined for the `Content-Encoding` header in Section 14.11 of RFC 2616 describing HTTP 1.1. A `compression` element **should not** be included for the `identity` encoding, which is implied.
- `description` : an extensible mechanism for communities to describe their repositories. For example, the `description` container could be used to include collection-level metadata in the response to the Identify request. Implementation Guidelines are available to give directions with this respect. Each `description` container **must** be accompanied by the URL of an XML schema describing the structure of the description container.

Examples

Request

```
http://memory.loc.gov/cgi-bin/oai?
       verb=Identify
```

Response

The below example of a response to the `Identify` request contains three `description` containers:

- The `oai-identifier` container complies to an XML Schema, which is available at http://www.openarchives.org/OAI/2.0/oai-identifier.xsd. This schema, provided in the accompanying Implementation Guidelines document, is used by repositories that choose to comply with a specific format of unique identifiers for items. The format of that identifier is explained by means of comments in the oai-identifier.xsd XML Schema.
- The `eprints` container complies to an XML Schema, which is available at http://www.openarchives.org/OAI/1.1/eprints.xsd. This schema, provided in the accompanying Implementation Guidelines document, has been agreed upon by the OAI e-print community, and contains information specific to repositories in that community.

- The `friends` container complies to an XML Schema, which is available at http://www.openarchives.org/OAI/2.0/friends.xsd. This schema, provided in the accompanying Implementation Guidelines document, is used by repositories that want to point harvesters to other repositories, by listing their base URLs. Usage of the `friends` container is **recommended**; it may support harvesters in discovering the network-location of repositories.

```
<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
         http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
  <responseDate>2002-02-08T12:00:01Z</responseDate>
  <request verb="Identify">http://memory.loc.gov/cgi-bin/oai</request>
  <Identify>
    <repositoryName>Library of Congress Open Archive Initiative
                    Repository 1</repositoryName>
    <baseURL>http://memory.loc.gov/cgi-bin/oai</baseURL>
    <protocolVersion>2.0</protocolVersion>
    <adminEmail>somebody@loc.gov</adminEmail>
    <adminEmail>anybody@loc.gov</adminEmail>
    <earliestDatestamp>1990-02-01T12:00:00Z</earliestDatestamp>
    <deletedRecord>transient</deletedRecord>
    <granularity>YYYY-MM-DDThh:mm:ssZ</granularity>
    <compression>deflate</compression>
    <description>
      <oai-identifier
        xmlns="http://www.openarchives.org/OAI/2.0/oai-identifier"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation=
            "http://www.openarchives.org/OAI/2.0/oai-identifier
        http://www.openarchives.org/OAI/2.0/oai-identifier.xsd">
        <scheme>oai</scheme>
        <repositoryIdentifier>lcoa1.loc.gov</repositoryIdentifier>
        <delimiter>:</delimiter>
        <sampleIdentifier>oai:lcoa1.loc.gov:loc.music/musdi.002</sampleIdentifier>
      </oai-identifier>
    </description>
    <description>
      <eprints
         xmlns="http://www.openarchives.org/OAI/1.1/eprints"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://www.openarchives.org/OAI/1.1/eprints
         http://www.openarchives.org/OAI/1.1/eprints.xsd">
        <content>
          <URL>http://memory.loc.gov/ammem/oamh/lcoa1_content.html</URL>
          <text>Selected collections from American Memory at the Library
                of Congress</text>
        </content>
        <metadataPolicy/>
        <dataPolicy/>
      </eprints>
    </description>
    <description>
      <friends
          xmlns="http://www.openarchives.org/OAI/2.0/friends/"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/friends/
         http://www.openarchives.org/OAI/2.0/friends.xsd">
       <baseURL>http://oai.east.org/foo/</baseURL>
       <baseURL>http://oai.hq.org/bar/</baseURL>
       <baseURL>http://oai.south.org/repo.cgi</baseURL>
     </friends>
   </description>
 </Identify>
</OAI-PMH>
```

4.3 ListIdentifiers

Summary and Usage Notes

This verb is an abbreviated form of `ListRecords`, retrieving only headers rather than records. Optional arguments permit selective harvesting of headers based on set membership and/or datestamp. Depending on the repository's support for deletions, a returned header **may** have a `status` attribute of "deleted" if a record matching the arguments specified in the request has been deleted.

