

PKP Harvester2 in an Hour
Version 2.0

Administrator's Guide



SIMON FRASER
UNIVERSITY library

This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivs License.
To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-nd/2.0/ca/>
or send a letter to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.





Table of Contents

Introduction.....	2
About the Public Knowledge Project.....	2
About PKP Harvester2.....	2
Features.....	3
For Users Of PKP Harvester 1.x.....	3
Installation.....	4
System Requirements.....	4
Unpacking Harvester2.....	5
Entering Administration.....	6
Archives.....	7
Adding an Archive.....	7
Deleting an Archive.....	8
Managing Archives.....	9
Command-line Harvesting.....	10
Crosswalks.....	10
URL-Based Queries.....	12
Plugins.....	13
Reading Tools.....	14
Modifying the Look and Feel of Harvester2.....	14
Getting More Information.....	15



Introduction

About the Public Knowledge Project

The Public Knowledge Project (<http://pkp.sfu.ca>) is dedicated to exploring whether and how new technologies can be used to improve the professional and public value of scholarly research. Bringing together scholars, in a number of fields, as well as research librarians, it is investigating the social, economic, and technical issues entailed in the use of online infrastructure and knowledge management strategies to improve both the scholarly quality and public accessibility and coherence of this body of knowledge in a sustainable and globally accessible form. The project seeks to integrate emerging standards for digital library access and document preservation, such as Open Archives and InterPARES, as well as for such areas as topical maps and doctoral dissertations.

About PKP Harvester2

The PKP Harvester2 is an open-source metadata harvester and aggregator that has been developed by the Public Knowledge Project through its federally funded efforts to expand and improve access to research. Harvester2 has been designed with flexibility in mind and supports multiple harvesting protocols and metadata formats with an emphasis on performance and simplicity of use. In concert with the PKP software suite, including Open Journal Systems and Open Conference Systems, the goal of Harvester2 is to promote open access publishing and contribute to the public good on a global scale.

Version 2.x represents a complete rebuild and rewrite of the PKP Harvester 1.x, based on the platform pioneered by the Public Knowledge Project with Open Journal Systems 2.x.

User documentation for Harvester2 can be found on the Internet at <http://pkp.sfu.ca/harvester2/demo/index.php/index/help>; a demonstration site is available at <http://pkp.sfu.ca/harvester2/demo>.

Features

- Fully functional harvesting and search engine without any coding required
- Built-in support for OAI Protocol for Metadata Harvesting (v1.1 and v2.0)
- Built-in support for Dublin Core, MODS, and MARC metadata formats
- Additional support for harvesting protocols and metadata formats may be added via plug-ins
- Reading Tools for content, based on administrator's choice
- Context-sensitive online help

For Users Of PKP Harvester 1.x

PKP Harvester2 is a complete rewrite of PKP Harvester 1.x based on the platform pioneered with OJS (Open Journal Systems) 2.x. Its appearance and basic functionality were taken from PKP Harvester 1.x, but technically, it has much more in common with OJS 2.x.

There is currently no migration script to transfer metadata from a Harvester 1.x installation to a Harvester2 installation. Because of the many technical differences between the two packages, it is unlikely that a script will be written to convert metadata; however, a script will probably be included with future releases of Harvester2 that will ease migrations of archive lists. If you are interested in such a script, contact pkp-support@sfu.ca for more information.

PKP Harvester2 currently supports all features of PKP Harvester 1.x.



Installation

For full installation instructions, please read the `docs/README` document shipped with Harvester2.

System Requirements

A server environment meeting the following requirements is recommended:

- PHP support (4.2.x or later)
- MySQL (3.23.23 or later)
- Apache (1.3.2x or later) or Apache 2.0 (2.0.4x or later) or Microsoft IIS 6 (PHP 5.x required)
- Linux, BSD, Solaris, Mac OS X, Windows operating systems

Other versions or platforms may work but are not supported and may not have been tested. We welcome feedback from users who have successfully run Harvester2 on platforms not listed above.

