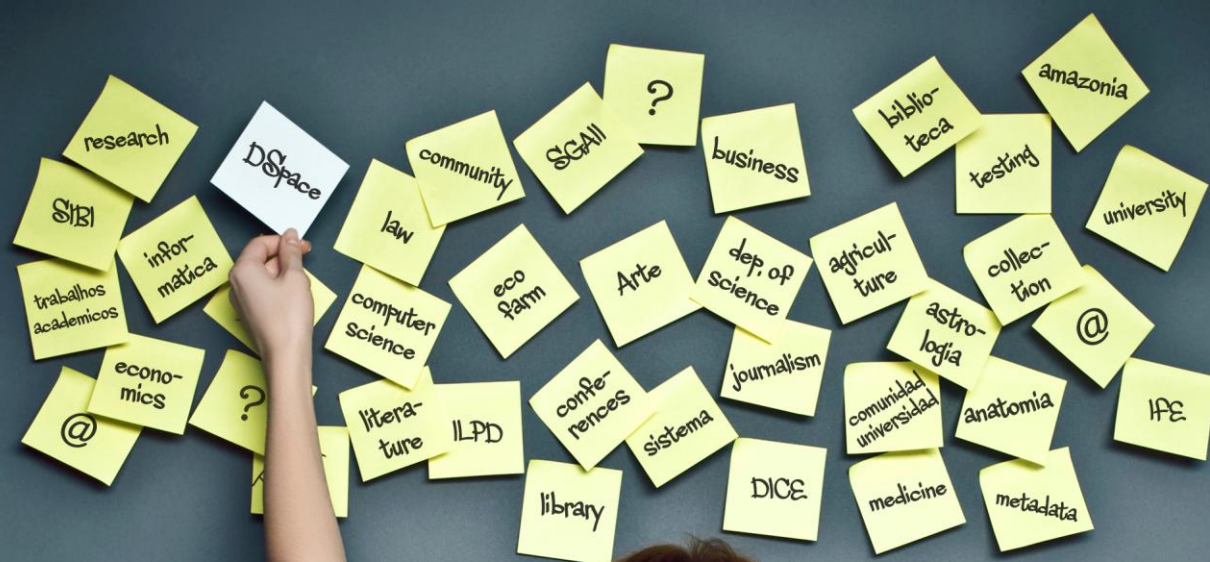


# XOAI for DSpace

Manual







## Introduction

Open Archives Initiative Protocol for Metadata Harvesting, which is a low-barrier mechanism for repository interoperability. *Data Providers* are repositories that expose structured metadata via OAI-PMH. *Service Providers* then make OAI-PMH service requests to harvest that metadata. OAI-PMH is a set of six verbs or services that are invoked within HTTP<sup>1</sup>.

## What is XOAI?

XOAI is a java implementation of an adaptable OAI-PMH data provider interface.

## Why XOAI?

Projects like OpenAIRE<sup>2</sup>, Driver<sup>3</sup> and EUBrazilOpenBio<sup>4</sup> have specific metadata requirements (to the published content through the OAI-PMH interface). As the OAI-PMH protocol doesn't establish any frame to these specifics, XOAI can, in a simple way, have more than one instance of an OAI interface, so one could define an interface for each project. That is the main purpose, although, XOAI allows much more than that.

## Concepts

To understand how XOAI works, one must understand the concept of Filter, Transformer and Context. With a Filter it is possible to select information from the data source. A Transformer allows one to make some changes in the metadata before showing it in the OAI interface. XOAI also adds a new concept to the OAI-PMH basic specification, the concept of context. A context is identified in the URL:

---

<sup>1</sup> <http://www.openarchives.org/pmh/>

<sup>2</sup> <http://www.openaire.eu/>

<sup>3</sup> <http://www.driver-support.eu/>

<sup>4</sup> <http://www.eubrazilopenbio.eu/>



<http://www.example.com/xoai/<context>>

Contexts could be seen as virtual distinct OAI interfaces, so with this one could have things like:

- <http://www.example.com/xoai/driver>
- <http://www.example.com/xoai/openaire>
- <http://www.example.com/xoai/request>

With this ingredients it is possible to built a robust solution that fulfill all requirements in *Driver*, *OpenAIRE* and also other projects specific requirements. As shown in Figure 1, with contexts one could select a subset of all available items in the data source. So when entering the *OpenAIRE* context, all OAI-PMH request will be restricted to that subset of items.

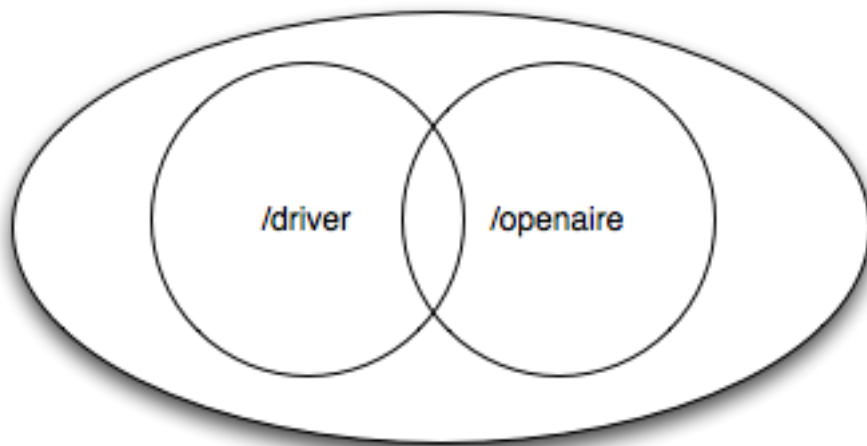


Figure 1: Distinct Contexts

At this stage, contexts could be seen as sets (also defined in the basic OAI-PMH protocol). The magic of X-OAI happens when one need specific metadata format to be shown in each context. Metadata requirements by *Driver* slightly differs from the





*OpenAIRE* ones. So for each context one must define it's specific transformer. So, contexts could be seen as an extension to the concept of sets.



## XOAI for DSpace

A specific implementation of the XOAI was released to DSpace<sup>5</sup>, one will describe how one could install it under a DSpace 1.8.2 installation.

### Install

#### Requirements:

- DSpace 1.8.2 Source Distribution
- Java JDK 1.6

#### Expressions:

- [XOAI-ADDON-DIR] Replace this expression by the path to the XOAI addon directory.
- [DSPACE-SOURCE] Replace this expression by the path to the DSpace source folder.

#### Automatic Install

Open the command line and execute the following commands:

```
$ cd [XOAI-ADDON-DIR]
$ ./installer.sh [DSPACE-SOURCE]
```

Then copy the content of the [XOAI-ADDON-DIR]/dspace-parent to the [DSPACE-SOURCE] directory.

---

<sup>5</sup> <http://www.dspace.org>





## Conclusion

XOAI for DSpace it is an easy to install add-on with few dependencies, but a powerful OAI Interface provided by Lyncode (<http://www.lyncode.com>).

## Help

If experiencing any problem, please contact [dspace@lyncode.com](mailto:dspace@lyncode.com)..