ECP-2007-DILI-517009

EuropeanaLocal

Technical support documentation and help desk online

Deliverable number	*D4.2*
Dissemination level	*PP*
Delivery date	*21 June 2009*
Status	*Final*
Author(s)	*Stein Runar Bergheim, Olav Tuften, Idar Thoresen Kvam, Rastislav Rehak with inputs from EDL Foundation, and the EuropeanaLocal project*

e*Content*plus

This project is funded under the *e*Content*plus* programme[1], a multiannual Community programme to make digital content in Europe more accessible, usable and exploitable.

[1] OJ L 79, 24.3.2005, p. 1.

Table of Contents

TABLE OF CONTENTS **2**

1 INTRODUCTION **4**

2 DOCUMENT CONVENTIONS **5**

3 EUROPEANALOCAL GLOSSARY **6**

4 OVERVIEW OF THE IMPLEMENTATION PROCESS **12**

5 BEFORE YOU BEGIN **13**

KNOWLEDGE REQUIREMENTS 13
HARDWARE REQUIREMENTS 13
SOFTWARE/PLATFORM REQUIREMENTS 13

6 SELECTING A REPOSITORY SOFTWARE PACKAGE **14**

RECOMMENDATIONS 14
SOFTWARE COMPARISON MATRIX 15

7 INSTALLING REPOSITORY SOFTWARE ON WINDOWS **18**

WINDOWS REQUIREMENTS: 18
INSTALLING PKP HARVESTER ON WINDOWS 18
INSTALLING REPOX ON WINDOWS 19
INSTALLING CELESTIAL ON WINDOWS 20

8 INSTALLING REPOSITORY SOFTWARE ON LINUX **21**

LINUX REQUIREMENTS 21
INSTALLING PKP HARVESTER ON LINUX 22
INSTALLING REPOX ON LINUX 23
INSTALLING CELESTIAL ON LINUX 24

9 RELEVANT STANDARDS **25**

XML – EXTENSIBLE MARKUP LANGUAGE 25
DUBLIN CORE 26
EUROPEANA SEMANTIC ELEMENTS 26
THE OAI-PMH WEB SERVICE PROTOCOL 28

10 METADATA EXTRACTION AND NORMALIZATION **31**

METADATA EXTRACTION 31
METADATA FIELD MAPPING 31
METADATA NORMALIZATION 31
WRITING THE XML FORMAT 32

11 TESTING YOUR REPOSITORY **34**

12 TROUBLESHOOTING AND SUPPORT **35**

EUROPEANALOCAL SUPPORT FORUM 35
OTHER SUPPORT OPTIONS 35

BIBLIOGRAPHY **37**

TABLES AND DIAGRAMS **38**

APPENDICES **39**

APPENDIX 1: DUBLIN CORE METADATA ELEMENTS 39
APPENDIX 2: OAI PROTOCOL FOR METADATA HARVESTING 39
APPENDIX 3: EUROPEANA SEMANTIC ELEMENTS V3.1 39
APPENDIX 4: ESE IMPLEMENTATION GUIDELINES 39

1 Introduction

This deliverable comprises the following technical support documents, together with an online help desk which can be viewed at: **http://www.avinet.no**, all end-users will receive username/password for this through the technical contact addresses provided in the technical survey.

For test-access to the forum, please use the username "**EuropeanaLocal**" (without the quotes) and password "**europeana**" (also without quotes and in lower-case letters).

This manual in combination with a series of technical workshops conducted throughout the project are designed to support EuropeanaLocal content provider partners in establishing OAI-PMH compliant repositories and populate them with metadata from their digital collections. The manual can be used as a stand-alone reference for those who wish to take on the task of implementing repositories without any aid, but will also be used as supporting material during the regional Europeana Local technical workshops where content contributor partners will receive hands-on training in installation, population and harvesting of OAI-PMH repositories.

2 Document Conventions

In order to ease the use of the manual, the following notations have been applied consistently throughout the document:

```
Examples including script code and data sections are shown in Courier
typeface on grey background
```

Italic text should be read and followed carefully as such tasks require careful attention.

Tips and tricks which may be useful for content contributors will be highlighted in bold.

3 EuropeanaLocal Glossary

The following list defines useful terms and abbreviations for content contributors to Europeana. The definitions below are valid when used within the context of Europeana Local. Several terms have multiple definitions, the definition given below is not necessarily the official one.

