

# Harvester Manual

## Index Data

## Contents

<b>Overview</b>	<b>1</b>
<b>Harvest Jobs</b>	<b>2</b>
Add new Resource . . . . .	2
General Job Settings . . . . .	2
Resource-specific Settings . . . . .	3

## Overview

This document describes the Harvester using the following notation:

- Resources to harvest and on what schedule
- Harvester Job, the actual process of harvesting a resource at a given time, a term used in the document
- Storages (or Local Unified Indexes)
- Transformation Steps, which transform data from one format to another or do transformations on the same format
- Transformation Pipelines, which consist of multiple Transformation Steps

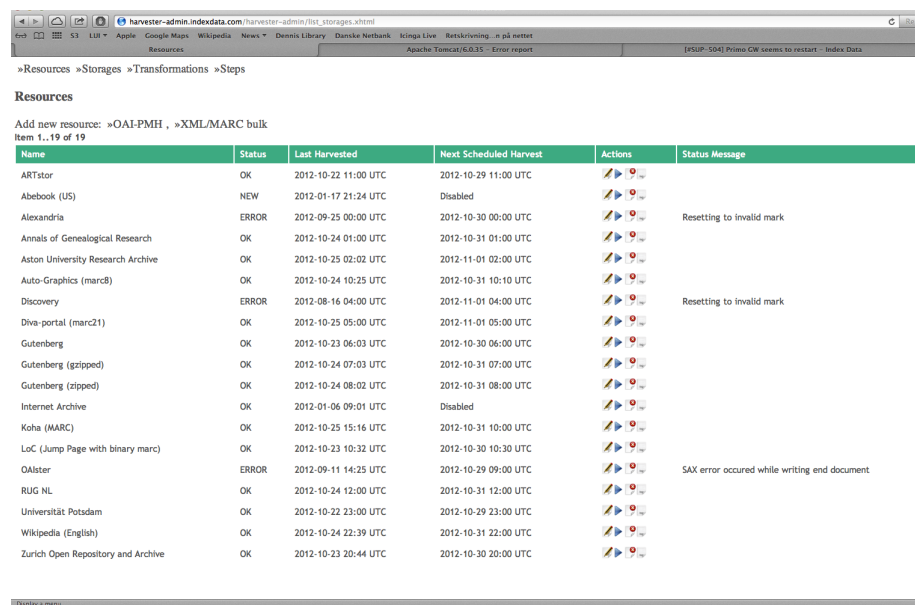
A Harvester Job fetches data via http from a remote Resource, does some initial extraction/conversion of records depending on the protocol and configuration, splits the data into minor chunks (necessary on larger datasets due to memory constraints), sends this data through a Transformation Pipeline, and then sends the data to a Storage (Local Unified Index) for indexing.

The indexed resources are exposed in a MasterKey 2 Toroid to be used as targets for a MasterKey 2 meta search solution.

# Harvest Jobs

When you access the Harvester Admin web site (URL and login credentials, or lack of thereof, is deployment specific) you will be welcomed by the Harvest Job page.

On the Harvest Job page, you will see a list of currently harvested resources and their status, and you can perform actions like Edit, Run, Delete and View the latest job log.















































































Name	Status	Last Harvested	Next Scheduled Harvest	Actions	Status Message
ARTstor	OK	2012-10-22 11:00 UTC	2012-10-29 11:00 UTC	   	
Abebook (US)	NEW	2012-01-17 21:24 UTC	Disabled	   	
Alexandria	ERROR	2012-09-25 00:00 UTC	2012-10-30 00:00 UTC	   	Resetting to invalid mark
Annals of Genealogical Research	OK	2012-10-24 01:00 UTC	2012-10-31 01:00 UTC	   	
Aston University Research Archive	OK	2012-10-25 02:02 UTC	2012-11-01 02:00 UTC	   	
Auto-Graphics (marc8)	OK	2012-10-24 10:25 UTC	2012-10-31 10:10 UTC	   	
Discovery	ERROR	2012-08-16 04:00 UTC	2012-11-01 04:00 UTC	   	Resetting to invalid mark
Divva-portal (marc21)	OK	2012-10-25 05:00 UTC	2012-11-01 05:00 UTC	   	
Gutenberg	OK	2012-10-23 06:03 UTC	2012-10-30 06:00 UTC	   	
Gutenberg (gzipped)	OK	2012-10-24 07:03 UTC	2012-10-31 07:00 UTC	   	
Gutenberg (zipped)	OK	2012-10-24 08:02 UTC	2012-10-31 08:00 UTC	   	
Internet Archive	OK	2012-01-06 09:01 UTC	Disabled	   	
Koha (MARC)	OK	2012-10-25 15:16 UTC	2012-10-31 10:00 UTC	   	
LoC (Jump Page with binary marc)	OK	2012-10-23 10:32 UTC	2012-10-30 10:30 UTC	   	
Oalster	ERROR	2012-09-11 14:25 UTC	2012-10-29 09:00 UTC	   	SAX error occurred while writing end document
RUG NL	OK	2012-10-24 12:00 UTC	2012-10-31 12:00 UTC	   	
Universität Potsdam	OK	2012-10-22 23:00 UTC	2012-10-29 23:00 UTC	   	
Wikipedia (English)	OK	2012-10-24 22:39 UTC	2012-10-31 22:00 UTC	   	
Zurich Open Repository and Archive	OK	2012-10-23 20:44 UTC	2012-10-30 20:00 UTC	   	