Arguments

- **`from`** an *optional* argument with a UTCdatetime value, which specifies a lower bound for datestamp-based selective harvesting.
- **`until`** an *optional* argument with a UTCdatetime value, which specifies a upper bound for datestamp-based selective harvesting.
- **`metadataPrefix`** a *required* argument, which specifies that headers should be returned only if the metadata format matching the supplied `metadataPrefix` is available or, depending on the repository's support for deletions, has been deleted. The metadata formats supported by a repository and for a particular item can be retrieved using the `ListMetadataFormats` request.
- **`set`** an *optional* argument with a `setSpec` value , which specifies set criteria for selective harvesting.
- **`resumptionToken`** an *exclusive* argument with a value that is the flow control token returned by a previous `ListIdentifiers` request that issued an incomplete list.

Error and Exception Conditions

- **`badArgument`** - The request includes illegal arguments or is missing required arguments.
- **`badResumptionToken`** - The value of the `resumptionToken` argument is invalid or expired.
- **`cannotDisseminateFormat`** - The value of the `metadataPrefix` argument is not supported by the repository.
- **`noRecordsMatch`**- The combination of the values of the `from`, `until`, and `set` arguments results in an empty list.
- **`noSetHierarchy`** - The repository does not support sets.

Examples

Request

List the headers of records in the oldArXiv metadata format that are added, modified or deleted since January 15, 1998 in the set physics:hep. [URL shown without encoding for better readability].

```
http://an.oa.org/OAI-script?
       verb=ListIdentifiers&from=1998-01-15&metadataPrefix=oldArXiv&set=physics:hep
```

Response

A list of four headers is returned. One header has a `deleted` status, indicating that a record in the metadata format specified by the `metadataPrefix` is no longer available. In addition, a `resumptionToken` (non-empty, value `xxx45abttyz`) has been returned, indicating that the list of headers is *incomplete* and that one or more subsequent requests will need to be issued to retrieve a *complete* list. In the example, the `resumptionToken` comes with all of the 3 optional attributes: `expirationDate` indicates that the `resumptionToken` will become unusable after 11:20 PM UTC on June 1st 2002; `completeListSize` indicates that the *complete* list consists of 6 identifiers; the zero-value for `cursor` indicates that no headers have been returned previous to this reply.

```
<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
```

```
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
        http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
  <responseDate>2002-06-01T19:20:30Z</responseDate>
  <request verb="ListIdentifiers" from="1998-01-15"
          metadataPrefix="oldarXiv"
          set="physics:hep">http://an.oa.org/OAI-script</request>
  <ListIdentifiers>
   <header>
    <identifier>oai:arXiv.org:hep-th/9801001</identifier>
    <datestamp>1999-02-23</datestamp>
    <setSpec>physic:hep</setSpec>
   </header>
   <header>
    <identifier>oai:arXiv.org:hep-th/9801002</identifier>
    <datestamp>1999-03-20</datestamp>
    <setSpec>physic:hep</setSpec>
    <setSpec>physic:exp</setSpec>
   </header>
   <header>
    <identifier>oai:arXiv.org:hep-th/9801005</identifier>
    <datestamp>2000-01-18</datestamp>
    <setSpec>physic:hep</setSpec>
   </header>
   <header status="deleted">
    <identifier>oai:arXiv.org:hep-th/9801010</identifier>
    <datestamp>1999-02-23</datestamp>
    <setSpec>physic:hep</setSpec>
    <setSpec>math</setSpec>
   </header>
   <resumptionToken expirationDate="2002-06-01T23:20:00Z"
      completeListSize="6"
      cursor="0">xxx45abttyz</resumptionToken>
 </ListIdentifiers>
</OAI-PMH>
```

Request

Issue a subsequent request to the one issued above. The single `resumptionToken` argument has the value returned in the previous response. [URL shown without encoding for better readability].

```
http://an.oa.org/OAI-script?
       verb=ListIdentifiers&resumptionToken=xxx45abttyz
```