Term	Definition
Application profile	Adaption of a metadata standard for a specific purpose or use.
Athena	A sister-project to EuropeanaLocal seeking to develop tools and methods for improved metadata extraction and enrichment from MLA content contributors across Europe.
Authentication	In the context of Europeana Local, the process of verifying the identity of an external party who wants to harvest a source repository.
Authority file	An authority file defines a list of known unique entities. Example: • A list of person names • A list of geographical names • A list of time periods
Creative Commons	A set of predefined rights expressions which may be used to state the IPR associated with content from a local content contributor.
Dictionary	A list with definitions of words and/or translations of words between different languages
Digital collection	In the context of Europeana Local, a digital collection is a uniform database held by a content provider.
Dublin Core	Metadata standard, defines a wide range of descriptive metadata for any kind of item
Dublin Core (qualified/simple)	Dublin Core with the minimal set of mandatory fields present and populated.
EDLF	Abbreviation for the European Digital Library Foundation. Foundation which among other things is responsible for running and maintaining the Europeana service.
ESE	Abbreviation: Europeana Semantic Elements, see Europeana Semantic Elements

Europeana	A web application run and developed by the EDLF. The equivalent of Google for searching within the domain of quality controlled cultural heritage information sourced from MLA organizations across Europe.
EuropeanaLocal	This project, aiming to identify the shortest path for local or regional content providers across Europe to connect their content to Europeana.
Europeana Semantic Elements	An application profile of Dublin Core adapted by the European Digital Library Foundation for the purpose of enabling desired functionality in Europeana.
Europeana themes	A set of themes proposed by Europeana which should be guiding for which collections should be prioritized. Available off the Europeana website but is subject to change.
GIS	Abbreviation for Geographical Information System
Harvesting	See metadata harvesting
Harvesting authority	A registry contained in a harvesting repository (e.g. Europeana itself, national aggregators, etc) where any number of source repositories may be entered and subsequently harvested.
ID	Abbreviation for identifier. See identifier, persistent identifier
Identifier	A number or string which is used to uniquely identify a content item within a global or local scope. Example: an item might have the ID 23568 in a local collection but once this item is harvested by Europeana other items with the same ID may exist because the ID is unique only unique within the scope of the local collection. By prefixing a local ID with a unique string it will be unique in the global scope. A common way of doing this is through the use of URIs. The URI may be the web address (URL) which if typed into a web browser will show the presentation page of the item from the web site of the content provider.
IPR	Abbreviation for Intellectual Property Rights
Item	An Item is one record of metadata describing an abstract or physical entity in terms of the elements described in ESE.

	See: ESE, application profile
Metadata	Data describing a resource in terms of intellectual property rights, what, where, when and who. In the context of Europeana Local, the resources described have their own unique web page.
Metadata enrichment	The process of improving the quality of metadata through manual or automated methods.
Metadata extraction	The process of reading metadata from custom digital collections held in e.g. SQL-databases and writing them in a format suitable for import to a repository.
Metadata harvesting	The process of reading metadata from a source repository over the web, typically using the OAI-PMH protocol which defines available web service methods (verbs) and their response format.
Metadata mapping	The process of mapping the fields of a source metadata format to the fields of a target metadata format. This process may include splitting of fields which may require extensive manual quality control. It may also include merging of fields and the automatic addition of static metadata which are missing in the source metadata format.
Miniature	In the context of Europeana, a miniature is a simplified representation of a content item, e.g. a thumbnail edition of a photo, a short audio-sample of a music track or a preview clip of a video file.
MLA	Abbreviation for Museums, Libraries and Archives.
Multi-lingual metadata	Metadata in more than one language. While this term is quite simple it points to a rather important problem. What can we use the metadata for if they are not in a common language? If the source vocabularies are not mapped to a common, multi-lingual vocabulary, searches for language specific words like “chair” will render hits only for English language resources while the equally important French “chaise” will be omitted.
OAI	Abbreviation for Open Archive Initiative. A body developing standards for storage, exchange and access to MLA data.
OAI-ORE	Abbreviation for Open Archive Initiative – Object Reuse

	and Embedding. A standard for harvesting of rich content items (images, audio-files, texts) as well as their metadata. OAI-ORE is envisaged to enable sophisticated services at the harvesting end but requires more efforts from content contributors to populate source repositories and requires larger bandwidth as the volume of data transferred will be large.
OAI-PMH	Abbreviation for Open Archive Initiative – Protocol for Metadata Harvesting. A standard which defines the allowed calls to and response from a web service offered by a repository. Is used to harvest the content of local repositories into aggregator repositories such as regional, national or trans-national portals such as Europeana itself.
Object	An Object is a composite representation of digital content consisting not only of metadata but also the actual content. An Object would typically be represented in a format such as RDF and contain embedded images, text, audio, video corresponding to the actual content resource.
Ontology	A comprehensive set of concepts with relationships between them. Ontology is defined in a language such as OWL or RDF. Example: A definition of a concept in an ontology typically consists of a source-relationship-target triplets as shown below: 1. Source: A chair 2. Relationship: Has parts 3. Target: A seat An ontology is instrumental in creating intelligent search and retrieval services as it may be used to aid users in searching across collections which are using differently named concepts to describe their containing items.
OWL	Abbreviation for Ontology Web Language. An XML-based format for defining ontology.
Persistent identifier	An identifier which remains unchanged over time.
RDF	Abbreviation for Resource Description Framework. An XML based format used to described metadata for content items or embed object data. RDF has a rich model for defining relationships between items. RDF may also be used to express ontology.
Repository	Metadata storage software which enables harvesting authorities to connect to it remotely over the Internet and

