



# Publishing Based on Data Provider

Version 16 and later

**Please note: Implementation of the following OAI tools requires an additional license agreement with Ex Libris. To learn more about licensing this functionality, please contact your regional sales representative for details and pricing.**

## CONFIDENTIAL INFORMATION

The information herein is the property of Ex Libris Ltd. or its affiliates and any misuse or abuse will result in economic loss. DO NOT COPY UNLESS YOU HAVE BEEN GIVEN SPECIFIC WRITTEN AUTHORIZATION FROM EX LIBRIS LTD.

This document is provided for limited and restricted purposes in accordance with a binding contract with Ex Libris Ltd. or an affiliate. The information herein includes trade secrets and is confidential.

## DISCLAIMER

The information in this document will be subject to periodic change and updating. Please confirm that you have the most current documentation. There are no warranties of any kind, express or implied, provided in this documentation, other than those expressly agreed upon in the applicable Ex Libris contract. This information is provided AS IS. Unless otherwise agreed, Ex Libris shall not be liable for any damages for use of this document, including, without limitation, consequential, punitive, indirect or direct damages.

Any references in this document to third-party material (including third-party Web sites) are provided for convenience only and do not in any manner serve as an endorsement of that third-party material or those Web sites. The third-party materials are not part of the materials for this Ex Libris product and Ex Libris has no liability for such materials.

## TRADEMARKS

"Ex Libris," the Ex Libris bridge, Primo, Aleph, Alephino, Voyager, SFX, MetaLib, Verde, DigiTool, Preservation, URM, Voyager, ENCompass, Endeavor eZConnect, WebVoyage, Citation Server, LinkFinder and LinkFinder Plus, and other marks are trademarks or registered trademarks of Ex Libris Ltd. or its affiliates.

The absence of a name or logo in this list does not constitute a waiver of any and all intellectual property rights that Ex Libris Ltd. or its affiliates have established in any of its products, features, or service names or logos.

Trademarks of various third-party products, which may include the following, are referenced in this documentation. Ex Libris does not claim any rights in these trademarks. Use of these marks does not imply endorsement by Ex Libris of these third-party products, or endorsement by these third parties of Ex Libris products.

Oracle is a registered trademark of Oracle Corporation.

UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Ltd.

Microsoft, the Microsoft logo, MS, MS-DOS, Microsoft PowerPoint, Visual Basic, Visual C++, Win32,

Microsoft Windows, the Windows logo, Microsoft Notepad, Microsoft Windows Explorer, Microsoft Internet Explorer, and Windows NT are registered trademarks and ActiveX is a trademark of the Microsoft Corporation in the United States and/or other countries.

Unicode and the Unicode logo are registered trademarks of Unicode, Inc.

Google is a registered trademark of Google, Inc.

Copyright Ex Libris Limited, 2014. All rights reserved.

Document released: June 2, 2014

Web address: <http://www.exlibrisgroup.com>

## Table of Contents

<b>OVERVIEW</b> .....	<b>4</b>
<b>ARCHITECTURE</b> .....	<b>4</b>
<b>PUBLISHING PLATFORM CONFIGURATION</b> .....	<b>6</b>
Converting to the Required Format .....	6
<b>OAI DATA PROVIDER CONFIGURATION</b> .....	<b>8</b>
<b>HOW TO RUN THE OAI DATA PROVIDER</b> .....	<b>11</b>
<b>PROTOCOL REQUESTS AND RESPONSES</b> .....	<b>11</b>
GetRecord .....	11
Identify .....	15
ListIdentifiers .....	16
ListMetadataFormats .....	18
ListRecords .....	19
ListSets .....	20

## Overview

The OAI (Open Archive Initiative) Data Provider implements the OAI Protocol Version 2.0. It supports 6 OAI-PMH requests specified by the OAI protocol.

For more information about the protocol, refer to the OAI Web site:

<http://www.openarchives.org/>.

This document describes the configuration and usage of the OAI Data Provider in ALEPH systems that implement Publishing. Refer to the ALEPH Publishing documentation for more information about Publishing. Refer to *OAI Data Provider* for more information on the non Publishing-based OAI Data Provider.

## Architecture

Publishing-based ALEPH OAI Data Provider exposes data prepared by the ALEPH Publishing process.

The ALEPH OAI Data Provider itself is a part of the ALEPH `www_server`. It works in request/response mode. The request is a URL request built according to OAI protocol. The response is in XML form built according to OAI protocol.