Response

Two more headers are returned. The `resumptionToken` element at the end of the list has no value, indicating that the list is now complete. The value of the `completeListSize` attribute remains 6, while the value of the `cursor` attribute has changed to 4, indicating that a previous reply has (or previous replies have) already delivered 4 identifiers.

```
<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
         http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
  <responseDate>2002-06-01T19:30:00Z</responseDate>
  <request verb="ListIdentifiers"
          resumptionToken="xxx45abttyz">http://an.oa.org/OAI-script</request>
  <ListIdentifiers>
    <header>
      <identifier>oai:arXiv.org:hep-th/9801020</identifier>
      <datestamp>1999-02-23</datestamp>
```

```
      <setSpec>physic:hep</setSpec>
    </header>
    <header>
    <identifier>oai:arXiv.org:hep-th/9801060</identifier>
      <datestamp>1999-02-23</datestamp>
      <setSpec>physic:hep</setSpec>
    </header>
    <resumptionToken completeListSize="6" cursor="4"/>
  </ListIdentifiers>
</OAI-PMH>
```

Request

List the headers of olac-formatted records, added or modified on January 1, 2001 in the set Perseus:collection:PersInfo. There are no matches for this request, hence, the response contains an error tag and does not contain any header elements [URL shown without encoding for better readability].

```
http://www.perseus.tufts.edu/cgi-bin/pdataprov?
       verb=ListIdentifiers&metadataPrefix=olac&from=2001-01-01&until=2001-01-01
       &set=Perseus:collection:PersInfo
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
         http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
  <responseDate>2002-02-08T14:27:19Z</responseDate>
  <request verb="ListIdentifiers" metadataPrefix="olac"
    from="2001-01-01" until="2001-01-01"
    set="Perseus:collection:PersInfo">
    http://www.perseus.tufts.edu/cgi-bin/pdataprov</request>
  <error code="noRecordsMatch"/>
</OAI-PMH>
```

4.4 ListMetadataFormats

Summary and Usage Notes

This verb is used to retrieve the metadata formats available from a repository. An optional argument restricts the request to the formats available for a specific item.

Arguments

- **`identifier`** an *optional* argument that specifies the unique identifier of the item for which available metadata formats are being requested. If this argument is omitted, then the response includes all metadata formats supported by this repository. Note that the fact that a metadata format is supported by a repository does *not* mean that it can be disseminated from all items in the repository.

Error and Exception Conditions

- **`badArgument`** - The request includes illegal arguments or is missing required arguments.
- **`idDoesNotExist`** - The value of the `identifier` argument is unknown or illegal in this repository.
- **`noMetadataFormats`** - There are no metadata formats available for the specified item.

Examples

Request

List the metadata formats that can be disseminated from the repository `http://www.perseus.tufts.edu/cgi-bin/pdataprov` for the item with unique identifier `oai:perseus.tufts.edu:Perseus:text:1999.02.0119` [URL shown without encoding for better readability].

```
http://www.perseus.tufts.edu/cgi-bin/pdataprov?
       verb=ListMetadataFormats&identifier=oai:perseus.tufts.edu:Perseus:text:1999.02.0119
```

Response

The response shows that 3 metadata formats are supported for the given identifier: oai_dc, olac and perseus. For each of the formats, the location of an XML Schema describing the format, as well as the XML Namespace URI is given.

```
<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
         http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
  <responseDate>2002-02-08T14:27:19Z</responseDate>
  <request verb="ListMetadataFormats"
    identifier="oai:perseus.tufts.edu:Perseus:text:1999.02.0119">
    http://www.perseus.tufts.edu/cgi-bin/pdataprov</request>
  <ListMetadataFormats>
   <metadataFormat>
     <metadataPrefix>oai_dc</metadataPrefix>
     <schema>http://www.openarchives.org/OAI/2.0/oai_dc.xsd
       </schema>
     <metadataNamespace>http://www.openarchives.org/OAI/2.0/oai_dc/
       </metadataNamespace>
   </metadataFormat>
   <metadataFormat>
     <metadataPrefix>olac</metadataPrefix>
     <schema>http://www.language-archives.org/OLAC/olac-0.2.xsd</schema>
     <metadataNamespace>http://www.language-archives.org/OLAC/0.2/
      </metadataNamespace>
   </metadataFormat>
   <metadataFormat>
     <metadataPrefix>perseus</metadataPrefix>
     <schema>http://www.perseus.tufts.edu/persmeta.xsd</schema>
     <metadataNamespace>http://www.perseus.tufts.edu/persmeta.dtd
       </metadataNamespace>
   </metadataFormat>
 </ListMetadataFormats>
</OAI-PMH>
```