Unpacking Harvester2

To begin the Harvester2 installation process, download the current release from <http://pkp.sfu.ca/harvester2> and unpack it into a path on your web server (e.g. /var/www). If your server is properly configured, you will be able to point a web browser at this location (e.g. <http://localhost/harvester-2.0.0>) to receive the installation page:



Figure 1: Harvester2 Installer

Fill in the required fields as described on the installation form and click the “Install Harvester2” button at the bottom of the page when finished. Once this step is successfully completed, Harvester2 is installed and ready to use.

Entering Administration

Once you've installed Harvester2, you will need to log in as Administrator to access the Administration functions of the application. If your Harvester2 installation is available at <http://localhost/harvester-2.0.0/index.php>, you can log in at <http://localhost/harvester-2.0.0/index.php/admin>.

The image below shows the main administration page:

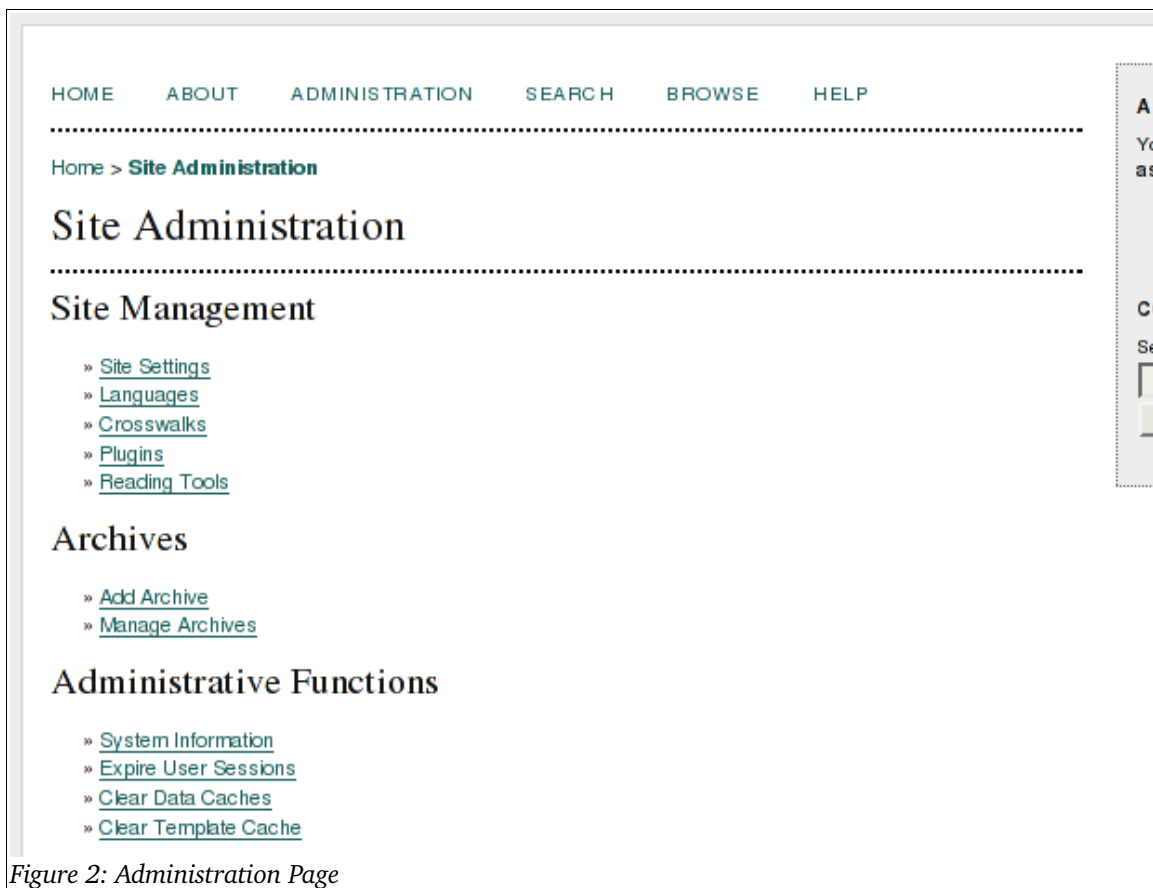


Figure 2: Administration Page

Each of the options on the Administration page are described in detail in the online help system. The most important components of Harvester2 administration, Archives, Crosswalks, Plugins, and Reading Tools, are described in the sections below.