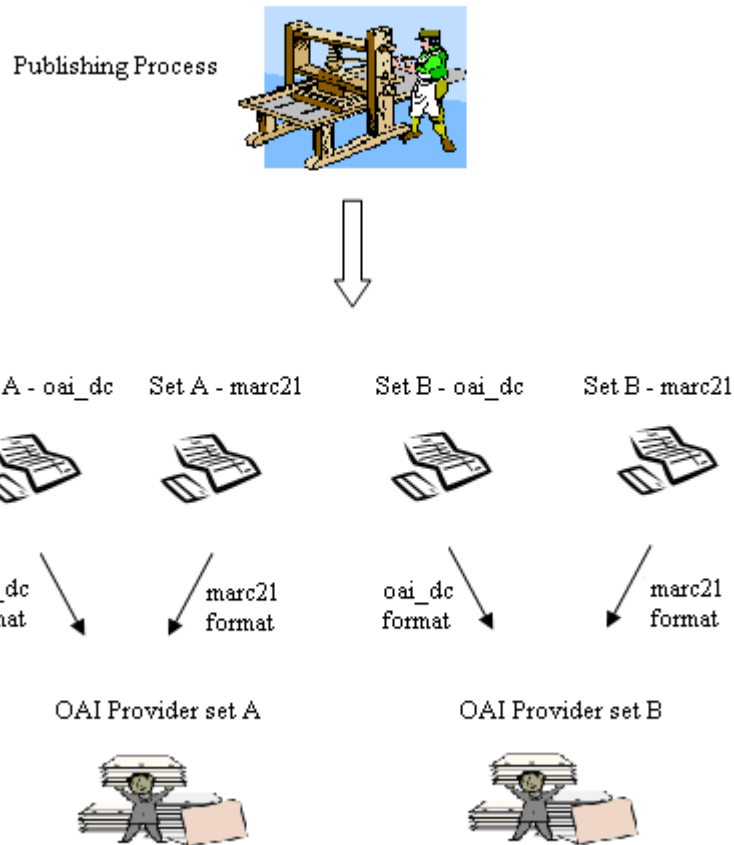
ALEPH OAI Data Provider has flat sets organization. It supports two formats

- **marc21** (<http://www.loc.gov/standards/marcxml/schema/MARC21slim.xsd>)
- **oai\_dc** ([http://www.openarchives.org/OAI/2.0/oai\\_dc.xsd](http://www.openarchives.org/OAI/2.0/oai_dc.xsd))

Information about deleted records is reported in transient mode.

ALEPH OAI Data Provider supports YYYY-MM-DDT:hh:mm:ssZ timestamps granularity. Consequent to the OAI protocol requirements the timestamps are in UTC form.

Publishing-based OAI Data Provider sets are based on published sets. The following diagram may be used to demonstrate the relationship between the OAI Data Provider sets and the published sets:



To expose ALEPH data via the Publishing-based OAI Data Provider the following actions must be performed:

1. ALEPH Publishing process must be performed.

As a rule, each set that is to be exposed by the Publishing-based OAI Data Provider must be published in each of the required formats. For example, if a set EXAM is to be provided in oai\_dc format and in marc21 format, two sets must be published, one in OAI\_DC\_XML format and one in OAI\_MARC21\_XML. Both sets publish the same data but in different formats.

The following is therefore true:

- At least one set in OAI\_DC\_XML format must be published. This set includes data from all of the sets that are to be exposed in oai\_dc format. The set is the OAI Data Provider's default set for oai\_dc format requests.
- A set in OAI\_MARC21\_XML format must be created if there is a need to expose data in marc21 format. This set includes data from all of the sets that are to be exposed in marc21 format. This set is the default set for marc21 format requests.
- If there is a need to divide data by some logical units, more sets may be created. Each set must be published in all required formats.

**Note:**

- Published Records with OAI\_DC\_XML format are exposed by OAI Data Provider as oai\_dc records. Records with OAI\_MARC21\_XML format are exposed by OAI Data Provider as marc21 records.
  - Set names created by the publishing process must not contain colons (:).
2. ALEPH OAI Data Provider must be configured. Each set exposed by the OAI Data Provider should be defined with the following:
- **An external set name** – This is the name by which external applications that harvest the OAI Data Provider’s information identify the set.
  - **Internal set names** – These names are the published sets (see above) from which the OAI set information is derived. Each external set is related to as many internal sets as there are formats in which it may be exposed.
    - If the set records are to be exposed in oai\_dc format then the external set should be related to a Publishing set with OAI\_DC\_XML format.
    - If set records are to be exposed also in marc21 format then the external set should be related to a Publishing set with OAI\_MARC21\_XML format.