Request

List the metadata formats that can be disseminated from the repository `http://memory.loc.gov/cgi-bin/oai`.

```
http://memory.loc.gov/cgi-bin/oai?
       verb=ListMetadataFormats
```

Response

The response shows that the repository supports two metadata formats: `oai_dc`, and `oai_marc`. For each of the formats, the location of an XML Schema describing the format is given. The support of these formats at the repository-level does not imply support of each format for each item of the repository.

```
<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
         http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
  <responseDate>2002-06-08T15:19:13Z</responseDate>
  <request verb="ListMetadataFormats">
           http://memory.loc.gov/cgi-bin/oai</request>
  <ListMetadataFormats>
   <metadataFormat>
    <metadataPrefix>oai_dc</metadataPrefix>
    <schema>http://www.openarchives.org/OAI/2.0/oai_dc.xsd</schema>
    <metadataNamespace>http://www.openarchives.org/OAI/2.0/oai_dc/
      </metadataNamespace>
   </metadataFormat>
   <metadataFormat>
    <metadataPrefix>oai_marc</metadataPrefix>
    <schema>http://www.openarchives.org/OAI/1.1/oai_marc.xsd</schema>
    <metadataNamespace>http://www.openarchives.org/OAI/1.1/oai_marc
      </metadataNamespace>
   </metadataFormat>
  </ListMetadataFormats>
</OAI-PMH>
```

Request

List the metadata formats that can be disseminated for the unique identifier `oai:lcoa1.loc.gov:loc.rbc/rbpe.00000111` in the repository `http://memory.loc.gov/cgi-bin/oai`. The identifier, however, does not exist and therefore, the response contains an `error` element and no metadataFormat container. [URL shown without encoding for better readability].

```
http://memory.loc.gov/cgi-bin/oai?
       verb=ListMetadataFormats&identifier=oai:lcoa1.loc.gov:loc.rbc/rbpe.00000111
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
         http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
  <responseDate>2002-06-08T15:19:13Z</responseDate>
  <request verb="ListMetadataFormats"
           identifier="oai:lcoa1.loc.gov:loc.rbc/rbpe.00000111">
           http://memory.loc.gov/cgi-bin/oai</request>
  <error code="idDoesNotExist">oai:lcoa1.loc.gov:loc.rbc/rbpe.00000111 has the
       structure of a valid LOC identifier, but it maps to no known
       item</error>
</OAI-PMH>
```

4.5 ListRecords

Summary and Usage Notes

This verb is used to harvest records from a repository. Optional arguments permit selective harvesting of records based on set membership and/or datestamp. Depending on the repository's support for deletions, a returned header **may** have a `status` attribute of "deleted" if a record matching the arguments specified in the request has been deleted. No metadata will be present for records with deleted status.

Arguments

- **`from`** an *optional* argument with a UTCdatetime value, which specifies a lower bound for datestamp-based selective harvesting.
- **`until`** an *optional* argument with a UTCdatetime value, which specifies a upper bound for datestamp-based selective harvesting.
- **`set`** an *optional* argument with a `setSpec` value , which specifies set criteria for selective harvesting.
- **`resumptionToken`** an *exclusive* argument with a value that is the flow control token returned by a previous `ListRecords` request that issued an incomplete list.
- **`metadataPrefix`** a *required* argument (unless the exclusive argument `resumptionToken` is used) that specifies the `metadataPrefix` of the format that should be included in the metadata part of the returned records. Records should be included only for items from which the metadata format matching the `metadataPrefix` can be disseminated. The metadata formats supported by a repository and for a particular item can be retrieved using the `ListMetadataFormats` request.