	download it's content in one or more formats. Repository software typically consists of a database with a web application which allows import of data, searching and browsing of the metadata and an OAI-PMH compliant web service interface. Examples of repository software include Fedora, Greenstone and DSpace.
Schema (database)	A database schema is a space in a database where information is kept separate from other information in the same database. Tables in a schema share some properties.
Schema (metadata)	A metadata schema is the same as a metadata format. ESE is a metadata schema based on Dublin Core. See: application profile.
Semantic Web	A type of web service which allows "intelligent" searches for information through definition of relationships between rich content items or linkage to a common ontology.
SKOS	XML format for describing a vocabulary, used to facilitate mapping between different vocabularies.
SKOSification/SKOSifying	The process of implementing a local vocabulary in the SKOS XML-format
Surrogate	The representation you can make of a metadata item based on the data contained in the metadata itself. A surrogate would typically be the page showing all the details of an item returned by a search in Europeana.
Thesaurus	Plural: thesauri. A dictionary defining synonyms for words. May be used to automate the mapping from one vocabulary to another.
URI	Abbreviation for Uniform Resource Identifier. World Wide Web Consortium defined standard for uniquely identifying a resource on the Internet. Example: http://www.europeanalocal.eu/id=12
Vocabulary	A vocabulary is a list of words systematically used to describe content in a digital collection.
Vocabulary mapping	The process of mapping the "words" of one vocabulary to the "words" of another. May be automated through the use of thesauri and/or dictionaries which automatically define relationships between items based on synonyms.

WHAT metadata	Descriptive metadata indicating the thematic nature of the content through a classification system, vocabulary, dictionary or free text keywords. Example: image, people, black and white, hunt, fox, horses
WHEN metadata	Metadata indicating the temporal validity of an item. Example: • 14.08.1942 • 1942 • The 1940s • Second world-war
WHERE metadata	Metadata indicating the geographical location of an item. Example: • coordinates • geographical names
WHO metadata	Metadata indicating known named entities which are relevant to an item. If the item is a photo, this may be the name of people in the photo. If the item is an article it may be the name of an organization mentioned in the article.

4 Overview of the Implementation Process

The detailed procedures described in this manual will take you through the seven steps required to enable your locally held content to be harvested by and shown in the Europeana service.
Once you have established a repository, it may be harvested by any number of harvesting authorities dependant on who you give access to your repository. Intelligent semantic search services may be built based on these aggregated services.

1. Establish compliant hardware/software platform
2. Install repository software
3. Configure repository software
4. Extract and normalize metadata
5. Populate repository
6. Test that harvesting is working

5 Before You Begin

This section describes the prerequisites for carrying out the tasks in this manual. There are three sets of requirements which need to be met:

Knowledge requirements

It is assumed that users of this manual will be technically skilled and will have advanced understanding of PCs. Users should be experienced in regular software installation. Users do not need to have programming expertise but having played around with some simple scripts will be an advantage.

Hardware requirements

A server which should be usable as a repository needs to meet the following minimum requirements to deliver satisfactory performance:

- Processor speed: >= 1 x 2,0 GHz
- Storage space: dimensioned to data volume >= 200 GB
- Physical memory: >= 2GB

Software/platform requirements

Servers need to be installed with basic operating system software, either Linux/Unix or Windows based. The following software requirements needs to be met:

- Ubuntu Linux
- Apache Web Server
- Tomcat Application Server

Or

- Windows 2000/2003 Server
- Apache Web Server / Internet Information Server
- Tomcat Application Server

6 Selecting a Repository Software Package

Repository software serves several purposes. It stores metadata and can be used as a basis for developing custom local services. It typically allows browsing of metadata through a web interface and it offers the data to harvesting repositories over the Internet through TCP/IP communication.
The main requirement for content to be harvestable by Europeana (and regional/national aggregators) is that the XML returned by issuing a call to the repository is compliant with the OAI-PMH standard.
Repositories may be classified into static repositories and dynamic repositories.

- A static repository is merely dump of metadata from a digital collection into a static XML-file. This file can then be harvested by aggregating repositories over the Internet
- A dynamic repository returns OAI-PMH compliant XML as well as support a set of OAI-PMH requests and responses.