Refer to the *OAI Data Provider Configuration* chapter for detailed information on configuring the OAI Data Provider.

3. ALEPH www\_server must be started.

## Publishing Platform Configuration

As described in the *Architecture* chapter, the first step for configuring a publishing based OAI Data Provider is to run a Publishing process. In order to make the oai\_dc format available, a set must be published in OAI\_DC\_XML format. Likewise, in order to make the marc21 format available, a set must be published in OAI\_MARC21\_XML format.

This chapter describes the required configuration for a successful publishing process to result in an OAI\_DC\_XML or OAI\_MARC21\_XML formatted set.

## Converting to the Required Format

If the catalog is not in the same format as the format that is required for publishing, a conversion is required.

The conversion is done by setting 4 tables:

- **tab\_publish** – This table contains the specifications for extracting ALEPH records for publishing purposes. Column 4 of the tab\_publish row in which the set is defined must be set with a fix routine that is defined in tab\_fix for the required conversion. If you need to use expand-routines for the published records, you can also add a section to tab\_expand with the section name from column 4 of tab\_publish.

- **tab\_fix** – This table contains a list of fix\_doc programs. The table must have a fix routine that makes the required conversion. The fix routine uses the fix\_doc\_convit program (column 2 of tab\_fix) with the parameter (column 3 of tab\_fix) set to the table in which the conversion is defined.
- **tab\_fix\_xxx2yyy** – A table that defines the required conversion from a format to a format. For example, tab\_fix\_mab2usm defines conversions from MAB to MARC, and tab\_fix\_usm2dc defines conversions from MARC to DC. The conversion tables have the following structure:

```
COL 1. 5; ALPHA_NUM, UPPER; #;
      Source Tag & indicator;
COL 2. 1; ALPHA_NUM, LOWER; ;
      Source Subfield;
COL 3. 5; ALPHA_NUM, UPPER; #;
      Target Tag & indicator;
COL 4. 1; ALPHA_NUM, LOWER; ;
      Target Subfield;
COL 5. 100; ALPHA_NUM, LOWER; _;
      Program name with arguments;
```

- If the required format is OAI\_DC\_XML then the output of the conversion in step 3 is further converted by the tab\_dc table.

For example:

If a set is required for publishing in OAI\_DC\_XML format, column 4 of the tab\_publish row must be defined with a fix routine that converts the MARC catalog to OAI\_DC\_XML format. The code that must be configured in this column must be the code of a tab\_fix routine that translates MARC to DC.

In tab\_publish:

```
!      1          2          3 4          5
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
OAI-USM-DC                                N USMDC OAI_DC_XML
```

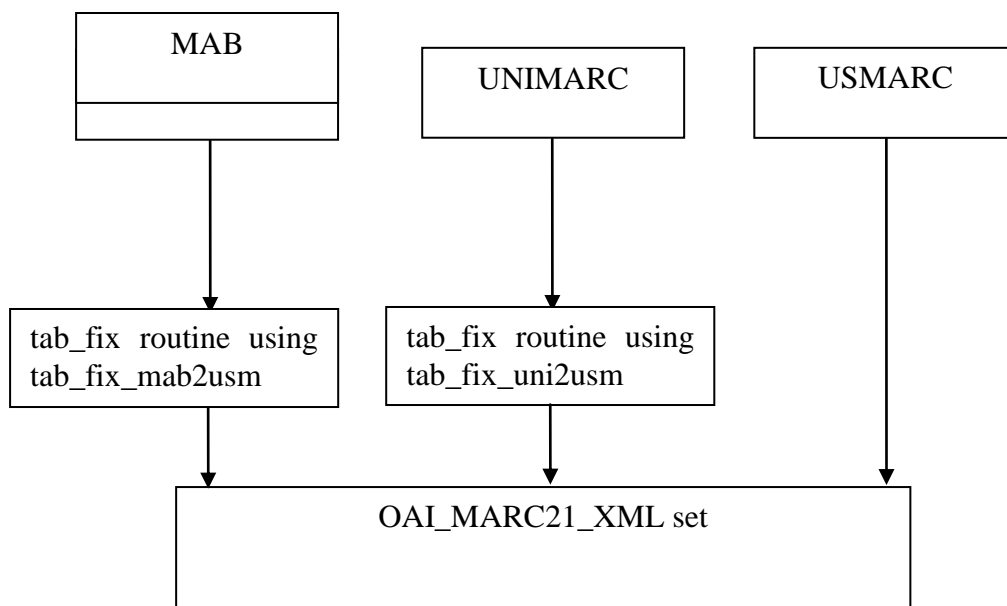
In tab\_fix:

```
1          2          3
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
USMDC fix_doc_convit                       FILE=tab_fix_usm2dc
```