Error and Exception Conditions

- **`badArgument`** - The request includes illegal arguments or is missing required arguments.
- **`badResumptionToken`** - The value of the `resumptionToken` argument is invalid or expired.
- **`cannotDisseminateFormat`** - The value of the `metadataPrefix` argument is not supported by the repository.
- **`noRecordsMatch`** - The combination of the values of the `from`, `until`, `set` and `metadataPrefix` arguments results in an empty list.
- **`noSetHierarchy`** - The repository does not support sets.

Examples

Request

List the records expressed in `oai_rfc1807` metadata format, that have been added or modified since January 15, 1998 in the `hep` subset of the `physics` set [URL shown without encoding for better readability].

```
http://an.oa.org/OAI-script?
       verb=ListRecords&from=1998-01-15&set=physics:hep&metadataPrefix=oai_rfc1807
```

Response

Two records are returned:

- The first record is expressed in the `oai_rfc1807` metadata. This record also has an `about` part, and the item from which it was disseminated belongs to two sets (`physics:hep` and `math`).
- The second has a header with a `status="deleted"` attribute (and therefore no metadata part).

Note: The reply only includes records for those items from which metadata in `oai_rfc1807` can be disseminated. No records are returned for those items that fit the `from`, `until`, and `set` arguments but from which the specified format can not be disseminated.

```
<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
```

```
        http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
 <responseDate>2002-06-01T19:20:30Z</responseDate>
 <request verb="ListRecords" from="1998-01-15"
          set="physics:hep"
          metadataPrefix="oai_rfc1807">
          http://an.oa.org/OAI-script</request>
 <ListRecords>
  <record>
    <header>
      <identifier>oai:arXiv.org:hep-th/9901001</identifier>
      <datestamp>1999-12-25</datestamp>
      <setSpec>physics:hep</setSpec>
      <setSpec>math</setSpec>
    </header>
    <metadata>
     <rfc1807 xmlns=
        "http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1807.txt"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation=
       "http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1807.txt
        http://www.openarchives.org/OAI/1.1/rfc1807.xsd">
        <bib-version>v2</bib-version>
        <id>hep-th/9901001</id>
        <entry>January 1, 1999</entry>
        <title>Investigations of Radioactivity</title>
        <author>Ernest Rutherford</author>
        <date>March 30, 1999</date>
     </rfc1807>
    </metadata>
    <about>
      <oai_dc:dc
          xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"
          xmlns:dc="http://purl.org/dc/elements/1.1/"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/
          http://www.openarchives.org/OAI/2.0/oai_dc.xsd">
        <dc:publisher>Los Alamos arXiv</dc:publisher>
        <dc:rights>Metadata may be used without restrictions as long as
           the oai identifier remains attached to it.</dc:rights>
      </oai_dc:dc>
    </about>
  </record>
  <record>
    <header status="deleted">
      <identifier>oai:arXiv.org:hep-th/9901007</identifier>
      <datestamp>1999-12-21</datestamp>
    </header>
  </record>
 </ListRecords>
</OAI-PMH>
```

Request

Request records in the `oai_dc` metadata format, modified or added between 2:15pm and 2:20pm UTC on May 1st 2002. [URL shown without encoding for better readability].

```
http://www.perseus.tufts.edu/cgi-b:in/pdataprov?
       verb=ListRecords&from=2002-05-01T14:15:00Z&until=2002-05-01T14:20:00Z&
       metadataPrefix=oai_dc
```