Archives

To create, delete, update, and otherwise manage the archives that are harvested by your installation of Harvester2, follow the “Add Archive” or “Manage Archives” link under the Archives heading in Administration.

Adding an Archive

Following the “Add Archive” link from Administration will lead you to a form requesting the major pieces of information about an archive:



The screenshot shows a web form titled "Add Archive" with a dotted line separator below the title. The form contains the following fields and controls:

- Title***: A text input field.
- Description**: A large text area for entering details.
- URL***: A text input field with a placeholder example: "e.g. http://www.yourarchive.com".
- Public ID**: A text input field with a note below it: "This unique identifier can be used in URL-based searches to identify this archive."
- Type***: A dropdown menu with "OAI" selected.
- OAI Base URL***: A text input field with a placeholder example: "e.g. http://www.yourarchive.com/oa/index.php".
- Index Method***: A dropdown menu with "ListRecords" selected.
- Metadata Format***: A dropdown menu with "Dublin Core" selected, and a "Refresh" button next to it.
- At the bottom, there are two buttons: "Save" (highlighted in green) and "Cancel".

Figure 3: Archive Add Form



The fields on this form will vary depending on the type of harvester selected (and, potentially, by additional plugins); in this case, the OAI harvester is chosen.

Regardless, this form will always include the Title, Description, Public ID, and URL fields. These are for informational purposes, and are presented to users when they request information about a particular archive.

The OAI harvester has two additional fields, OAI Base URL, which will be used to generate harvesting requests for the target system, and Metadata Format, which lists the metadata formats supplied by the archive and supported by Harvester2. Once an OAI Base URL is entered, press “Refresh” to fetch a list of supported formats. (The set of schema plugins are mapped to OAI metadata prefixes in `registry/schemaMap.xml`.)

Submitting this form will take the administrator to the archive's management page.

Deleting an Archive

Following the “Manage Archives” link from Administration displays the current list of archives. From this page, archives can be deleted. Note that deleting an archive also removes all harvested records.

Managing Archives

Following the “Manage Archives” link from Administration displays the current list of archives. These can be Edited (leading to the form in Figure 3: Archive Add Form), Deleted, or Managed.

The figure below illustrates the Management page for an archive:

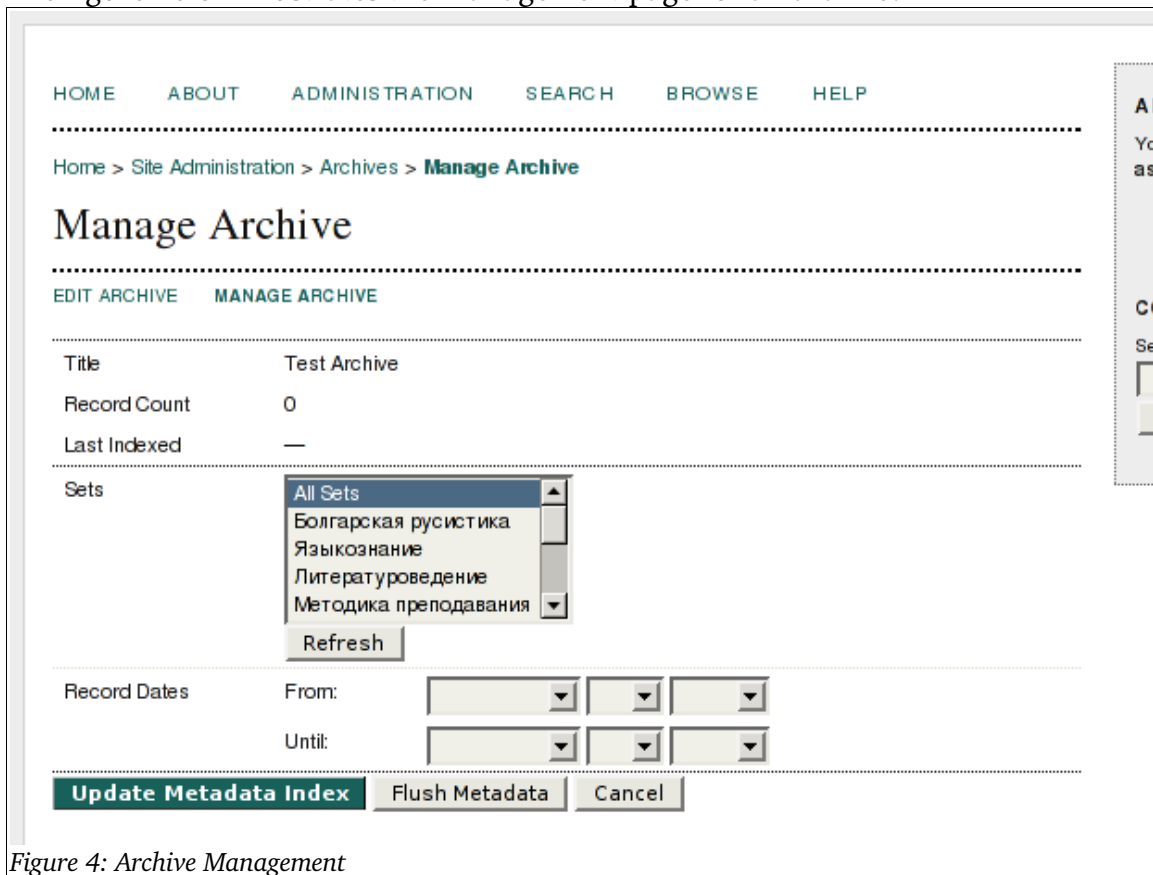


Figure 4: Archive Management

The content of this page is dependent on the harvester being used; in this case, the page for the OAI harvester is displayed. This page allows the administrator to flush records or harvest records from the remote server, selectively if desired.

If a record supplied by the remote server already exists in the local database, the existing record's metadata is deleted and replaced with the new values.

Command-line Harvesting

When harvesting a large archive using the web-based interface, it is possible for a time-out situation to occur; in this case, the archive will only be partially harvested. For large harvests, it is preferable to use the command-line harvesting tool, `tools/harvest.php`.

The tool is invoked as follows:

```
php tools/harvest.php list                # Lists archives
php tools/harvest.php [archiveId] usage  # Display usage
php tools/harvest.php [archiveId] flush  # Flush & harvest
```

Additional options are available, depending on the harvester plugin used, to implement features such as selective harvesting; use the above syntax to display usage for a full list of options. For example, the OAI harvester allows harvesting by sets and timestamps, just as it does in the web administration interface.

One useful strategy for keeping the metadata in your Harvester2 up to date is to set up a cron job to reharvest archives every day or week.

Crosswalks

Crosswalks are used to provide searching of and sorting by equivalent fields across multiple schema, such as Title fields in Dublin Core, MARC and MODS schema.

To define and edit crosswalks, follow the “Crosswalks” list from Administration. The figure on the following page illustrates the crosswalk creation form.

When creating a crosswalk, the administrator must choose the crosswalk type from three choices:

- **Text:** The fields chosen are treated as regular text-based fields.
- **Select:** The fields chosen are treated as a controlled vocabulary, and a select field is presented on the search form
- **Date:** The fields chosen are date fields. When this type is chosen, only date fields are displayed.

The list of fields may be extremely long; below the type selection is a “schema filter” selection that allows the administrator to limit the current display of fields to a single schema. (Note that changes to the list of selected fields should be saved before choosing a different schema from the schema filter.)

Add Crosswalk

Name*

Description*

Public Crosswalk ID
When specified, the public crosswalk ID can be used in URL-based searches.

Type

Text

Select

Date

Allow users to sort records using this crosswalk when browsing records

Schema Filter

Schema	Search	Field
Dublin Core	<input type="checkbox"/>	Identifier: An unambiguous reference to the resource.
	<input type="checkbox"/>	Abstract: A summary of the content of the resource.