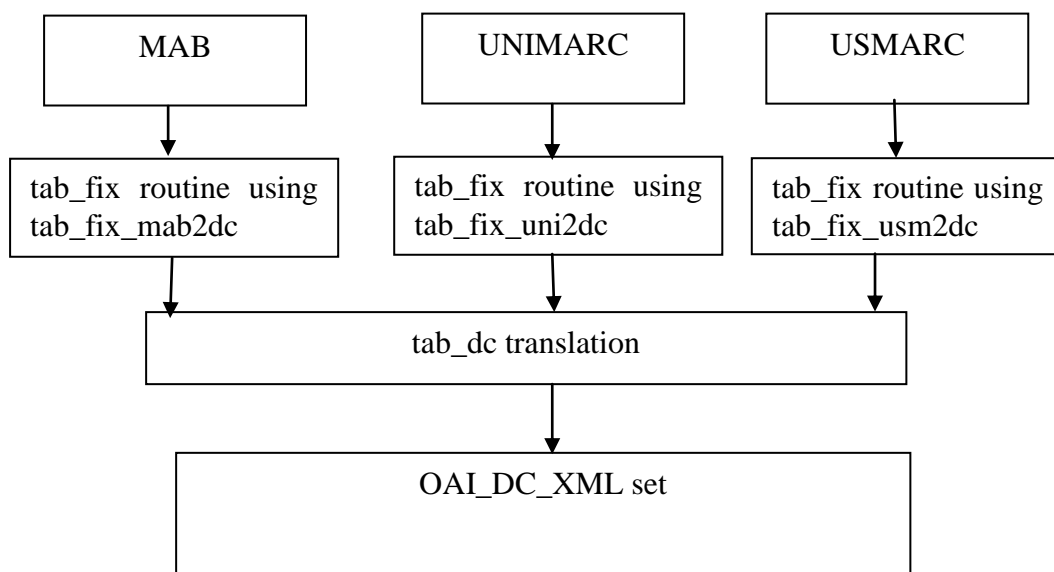
The fix\_doc\_convit routine fixes the document using the table that is defined in the parameters column (column 3). In this example, the tab\_fix\_usm2dc table defines conversions from MARC fields to DC fields.

When converting to DC, an additional step is performed, converting the output of the tab\_fix conversion to DC using the tab\_dc table.

In other words, when creating a OAI\_MARC21\_XML set, the translate mechanism may be described as follows:



When creating a OAI\_DC\_XML set, the translate mechanism may be described as follows:



## OAI Data Provider Configuration

Publishing-based ALEPH OAI Data Provider has a single configuration file in XML form:

```
./alephe/tab/oai/oaipubconf.xml
```

### oaipubconf.xml

The `oaipubconf.xml` file contains the following elements:

- `<oairoot>` – The root element

- **<set>** – A repetitive element which contains the following sub-elements:
  - **<setSpec>** – Contains set specs which occurs in OAI requests/responses. This element is mandatory.
  - **<setName>** – Contains the set name. This element is mandatory
  - **<internalSet>** – Contains the name of the Publishing set from which the records are taken, followed by a colon, followed by format oai\_dc or marc21.

For each set at most 2 <internalSet> elements can be defined – one for oai\_dc records and one for marc21 records. As OAI Data Provider must support oai\_dc format, for each set a <internalSet> element for oai\_dc records must be defined.