Response

Two records are returned. The second one has a `provenance` container in its `about` element, giving an insight in its chain of provenance.

```
<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
         http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
 <responseDate>2002-06-01T19:20:30Z</responseDate>
 <request verb="ListRecords" from="2002-05-01T14:15:00Z"
          until="2002-05-01T14:20:00Z" metadataPrefix="oai_dc">
          http://www.perseus.tufts.edu/cgi-bin/pdataprov</request>
 <ListRecords>
  <record>
    <header>
      <identifier>oai:perseus:Perseus:text:1999.02.0084</identifier>
      <datestamp>2002-05-01T14:16:12Z</datestamp>
    </header>
    <metadata>
      <oai_dc:dc
          xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"
          xmlns:dc="http://purl.org/dc/elements/1.1/"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/
          http://www.openarchives.org/OAI/2.0/oai_dc.xsd">
        <dc:title>Opera Minora</dc:title>
        <dc:creator>Cornelius Tacitus</dc:creator>
        <dc:type>text</dc:type>
        <dc:source>Opera Minora. Cornelius Tacitus. Henry Furneaux.
         Clarendon Press. Oxford. 1900.</dc:source>
        <dc:language>latin</dc:language>
        <dc:identifier>http://www.perseus.tufts.edu/cgi-bin/ptext?
          doc=Perseus:text:1999.02.0084</dc:identifier>
      </oai_dc:dc>
    </metadata>
  </record>
  <record>
    <header>
      <identifier>oai:perseus:Perseus:text:1999.02.0083</identifier>
      <datestamp>2002-05-01T14:20:55Z</datestamp>
    </header>
    <metadata>
      <oai_dc:dc
          xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"
          xmlns:dc="http://purl.org/dc/elements/1.1/"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/
          http://www.openarchives.org/OAI/2.0/oai_dc.xsd">
        <dc:title>Germany and its Tribes</dc:title>
        <dc:creator>Tacitus</dc:creator>
        <dc:type>text</dc:type>
        <dc:source>Complete Works of Tacitus. Tacitus. Alfred John Church.
         William Jackson Brodribb. Lisa Cerrato. edited for Perseus.
         New York: Random House, Inc. Random House, Inc. reprinted 1942.
          </dc:source>
        <dc:language>english</dc:language>
        <dc:identifier>http://www.perseus.tufts.edu/cgi-bin/ptext?
         doc=Perseus:text:1999.02.0083</dc:identifier>
      </oai_dc:dc>
      </metadata>
    <about>
      <provenance
       xmlns="http://www.openarchives.org/OAI/2.0/provenance"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/provenance
       http://www.openarchives.org/OAI/2.0/provenance.xsd">
       <originDescription harvestDate="2002-01-01T11:10:01Z" altered="true">
        <baseURL>http://some.oa.org</baseURL>
        <identifier>oai:r2.org:klik001</identifier>
        <datestamp>2001-01-01</datestamp>
```

```
        <metadataNamespace>http://www.openarchives.org/OAI/2.0/oai_dc/</metadataNamespace>
        </originDescription>
      </provenance>
    </about>
  </record>
 </ListRecords>
</OAI-PMH>
```

Request

Request records in the the `oai_marc` metadata format, modified or added between 2:00am and 3:00am UTC on June 1st 2002. The specified granularity is not supported by the repository and therefore, an `error` with `code` attribute of `badArgument` is returned. [URL shown without encoding for better readability].

```
http://memory.loc.gov/cgi-bin/oai?
       verb=ListRecords&from=2002-06-01T02:00:00Z&until=2002-06-01T03:00:00Z&metadataPrefix=oai_marc
```

Response

```
<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
         http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
  <responseDate>2002-06-01T19:20:30Z</responseDate>
  <request verb="ListRecords" from="2002-06-01T02:00:00Z"
           until="2002-06-01T03:020:00Z"
           metadataPrefix="oai_marc">
           http://memory.loc.gov/cgi-bin/oai</request>
  <error code="badArgument"/>
</OAI-PMH>
```

4.6 ListSets

Summary and Usage Notes

This verb is used to retrieve the set structure of a repository, useful for selective harvesting.

Arguments

- **`resumptionToken`** an *exclusive* argument with a value that is the flow control token returned by a previous `ListSets` request that issued an incomplete list.

Error and Exception Conditions

- **`badArgument`** - The request includes illegal arguments or is missing required arguments.
- **`badResumptionToken`** - The value of the `resumptionToken` argument is invalid or expired.
- **`noSetHierarchy`** - The repository does not support sets.