A number of different software packages offer dynamic repository functionality and if technically sophisticated partners wish to contribute content to Europeana she or he is free to do so by installing any compliant product. Partners who are in need of technical aid in order to set up their repositories are however recommended to choose one of the software packages which are described in this manual.

Recommendations

There is a large number of available repository software packages, most of which are open source and claim to be superior to each other left, right and center. EuropeanaLocal technical partners have therefore performed an assessment of different repository software packages based on a set of simple criteria reflecting the needs of local and regional content providers.

6.1.1 Recommended content provider repository software

For content providers, EuropeanaLocal technical partners have seen two criteria as being key:

- Ease of installation
- Ease of running and maintenance

Based on these criteria EuropeanaLocal technical partners recommend Repox (TEL+ Project, 2009) as content provider repository. The software package is easy to install, comes with its own web server and database and supports a multitude of ways of ingesting content from local files. The software also has an attractive interface for defining crosswalks between different metadata formats and is being further developed within the Europeana family of projects through Europeana v1. Repox is scheduled to support ESE starting in October 2009 until which time such support will have to be customized.
Other good choices include Celestial (simple) and Fedora (comprehensive).

6.1.2 Recommended aggregator repository software

For aggregators, EuropeanaLocal technical partners have seen three criteria as key:

- Ease of installation
- Ease of running and maintenance
- Customizability

The final criterion is defined because aggregators are not merely there to aggregate content to be passed on to Europeana but mainly to build value-added search and discovery end-user-services at regional and national level. The ease of developing such services based on the aggregated content is therefore an important factor in selecting a repository software package for an aggregator.
Based on the above criteria, EuropeanaLocal technical partners recommend PKP Harvester (Public Knowledge Project, 2009) as an aggregator repository. PKP Harvester is a very mature harvesting software package with a comprehensive APi for programming your own applications based on its data store. Recently a promising OAI-PMH provider interface has been added which makes this a good choice for aggregators wishing to develop their own applications as well as providing content for Europeana.
Other good choices include: Repox (provided all features promised will be present in the next version – currently lacks harvester), Fedora (provided additional harvester software is integrated or ingestion of content from underlying entities does not happen according over the OAI-PMH protocol. Austria has also shown good results using Celestial though this software package does not offer any API for customization or development of specialized search applications.

Software comparison matrix

Table 1: Feature comparison matrix for assessed repository software packages

Issue	Description	Celestial	Repox	PKP	Fedora[1]	Static
Import		n	y	n	y	n
Harvester	Has built-in	y	y	y	n	n
Sets		y	y	y	y	n
- harvest		y	y	y	y	
- provide		y	y	y	y	
- preserve		y	n	n	y	
Flow control	Resumption	y	y	y	y	n
Provenance	Identifier and rights	y	n	n	y	n
Stores format		the actual	tel	the actual	fobject	the act.
Crosswalks		n	y	y	y	n
- existing	Repox will get ESE transformation	na	dc, marc	dc, marc, mods	dc, marc, mods, mets	
Search	Fedora has RDF-based ResourceIndex	n	n	y	y	n
Database		Mysql Postgres	Derby Mysql	Mysql Postgres	Mysql Postgres Oracle	xml file
- how stored	As xml, fields, both	xml		xml	partly both	xml
Ease of installation	Demanding Normal Easy	e	e	e	n Windows n/d Linux	e

[1] Note. Fedora is also representing the tree other big content systems, DSpace, Eprints and Greenstone. They would give much the same results in the table.

OS Platform	Linux/Windows	both	both	both	both	any
Web Server		Apache IIS	Jetty	Apache IIS	Tomcat	Perl
Language	Programming language	Perl	Java	PHP	Java	Perl
ESE support	Must make crosswalks. Repox will support ESE.	n/a	Cross-walk	Plug-in	Cross-walk	n/a
Capacity Provider	Recognised databases	-	-	-	30 mill	5.000
Capacity Harvester	10.000 records at our test site, PKP search off/on	1.5 min	1.5 min	3/18 min	14 min	n/a
Scalability	Recognised databases.	y	y	y	y	n
Support/ Maintenance	Repox is maintained in eContent projects.	n	y	y	y	n
Deleted records	Repox will support. Is reported to PKP.	y	n	n	y	n/a
Suitable for whom	Provider Aggregator	A,P	P	A,P	A,P	P
Installations	Registries like Roar, Oaister, Celestial shows	Some 1180	TELPlus Other?	Many as harvester	Many	
Native language	English is standard	n	n	y	y	n
Supporting tools and utilities						
Customizability	Demanding Normal Easy	D	D	N,E	N,E	
API's		n	N	y	y	n
Security	Simple, more Advanced	s	S	a	a	n
Admin GUI		very simple	Simple	simple	simple	none
Command line	Harvest/Import	y	N	y	y	y
Content management		n	N	n	y	n

While the above matrix shows the comparison of features between the different software versions, the below table shows what are the major strong and weak points of the assessed technologies in the context of EuropeanaLocal.