For example, if the OAI Data Provider set EXAM is exposed in oai\_dc format and in marc21 format then:

- There are two published sets – EXAM\_DC and EXAM\_21. Both have the same records, with the only difference being that EXAM\_DC is in OAI\_DC\_XML format and EXAM\_21 in OAI\_MARC21\_XML format.
- The <set> tag is configured as follows:

```

<set>
  <setSpec>EXAM</setSpec>
  <setName>Example Set</setName>
  <internalSet>EXAM_DC:oai_dc</internalSet>
  <internalSet>EXAM_21:marc21</internalSet>
</set>

```

**Note:** The first <set> element listed in the file must contain the default set (a set containing all the records exposed by OAI Data Provider).

- **<repositoryName>** – Name of the repository. This element is mandatory.
- **<baseURL>** – URL used in OAI requests. It must have the following form:
  - *http://<ALEPH web server host>:<ALEPH apache port>/OAI.*
 This element is mandatory.
- **<adminEmail>** – Email of the repository administrator. This element is mandatory.
- **<description>** – Information about the repository. This element is mandatory. It must contain an <oai-identifier> element with a description of the item identifiers naming policy. It may also contain sub-elements with additional information about the repository.

The <oai-identifier> element must have the following format:

```

<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>il-aleph05</repositoryIdentifier>
  <delimiter>:</delimiter>
  <sampleIdentifier>oai:il-aleph05:USM01-000000001</sampleIdentifier>
</oai-identifier>
</description>

```

The <repositoryIdentifier> element must contain the repository ID used as part of OAI identifiers. It must start with a Latin letter.

The <sampleIdentifier> element must contain a sample identifier.

The syntax must be:

```
oai:<repository id>:<physical library>-<doc number>
```

Below is a sample oaipubconf.xml file:

```

<?xml version="1.0" encoding="UTF-8"?>
<oirroot>
<set>
  <setSpec>OAI-SET</setSpec>
  <setName>OAI SET</setName>
  <internalSet>OAISETD:oai_dc</internalSet>
  <internalSet>OAISETM:marc21</internalSet>
</set>
<set>
  <setSpec>HISTORY</setSpec>
  <setName>History</setName>
  <internalSet>HISTORYD:oai_dc</internalSet>
  <internalSet>HISTORYM:marc21</internalSet>
</set>
<set>
  <setSpec>SCIENCE</setSpec>
  <setName>Science</setName>
  <internalSet>SCIENCED:oai_dc</internalSet>
  <internalSet>SCIENCEM:marc21</internalSet>
</set>
<repositoryName>ExLibris Repository</repositoryName>
<baseURL>http://il-aleph05:8995/OAI</baseURL>
<adminEmail>Irina.Dijour@exlibris.co.il</adminEmail>
<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>il-aleph05</repositoryIdentifier>
  <delimiter>:</delimiter>
  <sampleIdentifier>
    oai:il-aleph05:USM01-000000001</sampleIdentifier>

```

```
</oai-identifier>
</description>
</oairoot>
```

This example defines three sets: OAI-SET, SCIENCE, and HISTORY. All the sets expose data in oai\_dc and marc21 format. The set OAI-SET is the default set. It contains all the records contained in sets SCIENCE and HISTORY. It can also contain additional records.

- When the user asks for the OAI-SET set (or does not specify a set), and the request is for the oai\_dc format - records from the Publishing set OAISETD are exposed.
- When the user asks for the OAI-SET set (or does not specify set), and the request is for the marc21 format - records from the Publishing set OAISETM are exposed.
- When the user asks for the HISTORY set, and the request is for the oai\_dc format - records from Publishing set HISTORYD are exposed.
- When the user asks for HISTORY set, and the request is for the marc21 format - records from Publishing set HISTORYM are exposed.

## How to Run the OAI Data Provider

The OAI data provider is part of the Web server.

To access OAI data, type the value of <baseURL> from oaipubconf.xml followed by '?' followed by OAI verb and OAI verb parameters in the address bar of your Web browser

For example:

<http://il-aleph05:8995/OAI?verb=ListSets>

**Note:** if the OAI configuration in ./alephe/tab/oai is changed, the Web server must be restarted.

## Protocol Requests and Responses

### GetRecord

#### Summary

Retrieves a specified record in a specified format from the repository.