Examples

Request

```
http://an.oa.org/OAI-script?
       verb=ListSets
```

Response

The following response indicates a set hierarchy with two top level sets with respective setSpec `music` and `video`. The `music` set has two subsets, with setSpec `music:(muzak)` and `music:(elec)`. The subsets identified by setSpec `music:(elec)`, has a `setDescription` element which holds a Dublin Core container, used to describe its contents.

```
<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
         http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
 <responseDate>2002-08-11T07:21:33Z</responseDate>
 <request verb="ListSets">http://an.oa.org/OAI-script</request>
 <ListSets>
  <set>
    <setSpec>music</setSpec>
    <setName>Music collection</setName>
  </set>
  <set>
    <setSpec>music:(muzak)</setSpec>
    <setName>Muzak collection</setName>
  </set>
  <set>
    <setSpec>music:(elec)</setSpec>
    <setName>Electronic Music Collection</setName>
    <setDescription>
      <oai_dc:dc
          xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"
          xmlns:dc="http://purl.org/dc/elements/1.1/"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/
          http://www.openarchives.org/OAI/2.0/oai_dc.xsd">
          <dc:description>This set contains metadata describing
             electronic music recordings made during the 1950ies
             </dc:description>
       </oai_dc:dc>
    </setDescription>
   </set>
   <set>
    <setSpec>video</setSpec>
    <setName>Video Collection</setName>
   </set>
 </ListSets>
</OAI-PMH>
```

Request

```
http://purl.org/alcme/etdcat/servlet/OAIHandler?
       verb=ListSets
```

Response

The response shows that the repository does not have a set hierarchy.

```
<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
        xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
        http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
  <responseDate>2001-06-01T19:20:30Z</responseDate>
  <request verb="ListSets">
          http://purl.org/alcme/etdcat/servlet/OAIHandler</request>
  <error code="noSetHierarchy">This repository does not
        support sets</error>
</OAI-PMH>
```

5. Dublin Core

The following table shows the XML Schema for Dublin Core without qualification, which is associated with the reserved `metadataPrefix oai_dc` in the OAI-PMH. All examples in this document that include Dublin Core metadata, validate against this XML schema. Schema for other metadata formats are provided in the accompanying Implementation Guidelines document.

A XML schema for validating Unqualified Dublin Core metadata associated with the reserved `oai_dc metadataPrefix`

```
<schema targetNamespace="http://www.openarchives.org/OAI/2.0/oai_dc/"
        xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"
        xmlns:dc="http://purl.org/dc/elements/1.1/"
        xmlns="http://www.w3.org/2001/XMLSchema"
        elementFormDefault="qualified" attributeFormDefault="unqualified">

<annotation>
  <documentation>
      XML Schema 2002-03-18 by Pete Johnston.
      Adjusted for usage in the OAI-PMH.
      Schema imports the Dublin Core elements from the DCMI schema for unqualified Dublin Core.
      2002-12-19 updated to use simpledc20021212.xsd (instead of simpledc20020312.xsd)
  </documentation>
</annotation>

<import namespace="http://purl.org/dc/elements/1.1/"
        schemaLocation="http://dublincore.org/schemas/xmls/simpledc20021212.xsd"/>

<element name="dc" type="oai_dc:oai_dcType"/>

<complexType name="oai_dcType">
  <choice minOccurs="0" maxOccurs="unbounded">
    <element ref="dc:title"/>
    <element ref="dc:creator"/>
    <element ref="dc:subject"/>
    <element ref="dc:description"/>
    <element ref="dc:publisher"/>
    <element ref="dc:contributor"/>
    <element ref="dc:date"/>
    <element ref="dc:type"/>
    <element ref="dc:format"/>
    <element ref="dc:identifier"/>
    <element ref="dc:source"/>
    <element ref="dc:language"/>
    <element ref="dc:relation"/>
    <element ref="dc:coverage"/>
    <element ref="dc:rights"/>
  </choice>
</complexType>
```

```
</schema>
```

This Schema is available at http://www.openarchives.org/OAI/2.0/oai_dc.xsd

Examples

```
<?xml version="1.0" encoding="UTF-8"?>
<oai_dc:dc
    xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/
    http://www.openarchives.org/OAI/2.0/oai_dc.xsd">
  <dc:title xml:lang="en">The Cornell Law Quarterly</dc:title>
  <dc:date>1915-1916</dc:date>
  <dc:identifier>http://heinonline.org/HeinOnline/show.pl?
      handle=hein.journals/clqv1%26id=1%26size=4</dc:identifier>
  <dc:rights>Available by Subscription.
             See http://www.wshein.com</dc:rights>
</oai_dc:dc>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<oai_dc:dc
    xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/
    http://www.openarchives.org/OAI/2.0/oai_dc.xsd">
  <dc:title xml:lang="en">Grassmann's space analysis</dc:title>
  <dc:creator>Hyde, E. W. (Edward Wyllys)</dc:creator>
  <dc:subject>LCSH:Ausdehnungslehre; LCCN QA205.H99</dc:subject>
  <dc:publisher>J. Wiley &amp;amp; Sons</dc:publisher>
  <dc:date>Created: 1906; Available: 1991</dc:date>
  <dc:type>text</dc:type>
  <dc:identifier>http://resolver.library.cornell.edu/math/1796949
     </dc:identifier>
  <dc:language>english</dc:language>
  <dc:rights xml:lang="en">Public Domain</dc:rights>
</oai_dc:dc>
```