Figure 5: Crosswalk Form

Crosswalks can additionally be designated as “sortable” by checking the “Allow users to sort records using this crosswalk when browsing records” checkbox. In this case, a second row of options will appear next to the available column names. This allows the administrator to choose specific fields for sorting; while any number of fields can be selected for cross-schema searching, only one may be chosen for sorting.



Crosswalks are displayed and used only when metadata formats are spanned, such as when several archives consisting of several formats are being searched.

URL-Based Queries

Harvester2 provides a simple syntax for performing searches using a standard GET URL. This feature is useful if you want to provide dynamically generated links to Harvester2 content from other web applications.

The base URL used in this type of query is <http://localhost/harvester-2.0.0/index.php/search/byUrl>. To specify a query, add parameters to this base URL; for example, in order to search a specific archive, set `archive=[public archive ID]`, where `[public archive ID]` is the value set in the archive's administration form. Multiple values for this parameter are allowed. Likewise, to search a specific field, address it by name; for example, `byUrl?title=[search value here]`.

When searching with a specific archive or several archives of the same metadata format, all field names for that schema are allowed. When searching without specifying an archive, or when the specified archives aren't of the same metadata format, only public crosswalk IDs can be specified (see the crosswalk form).

Some examples:

To search all archives using the "title" crosswalk for the word "test":

<http://.../byUrl?title=test>

To search the archive with the public ID "test1" using the DC field "title" for the word "test":

<http://.../byUrl?archive=test1&title=test>

To search the archives with public IDs "test1" and "marc" using the "title" crosswalk for the word "test":

[http://.../byUrl?archive\[\]=test1&archive\[\]=marc&title=test](http://.../byUrl?archive[]=test1&archive[]=marc&title=test)

Plugins

Many of the core features of the Harvester2 are implemented using plugins. For example, the OAI harvester and the schema formats themselves (Dublin Core, MARC and MODS) are all implemented as plugins that ship with Harvester2; additional harvesting protocols and schema formats can be added easily without modifying the existing codebase.

The technical aspects of plugins are described in greater detail in the Harvester2 Technical Document, also available from the PKP web site.

In addition to harvesters and metadata formats, plugins can be used to implement several additional functions such as filtering harvested data and extending metadata handling. Two such plugins ship with Harvester2:

- **Language Map Preprocessor:** Using an XML file defining specific mappings (`plugins/preprocessors/languagemap/mapping.xml`), this plugin converts language codes to preferred values (generally two-letter language codes to three-letter codes).
- **PKP Dublin Core Extender:** When configured on a per-archive basis (see the Archive form when this plugin is enabled), this plugin modifies the Dublin Core schema by adding two additional fields, “Discipline” and “Subject Classification”, that are supplied by Open Journal Systems via the OAI harvesting protocol.

These plugins must be individually enabled by following the “Plugins” link from Administration and clicking “Enable” below the plugin's name. If additional plugins are installed, they may be managed likewise in the “Plugins” page.



Reading Tools

The Reading Tools provide relevant links to external resources like search engines and dictionaries. These are configured by following the “Reading Tools” link from Administration. Each archive has a fully configurable set of reading tools; a set is provided and can be installed by following the “Versions” link from Reading Tools and clicking “Restore Versions to Defaults”.

When enabled, Reading Tools are displayed on the right-hand side of the record display. Readers can double-click on a term to launch a window providing resources to find further information on the particular term.

Modifying the Look and Feel of Harvester2

Harvester2 uses PHP Smarty Templates (<http://smarty.php.net/>) and Cascading Stylesheets, so modifying its look and feel is straight forward. Templates and styles are located in the `templates/` and `styles/` directories of the Harvester2 installation, respectively.



Getting More Information

For more information, see the PKP web site at <http://pkp.sfu.ca>. There is a Harvester2 support forum available at <http://pkp.sfu.ca/support/forum>; this is the preferred method of contacting the Harvester2 team. Please be sure to search the forum archives to see if your question has already been answered.

If you have a bug to report, see the bug tracking system at <http://pkp.sfu.ca/bugzilla>.

The team can be reached by email at pkp-support@sfu.ca.