#### Arguments

- **identifier** (required) – has the form oai:<repository id>:<library>-<doc number> (for example, oai:il-aleph05:USM01-000000002)
- **metadataPrefix** (required) – supported values – oai\_dc, marc21

#### Request Example

http://il-aleph05:8995/OAI?verb=GetRecord&identifier=oai:il-ALEPH05:USM01-000000002&metadataPrefix=marc21

## Flow

The OAI Data Provider uses the first <set> element listed in oaipubconf.xml as default set (e.g. OAI-SET). It takes the required Publishing set name from the <internal\_set> sub-element that is marked with the requested format. It then retrieves a record from Z00P that matches to the following values:

- Z00P-SET is equal to the publishing set defined by oaipubconf.xml (e.g. OAISETM)
- Z00P-LIBRARY is equal to the library from the requested identifier (e.g. USM01)
- Z00P-DOC-NUMBER is equal to doc number from the requested identifier (e.g. 000000002)

If the record is deleted (Z00P-STATUS is equal to DELETED) then the status attribute of the header is set to DELETED. Otherwise the XML record from Z00P is added to response.

## Response

A sample response is found below.

Below is an explanation of the elements in the response:

- **<GetRecord>** – Contains <record> element
- **<record>** – Contains <header> element. If the record is not deleted it contains also <metadata> element
- **<header>** – Contains <identifier>, <timestamp> and <setSpec> elements. If the record is deleted then the status attribute of the <header> element is set to deleted.
- **<identifier>** – Contains the full record identifier from the request
- **<timestamp>** – Contains the date and time of last record update in UTC format, obtained from Z00P-TIMESTAMP value of the retrieved record
- **<setSpec>** – Contains set spec value that is equal to <setSpec> value of the current set in oaipubconf.xml
- **<metadata>** – Contains the XML record from the retrieved Z00P

## Response Example:

```
<?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>2007-12-25T14:07:39Z</responseDate>
  <request      verb="GetRecord"      identifier="oai:il-aleph05:USM01-000000508"
metadataPrefix="marc21">http://il-aleph05:8995/OAI</request>
- <GetRecord>
- <record>
- <header>
  <identifier>oai:il-aleph05:USM01-000000508</identifier>
  <timestamp>2007-12-17T08:37:25Z</timestamp>
  <setSpec>OAI-SET</setSpec>
  </header>
- <metadata>
-           <marc:record                               xmlns:marc="http://www.loc.gov/MARC21/slim"
```

```

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.loc.gov/MARC21/slim
http://www.loc.gov/standards/marcxml/schema/MARC21slim.xsd">
  <marc:leader>02151cas 2200589 4500</marc:leader>
  <marc:controlfield tag="001">000000508-8</marc:controlfield>
  <marc:controlfield tag="005">20031108093029.0</marc:controlfield>
  <marc:controlfield tag="008">750907c19349999pauqr1p 0uu1a0engdd</marc:controlfield>
- <marc:datafield tag="010" ind1="" ind2="">
  <marc:subfield code="a">43045299 //r542</marc:subfield>
</marc:datafield>
- <marc:datafield tag="022" ind1="" ind2="">
  <marc:subfield code="a">0027-4380</marc:subfield>
</marc:datafield>
- <marc:datafield tag="035" ind1="0" ind2="">
  <marc:subfield code="a">(OCoLC)09536003</marc:subfield>
</marc:datafield>
- <marc:datafield tag="035" ind1="0" ind2="">
  <marc:subfield code="a">(OCoLC)01605994</marc:subfield>
  <marc:subfield code="z">(OCoLC)02449172</marc:subfield>
</marc:datafield>
- <marc:datafield tag="035" ind1="0" ind2="">
  <marc:subfield code="a">(OCoLC)01605994</marc:subfield>
</marc:datafield>
- <marc:datafield tag="040" ind1="" ind2="">
  <marc:subfield code="a">MUL</marc:subfield>
  <marc:subfield code="c">MUL</marc:subfield>
  <marc:subfield code="d">NSD</marc:subfield>
  <marc:subfield code="d">DLC</marc:subfield>
  <marc:subfield code="d">NSD</marc:subfield>
  <marc:subfield code="d">DLC</marc:subfield>
  <marc:subfield code="d">m.c.</marc:subfield>
  <marc:subfield code="d">OCL</marc:subfield>
  <marc:subfield code="d">NSD</marc:subfield>
  <marc:subfield code="d">AIP</marc:subfield>
</marc:datafield>
- <marc:datafield tag="042" ind1="" ind2="">
  <marc:subfield code="a">nsdp</marc:subfield>
  <marc:subfield code="a">lc</marc:subfield>
</marc:datafield>
- <marc:datafield tag="050" ind1="0" ind2="">
  <marc:subfield code="a">ML27.U5</marc:subfield>
  <marc:subfield code="b">M695</marc:subfield>
</marc:datafield>
- <marc:datafield tag="082" ind1="" ind2="">
  <marc:subfield code="a">026.78</marc:subfield>
</marc:datafield>
- <marc:datafield tag="110" ind1="2" ind2="">
  <marc:subfield code="a">Music Library Association.</marc:subfield>
</marc:datafield>
- <marc:datafield tag="210" ind1="0" ind2="">
  <marc:subfield code="a">Notes</marc:subfield>
</marc:datafield>
- <marc:datafield tag="222" ind1="0" ind2="0">
  <marc:subfield code="a">Notes</marc:subfield>
</marc:datafield>
- <marc:datafield tag="245" ind1="1" ind2="0">
  <marc:subfield code="a">Notes.</marc:subfield>
</marc:datafield>
- <marc:datafield tag="260" ind1="0" ind2="0">
  <marc:subfield code="a">[Philadelphia, PA, etc.]</marc:subfield>