6. Implementation Guidelines

Some passages in this document refer to the existence and goals of the accompanying Implementation Guidelines document.

Acknowledgements

Support for the development of the OAI-PMH and for other Open Archives Initiative activities comes from the Digital Library Federation, the Coalition for Networked Information, and from the National Science Foundation through Grant No. IIS-9817416.

This document is based on the deliberations of the OAI Technical Committee: Caroline Arms (Library of Congress), Thomas Baron (CERN), Steven Bird (University of Pennsylvania), Les Carr (University of Southampton), Tim Cole (University of Illinois at Urbana Champaign), Thomas Krichel (Long Island University), Carl Lagoze (Cornell University), Michael Nelson (NASA), Andy Powell (UKOLN & University of Bath), Mogens Sandfaer (Danmarks Tekniske Videncenter), Hussein Suleman (Virginia Tech), Robert Tansley (HP), Herbert Van de Sompel (Los Alamos National Laboratory), Simeon Warner (Cornell University), Muhammad Zubair (Old Dominion University) and Jeff Young (OCLC).

Many thanks to all involved in alpha-testing of version 2.0 of the OAI-PMH. In addition to the above: Tim Brody (University of Southampton), Irena Dijour (Ex Libris), Naomi Dushay (Cornell University), Susanne Dobratz (Humboldt Universität zu Berlin), Curtis Fornadley (UCLA), Christopher Gutteridge (University of Southampton), Alan Kent (InQuirion Pty Ltd & RMIT University), David Letts (The British Library), Xiaoming Liu (Old Dominion University), Jon Phipps (Cornell University) and Francois Schiettecatte (FS Consulting Inc).

Special thanks to Pete Johnston (UKOLN & University of Bath) and Andy Powell (UKOLN & University of Bath) for work on the Dublin Core schema, and to Donna Bergmark (Cornell University) for work on the OAI validation and registration service.

Many thanks to everyone involved in the compilation and alpha-testing of version 1.0 and 1.1 of the OAI-PMH, and to all of you using this protocol.

Document History

2015-01-08: Add explicit CC BY-SA license, HTML fixes. No change to protocol.
2008-12-07: Fix links to previous versions.
2008-12-02: Spell checked after all these years and several errors corrected. No change of meaning. Added links to previous versions.
2004-10-12: Changed wording and schema definition for characters allowed in `setSpec` and `metadataPrefix` to agree.
2004-09-15: Added section 2.5.1. Corrected section 2.6. Corrected second example in section 5. Changed schema to defined a type for `protocolVersion` and to enforce use of `Z` notation for UTC datetime.
2003-02-21: Changed identifiers in the examples so that they conform to version 2.0 of the `oai-identifier` specification.
2002-12-19: Updated `oai_dc` schema to use revised Dublin Core schema `simpledc20021212.xsd`. Corrected `provenance` blocks in examples (sections 2.5 and 4.5).
2002-06-14: Release of OAI-PMH version 2.0.
2002-05-02: Release of beta version of OAI-PMH version 2.0.
2002-05-06: Release of alpha-4 version of OAI-PMH version 2.0. Changed document to reflect association of datestamps and deleted status with records as opposed to items. Changed `requestURL` to `request`. Changed schema location of `oai-identifier` and `oai_dc` schema. Changed validation of `about`, `metadata`, `description` and `setDescription` to strict.
2002-04-07: Changed document to reflect the usage of a single schema to validate all OAI-PMH responses.
2002-03-30: Release of alpha two version of OAI-PMH version 2.0.
2002-03-01: Release of alpha version of OAI-PMH version 2.0