Table 2: Advantages and drawbacks of repository software packages

Tools	Pros	Cons
Static repository	- Simple, just a XML file - General, replaces import	- No sets. - No resumption token. - Only few records.
Celestial	- Perl, fast - Simple, but good datamodel - Mirror, preserve everything - Takes any metadata format	- No Import - No Search - Too simple admin - Lacks crosswalk to dc, must store it

	- Has provenance	- No maintenance
Repox	- Java, pretty fast - Will get ESE support and Import - Maintained - By eContent project	- No search - No command line - Memory consuming, Static a 100 recs Is reported: - Don't handle deleted record - Overwrite sets - Must be harvested by set
PKP Harvester	- PHP, fast when Search disabled. - Flexible, several API - Supported - Maintained - Search engine	- No Import Is reported: - Deletes 'deleted record' - Overwrites sets - Cannot be harvested by set - New identifier, preserve old for search
Fedora	- Java - Scalable - Several APIs, e.g. Search - Maintained	- No harvester - Content system, extended

7 Installing Repository Software on Windows

We have tested our repositories on both work station and server computers. They all work fine on Windows 2000/XP/2003 servers.
In this setup we have been using Apache as web-server on the windows platform also, this is mainly because we prefer to keep the two environments as identical as possible. However things will also work with IIS on Windows.

Windows requirements:

- Windows 2000/2003 Server or newer
- PHP and Perl needs to be installed, (works equally well on IIS as on Apache)
- If **Repox** is chosen, it will install it's own Java Servlet engine, Jetty.
- If **Celestial** is chosen, Avinet will supply a modified package of this software
- **PKP Harvester** is a PHP product and works as long as PHP is installed
- A database.
    - Repox comes default with **Derby**, a java based RDB.
    - Celestial should work on most databases which support the use of blob or blob-like data types. We have been using **mySql**.
    - PKP Harvester supports numerous databases as long as the php-driver files are installed.
    - *Regarding PKP and PHP. Make sure to enable the following extension in the PHP.ini file: extension=php_mbstring.dll before you import any data.*
- Make sure the computer can access the internet. If this is sensitive data, one might want to setup some type of authentication.

7.1.1 Installation details

Download and install the latest version of Perl. Perl is needed regardless of what repository you choose.
Perl can be found at www.perl.net. Installation should be pretty straight forward. When perl is installed you should have access to cpan. Cpan is a tool for installing additional perl libraries. You need to atleast install the following ones:
Download and install a database. Repox have its own, Derby. Celestial have been tried and tested on mySql and PKP Harvester supports 8 or 9 different ones. We have used mySql and PostgreSQL for PKP Harvester installations.
You do not need to download Apache webserver for Windows. PKP Harvester and Celestial should run on IIS. Repox has its own java server and can run on that.
If you are going to use Repox, download and install the latest java version. From http://java.sun.com/javase/downloads/index.jsp

Installing PKP Harvester on Windows

Download and extract the files into the c:\inetpub\wwwroot directory. Make sure that the cache- and public directory is writeable. Also make sure the config.inc.php file is writeable. Make sure the php-files for your chosen database are installed.

7.1.2 Configuring MySQL

Log into the database, create a user and set a password for it.

```
Mysql –u root.
```

After logging in it would be smart to give the rootuser a password.

```
Set password = password('something');
```

Create a database user for PKP Harvester named **harvester2** :

```
create user harvester2@localhost;
set password for harvester2@localhost = password('harvester2')..
```

7.1.3 Configuring PKP Harvester

After completing the above steps, open your web browser, go to http://localhost/harvester2 and configure the following options:

- Locale settings:
  - Primary Locale English
  - Client Character set Unicode (UTF-8)
  - Connection character set Unicode(UTF-8)
  - Database character set Unicode(UTF-8)
- Security settings
  - SHA1, since it is more secure then MD5
- Administrator Account
  - Type in the username/password you created when you set up the database'
  - Insert an administrator e-mail address
- Database settings.
  - If there are brackets around the database you wanna use, the php drivers for that database has not been installed.
  - Host is usually localhost
  - Insert the username/password you created in the database setup

Push the Install Harvester2 button. The installation process usually takes no longer than 3-4 seconds.
If there are any errors in the data you have inserted you will get the same installation page again with an error message.
If everything looks ok you will get a confirmation page telling you that the installation was successful and a link to the login page.
Login and enjoy

Installing Repox on Windows

Download the installer from http://repox.ist.utl.pt/REPOX_1.2-installer.jar and install it by double clicking the downloaded file.
Follow the on-screen instructions. Make sure the user making the installation has got administrator privileges on the computer where the installation takes place.