```

```

    <marc:subfield code="b">Music Library Association.</marc:subfield>
  </marc:datafield>
- <marc:datafield tag="300" ind1="" ind2="">
  <marc:subfield code="a">v.</marc:subfield>
  <marc:subfield code="b">ill., ports., facsims.</marc:subfield>
  <marc:subfield code="c">24-29 cm.</marc:subfield>
</marc:datafield>
- <marc:datafield tag="310" ind1="" ind2="">
  <marc:subfield code="a">Quarterly,</marc:subfield>
  <marc:subfield code="b">1943-</marc:subfield>
</marc:datafield>
- <marc:datafield tag="321" ind1="" ind2="">
  <marc:subfield code="a">Irregular,</marc:subfield>
  <marc:subfield code="b">1934-42</marc:subfield>
</marc:datafield>
- <marc:datafield tag="362" ind1="0" ind2="">
  <marc:subfield code="a">no. 1-15, July 1934-Dec. 1942; ser. 2, v. 1- Dec. 1943-
</marc:subfield>
</marc:datafield>
- <marc:datafield tag="500" ind1="" ind2="">
  <marc:subfield code="a">Title varies slightly.</marc:subfield>
</marc:datafield>
- <marc:datafield tag="515" ind1="" ind2="">
  <marc:subfield code="a">Suspended during 1939.</marc:subfield>
</marc:datafield>
- <marc:datafield tag="550" ind1="" ind2="">
  <marc:subfield code="a">"The quarterly journal of the Music Library
Association."</marc:subfield>
</marc:datafield>
- <marc:datafield tag="555" ind1="" ind2="">
  <marc:subfield code="a">No. 1-15, 1934-42 (Suppl. to ser. 2, v. 1, no. 2) with no. 13-15.
Indexes for ser. 2 issued for each two volumes and beginning with v. 2 are bound with
alternate volumes.</marc:subfield>
</marc:datafield>
- <marc:datafield tag="530" ind1="" ind2="">
  <marc:subfield code="a">Also available in an electronic version.</marc:subfield>
</marc:datafield>
- <marc:datafield tag="650" ind1="" ind2="0">
  <marc:subfield code="a">Music</marc:subfield>
  <marc:subfield code="x">Periodicals.</marc:subfield>
</marc:datafield>
- <marc:datafield tag="650" ind1="" ind2="0">
  <marc:subfield code="a">Music libraries</marc:subfield>
  <marc:subfield code="x">Periodicals.</marc:subfield>
</marc:datafield>
- <marc:datafield tag="650" ind1="" ind2="0">
  <marc:subfield code="a">Buddhism</marc:subfield>
  <marc:subfield code="x">Periodicals.</marc:subfield>
</marc:datafield>
- <marc:datafield tag="655" ind1="" ind2="7">
  <marc:subfield code="a">Computer network resources.</marc:subfield>
  <marc:subfield code="2">local</marc:subfield>
</marc:datafield>
- <marc:datafield tag="655" ind1="" ind2="7">
  <marc:subfield code="a">Electronic journals.</marc:subfield>
  <marc:subfield code="2">lcsh</marc:subfield>
</marc:datafield>
- <marc:datafield tag="700" ind1="1" ind2="">
  <marc:subfield code="a">O'Meara, Eva Judd,</marc:subfield>
  <marc:subfield code="e">ed.</marc:subfield>