Figure 1: Figure 1-1. Harvest Job page.

From here you can manage existing resources or add new ones.

## Add new Resource

The Harvester currently supports harvesting OAI-PMH resources and XML/MARC binary bulk data.

Click the OAI-PMH or XML/Marc Bulk to add a new Resource to harvest.

## General Job Settings

Setting up a new Harvesting job consists of entering some general harvesting information plus specific settings for this type (OAI-PMH or XML/Marc Bulk).

The screen capture below shows the general settings:

- Name: Preferably a unique name for users to identify this Harvester resource.
- Content Description and Technical and Contact Notes (not used by the Harvester, but for ease of support staff). Harvest schedule: A recurring time at which the Harvester job should run.

Checkboxes for:

- Harvest now: Run the harvesting job right after Add/Save.
- Harvest job enabled: The harvesting job will only be scheduled if checked.
- Overwrite: Delete all data before the run. Since OAI-PMH supports increment updates, this is not generally used with OAI-PMH jobs. If used with OAI-PMH, it is necessary to clear “Harvest From” and Resumption Token, otherwise it will only be a partial data set.
- Transformation Pipeline: In order to get the data stored, we need to transform the data from whatever input format the feed is delivering, into an internal format used by the Harvester.
- Storage: Select which storage the harvested data should use. The Harvester supports multiple backend storages. This could be for staging like Development, Testing and Production, or it could be for different customers.

### **Resource-specific Settings**

Depending on which resource type you choose, the following settings will apply.

### **OAI-PMH Specific Information:**

- OAI Repository URL: a link (http-based) to the resource to harvest. The base link defined by OAI Set Name: some resources have multiple sets within the repository. If none is selected, the full repository will be harvested.
- Metadata Prefix: A repository uses one of two prefixes (or data embedding format): Dublin Core or MARC XML embedded data. It is important to choose the right one otherwise no data will be harvested. Also, the Transformation selected should match this format otherwise no record will be found.

- Use long data format: Check box to indicate whether to use a long data format when asking for records from the OAI-PMH resource. This is not used very often, but should match what the resource requires.
- Harvest from date: If empty and no resumption token is set, the Harvester will harvest the full data set from the resource. On completion the Harvester will set the date to yesterday, so next run will only harvest the newer records received since last run.
- Resumption token: The OAI-PMH protocol supports chunking bigger datasets into smaller chunks. On delivery of a chunk the OAI-PMH returns a token which the next request should use in order to get the next chunk. If an OAI-PMH job halts before completion the resumption token will be set in this field. Sometimes it is possible to run it again from this resumption point at a later stage, but this is not always supported.

**XML Bulk Specific Information:** The XML/MARC specific settings look like this:

- URLs: One or more space-separated URL to XML or MARC binary data. Jump or index pages (e.g. html pages with URLs) are supported as well.
- Split at depth: For XML data. This should usually be set to 1 for XML feeds, if we want to harvest the record elements in the data structured like:

```
<root>
  <record/>
  <record/>
  ...
</root>
```

- Split (number of records): The Harvester tries to imply streaming parsing where possible, but some (rather most) XSL Transformations will not support this. Attempting to transform millions of records will be too memory consuming. Breaking into chunks of 1000 seems to be a reasonable option.
- MIME-type for compressed data: The Harvester detects the type (XML vs MARC binary) from the MIME-type. A correctly configured web site will send a MIME-type of Application/marc if the file type is .mrc. If the MIME-type received is different than expected (because of a wrongly configured web site or wrong file type), the MIME-type might need to be overridden. The format of the field is: **MIME-type [; optional character encoding]**. The Harvester supports gzipped data (and partly supports zipped data: only the first entry will be extracted), but the Harvester then needs to be configured of the format the compressed data contains (XML or MARC).

- Output format: This express the output format of binary MARC reading which will be the input for the transformation pipeline. If the Transformation Pipeline expects MARC21 XML, this should be set to Application/marc. If the pipeline expects Turbo MARC XML, it should be set to Application/tmarc.