On Windows Vista you have to do the following:

Go to Start > All Programs > Accessories > then right-click on "Command Prompt" and select "Run as administrator".

Then go to the directory where you put the file and run

```
java -jar install.jar
```

When installation is done, login – and enjoy.

Installing Celestial on Windows

Download the package containing the Celestial files. A modified version is made available by EuropeanaLocal.
With the files we have supplied some libraries that need to be built and installed. Make sure your system has a make-builder installed. One version can be found here
http://unxutils.sourceforge.net/
Unpack the library file and do the following

```
perl Makefile.pl
Make
make install
```

When that is done, there are some other PERL libraries that needs to be installed, if they are not already present:
XML::Parser::Perlsax, XML::Parser, XML::DOM, XML::Writer

Make sure the user and the table name have been created already in your database of choice.

7.1.4 Configuring MySQL

We assume mySql is being used:

```
mySql –u root
```

When prompted, type in the root password for MySQL determined at time of installation.

```
Create user celestial@localhost
Set password for celestial@localhost = password('password');
Create table celestial
Grant all privilieges on celestial.* to celestial@localhost
```

When this is done, run the installer script

```
perl –w install.pl, and follow the onscreen instructions.
```

After installation. Run http://localhost/celestial/config and set it up to harvest from files or from other repositories.

8 Installing Repository Software on Linux

We have tested all our solutions on both desktop and server versions of Ubuntu linux. And the solutions works fine on either platform.
On both versions, several software packages needs to be added after a clean installation. If you use an existing server, chances are that many of those packages are already installed.
We have used **Apache** as web-server for PKP Harvester and Celestial, alongside the built-in **Jetty** web server that comes with Repox.
As database we have used **MySQL** for PKP Harvester and **Derby** which comes packaged with Repox.

Linux requirements

Linux distribution of choice
PHP and Perl needs to be installed, along with a web server
If Repox is chosen, it will install its own Java Servlet engine, Jetty
If Celestial is chosen, EuropeanaLocal will supply a modified package of this software
PKP Harvester is a PHP product and works as long as PHP is installed
You need a database. Repox comes default with Derby, but should work with other RDB's, like PostgreSQL, MySql, MS SQL Server
Celestial should work on most databases that support the use of blob or blob-like data types.
PKP Harvester supports numerous databases as long as the php-driver files are installed.
*One little thing regarding PKP and PHP. Make sure to install the php extension for **mbstring** before you import any data. E.g:*

```
apt-get install php5_mbstring
```

Make sure the computer has accessible to the internet. If this is sensitive data, one might want to setup some type of authentication.

8.1.1 Installation instructions on Ubuntu platform

Start with a clean ubuntu installation
Type in the nessecary information along the process, language, keyboard, server name, user name and password.
When you get to choose what type of server you would like to install, choose only the ssh-server option.
When all is done it's time to log on with the user you created.
First of all, start with an update.

```
Sudo apt-get update
Sudo apt-get upgrade.
```

Then we start to install necessary software:

```
Sudo apt-get install apache2. This is the webeserver.
Sudo apt-get install php5-cgi Cgi binary for php
Sudo apt-get install perl (Is part of the ubuntu installation)
Sudo apt-get install sun-java6-sdk. Java is only nessecary if you want to
run Repox as repository.
```

Install your *favourite* database for the repository. The solutions we have tried out uses **MySql** and **PostgreSQL**. **Repox** comes with its own RDBMS, **Derby**. PKP Harvester supports a multitude of databases.

```
Sudo apt-get install mysql
```

Install the php databasedriver for your selected database, like php5-mysql.

```
Sudo apt-get install php5-mysql
```

Log on to the database and create a user, and set a password.

```
Mysql –u root
```

After logging in it would be smart to give the rootuser a password.

```
Set password = password('something');
```

Create a user for PKP Harvester:

```
create user harvester2@localhost
set password for harvester2@localhost = password('harvester2')
```

Create a user for Celestial:

```
create user celestial@localhost
set password for celestial@localhost = password('celestial').
```

Now we are ready to install the repositories.

Installing PKP Harvester on Linux

Download the source from http://pkp.sfu.ca/harvester2/download/harvester-2.3.0.tar.gz
Unzip/untar the file and put the contents in your www-directory

```
C:\inetpub\wwwroot or /var/www or similar
```

Make the cache and public area writeable to anyone. To make installation easier also set all rights for everyone on the config.inc-php file. And change it back to read only afterwards.
If you choose to use something else than ubuntu, make sure that the php5_mbstring file is installed.
Start your favorite web browser and go to http://localhost/harvester2
On the Pre-Installation Steps screen - make sure all the lines say "yes".