```

```

</marc:datafield>
- <marc:datafield tag="700" ind1="1" ind2="">
  <marc:subfield code="a">Fox, Charles Warren,</marc:subfield>
  <marc:subfield code="d">1904-</marc:subfield>
  <marc:subfield code="e">ed.</marc:subfield>
</marc:datafield>
- <marc:datafield tag="700" ind1="1" ind2="">
  <marc:subfield code="a">Hill, Richard S.</marc:subfield>
  <marc:subfield code="q">(Richard Synyer),</marc:subfield>
  <marc:subfield code="d">1901-1961,</marc:subfield>
  <marc:subfield code="e">ed.</marc:subfield>
</marc:datafield>
- <marc:datafield tag="740" ind1="0" ind2="">
  <marc:subfield code="5">wid</marc:subfield>
  <marc:subfield code="a">Notes (Online)</marc:subfield>
</marc:datafield>
</marc:record>
</metadata>
</record>
</GetRecord>
</OAI-PMH>

```

## Identify

### Summary

Retrieves information about the repository

### Arguments

None

### Request Example

<http://il-aleph05:8995/OAI?verb=Identify>

### Flow

The OAI Data Provider retrieves information for the Identify response from the oaipubconf.xml configuration file. The earliest timestamp of the repository is retrieved from Z00P.

### Response

A sample response is found below.

Below is an explanation of the elements in the response:

- **<repositoryName>** – Name of repository, retrieved from the repositoryName tag of oaipubconf.xml
- **<baseURL>** – Base URL for OAI requests, retrieved from the baseURL tag of oaiconf.xml
- **<protocolVersion>** – Equal to 2.0
- **<adminEmail>** – Email of the repository administrator, retrieved from the adminEmail tag of oaipubconf.xml
- **<earliestDatestamp>** – Minimal Z00P-TIMESTAMP from Z00P
- **<deletedRecord>** – Equal to transient
- **<granularity>** – Equal to YYYY-MM-DDThh:mm:ssZ



**Flow:**

If a set is not specified, the OAI Data Provider uses the first <set> element listed in oaipubconf.xml as default set (e.g. OAI-SET). It retrieves the Publishing set name from the <internalSet> sub-element marked with the requested format (e.g. <internalSet>OAISETM:marc21</internalSet>).

If a set is specified, the OAI Data Provider uses the <set> element from oaipubconf.xml with a <setSpec> element that equals the requested set. It retrieves the Publishing set name from the <internalSet> sub-element that is marked with the requested format (e.g. <internalSet>OAISETM:marc21</internalSet>).

After the Publishing set value is determined, the OAI Data Provider retrieves record identifiers from Z00P records for which the Z00P-SET equals the requested set value. If from/until are also specified then only records which fall in the specified interval are retrieved.

The OAI Data Provider adds 30 record identifiers to the ListIdentifier response. All the records with the same Z00P-TIMESTAMP are also added to the same response.

If there are additional records in Z00P which satisfy the request criteria then the OAI Data Provider adds a resumption token to the response. The resumption token contains 'set', 'from' and 'until' values for the next retrieve action. If the user wants to retrieve additional records, the ListIdentifiers request must be sent with the resumption token. The OAI Data Provider will obtain 'set', 'from' and 'until' values from the resumptionToken, and retrieve additional record identifiers from Z00P according to these criteria.

**Response:**

A sample response is found below.

- <ListIdentifiers> element contains a number of <header> element, optionally followed by <resumptionToken> element.
- Each <header> element corresponds to a Z00P record. If the record is deleted then the status attribute of the <header> element is set to deleted.
- The <identifier> element contains an identifier in the form *oai:<repository identifier from oaipubconf.xml>:<Z00P-LIBRARY>-<Z00P-DOC-NUMBER>*.
- The <timestamp> element contains a timestamp in UTC form, obtained from Z00P-TIMESTAMP.
- The <setSpec> element contains <setSpec> value from oaipubconf.xml.

**Response Example:**

```
<ListIdentifiers>
<header>
  <identifier>oai:il-aleph05:USM01-000000001</identifier>
  <timestamp>2006-12-18T10:42:36Z</timestamp>
  <setSpec>OAI-SET</setSpec>
</header>
<header>
```







```
<setSpec>HISTORY</setSpec>  
<setName>History</setName>  
</set>  
<set>  
<setSpec>SCIENCE</setSpec>  
<setName>Science</setName>  
</set>  
</ListSets>
```