Configure locale settings

- Primary Locale English
- Client Character set Unicode (UTF-8)
- Connection character set Unicode(UTF-8)
- Database character set Unicode(UTF-8)

Configure Security settings

- SHA1, since it is more secure then MD5

Configure Administrator Account

- Type in the username/password you created when you set up the database
- Insert an administrator e-mail address

Configure Database settings.

- If there are brackets around the database you wanna use, the php drivers for that database has not been installed.

- Host is usually localhost
- Insert the username/password you created in the database setup

Push the Install Harvester2 button. The installation process usually takes no longer than 3-4 seconds.
If there is any errors in the data you have inserted you will get the same installation page again with an error message.
If everything is ok you will get a confirmation page that the installation was successful, and a link to the login page.
Login and enjoy

Installing Repox on Linux

The previous versions of Repox needed access to an x-server in order to be installed. But in the newest relase, an easier way of installing it on Linux has been made.

We will support you with an altered version of Repox which works. This is due to some small alterations that most likely has to be done in the installation script.
Unzip the file REPOX_1.2_unix-installer.zip. You'll end up with a directory named repoxlinux. Move this to the /opt catalogue.
Run the install.sh script and follow the onscreen instructions.

```
Urn:name.of.your.choice
Jetty server port – keep the 8080 port
Data path /opt/repoxlinux
Admin email
User login – keep the admin
User password – whatever you like. Just remember it
```

When the installation is done, Repox will be available on http://localhost:8080 unless you specified something different during installation.
If Repox is the only web application that is going to run on your server, you don't need apache webserver. The Jetty server installed with Repox should be sufficient.
If you are planning to run other web-related material on this server, apache is a must. Then we can "hide" Repox behind the apache-server, and we can access Repox on the standard html port, 80.

To do so we need to make some changes in the httpd.conf

```
LoadModule proxy_module /path/to/modules/mod_proxy.so
ProxyRequests Off
<proxy *>
Order deny,allow
Allow from all
</proxy>
ProxyPass /repox http://localhost:8080/repox
ProxyPassReverse /repox http://localhost:8080/repox
ProxyPreserveHost On
Proxystatus On
```

Installing Celestial on Linux

The original Celestial distribution has been slightly modified by Technical Partners. The modified package is available to partners.

Download the file celestial.tar.gz
Untar the file:

```
tar xzf celestial.tar.gz
```

You will end up with X directories. Move the contents of the bin directory to your cgi-bin catalogue and the data directory to /opt/celestial.
The config file is pre-setup for use with mysql
Included in this package is another tarfile, called oai-perl-2.15.tar.gz. It contains perl libraries that provides an API to OAI-PMH.

```
Tar xzf oai-perl-2.15.tar.gz
```

Cd into the directory that was created, build a *makefile* with the command:

```
perl Makefile.pl
```

Then run make
Then run make install
With the libraries installed, we can turn to the database.

```
perl -w install.pl
```

And follow the onscreen instructions.
After installation. Run http://localhost/celestial/config and set it up to harvest from file or from another repository.

9 Relevant Standards

The following chapter describes relevant standards for the implementation of content provider repositories among EuropeanaLocal content providers across Europe.

XML – eXtensible Markup Language

XML is at the heart of all data exchange in EuropeanaLocal, it is therefore imperative that technical personnel are acquainted with this way of embedding data before attempting to create any extraction scripts and/or ingest data into a repository.
XML is a very simple and flexible markup language and it suffices to be aware of the following conventions and concepts in order to create standards compliant XML. Consider the below example.

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
<namespace:element attribute="attribute value">element value</element>
</root>
<!--Comment -->
```

Let us take a look at the first line:

```
<?xml version="1.0" encoding="UTF-8"?>
```

This line looks like any HTML tag, except inside the tag follows a question mark <?xml … ?>. This signifies the start of an XML formatted text document.
The important thing to be aware of here is the encoding attribute which in the above example is using UTF-8 – a character set capable of storing multi-byte strings, thereby enabling the representation of any special characters found outside the Latin alphabet without breaking. If you input a character into the XML which is not supported by the character encoding given in this line, an error will result.
XML is not tolerant to characters like >, <, & or similar. For this reason, these should be encoded into their corresponding XML-entities during export in order to avoid broken characters in exported files. The entity "**&**" can for an example be represented legally like this "**&**".
Let's move on to the next line:

```
<root>
```

An element is a keyword enclosed by angular brackets "**<>**". A start element may look like this: **<element>** and an end-element may look like this: "**</element>**" with the containing value between them. Empty elements may look like this: "**<element/>**".
The first element following the initial xml-tag, is called the "**root element**". Any subsequent element is known merely as elements.
Continuing to the third line we find the following:

```
<namespace:element attribute="attribute value">element value</element>
```

In this line we see that the element keyword contains a colon. The keyword before the colon indicates the "**namespace"** to which the keyword after the colon belongs. A namespace is a controlled set of element names associated with a "**schema"** which defines permissible values and attributes for the each element.
In addition to the namespace:keyword inside the element tag, we find another keyword followed by an equal sign and a set of double quotes encapsulating a value. This second

keyword is called an "**attribute"**. Attributes may look like this: '**attribute="attribute value"**' and will always exist within elements.
Finally, let us have a look at the notation for adding inline comments to XML markup.

```
<!--Comment -->
```

Any text between a "**<!—**" and "**-->**" will be considered as comments and ignored by XML interpreters. This is handy if we wish to include additional attributes which are not recognized by standards such as ESE 3.1 and Simple Dublin Core.
There is a wide range of good XML-references on the Internet, among which EuropeanaLocal technical partners would recommend w3schools XML-tutorial, easily found using Google. The above will suffice as a basic primer to prepare a basic XML-file. (w3schools.com, 2009)

Dublin Core

Dublin Core describes standardized ways of structuring and naming metadata elements so that they can be read and interpreted properly by Dublin Core aware computer systems.
The following example shows a **Simple Dublin Core** metadata description of the resource found at http://www.europeanalocal.eu, the project web site of EuropeanaLocal.

```
<?xml version="1.0"?>
<metadata
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:dc="http://purl.org/dc/elements/1.1/">
      <dc:title>
      EuropeanaLocal Project Web Site
      </dc:title>
      <dc:description>
      EuropeanaLocal is a project funded under eContentplus, identifying
best practices for inclusion of locally held content into Europeana.
      </dc:description>
      <dc:publisher>
      EuropeanaLocal
      </dc:publisher>
      <dc:identifier>
      http://www.europeanalocal.eu
      </dc:identifier>
</metadata>
```

This is relevant to EuropeanaLocal because the core metadata set determined by Europeana is an extension of Dublin Core by the name of Europeana Semantic Elements, described in greater detail below.
The Dublin Core metadata set is defined and evolved by the Dublin Core Metadata Initiative and its working groups. A lot of useful information as available at the Dublin Core Metadata Initiative's web site including their **Guidelines for implementing Dublin Core in XML** (Powel & Johnston, 2003)

Europeana Semantic Elements

Europeana Semantic Elements is an application profile based on Dublin Core. It defines a number of properties which can be used to describe digital objects. These properties are listed

in the table below. Please notice the XML concepts namespace and element. The column occurrence states whether the element is optional (0..1/0..*) mandatory (1..1/1..*) and whether it can occur only once or multiple times in the XML (1/*)

Table 3: ESE elements

Namespace	Element (Abstract)	Element (Refinement)	Occurrence
Dc dc:terms	Title	 alternative	1..1 0..*
Dc	Creator		0..*
Dc	Subject		0..*
Dc dc:terms	Description	 tableOfContents	0..* 0..*
Dc	Publisher		0..*
Dc	Contributor		0..*
Dc dc:terms dc:terms	date	 created; issued	0..* 0..* 0..*
Dc	type		0..*
Dc dc:terms dc:terms	Format	 extent; medium	0..* 0..* 0..*
Dc	Identifier		1..1
Dc	source		0..*
Dc	Language		0..*
Dc dc:terms dc:terms dc:terms dc:terms dc:terms dc:terms dc:terms dc:terms dc:terms dc:terms dc:terms dc:terms	relation	 isVersionOf; hasVersion; isReplacedBy; replaces; isRequiredBy; requires; isPartOf; hasPart; isReferencedBy; references; isFormatOf; hasFormat; conformsTo	n/a 0..* 0..* 0..* 0..* 0..* 0..* 0..* 0..* 0..* 0..* 0..* 0..*
Europeana europeana Europeana	relation	 isShownBy[1]; isShownAt	n/a 0..1 0..1
Dc dc:terms dc:terms	Coverage	 spatial; temporal	0..* 0..* 0..*
Dc	rights		0..*

[1] Either isShownAt or isShownBy MUST be present in the XML

Dc dc:terms	terms	Provenance	n/a 0..*
Europeana	userTag[1]		0..0
Europeana	unstored[2]		0..1
Europeana	object[3]		0..1
Europeana	language		1..1
Europeana	provider[4]		1..1
Europeana	type[5]		1..1
Europeana	uri[6]		1..1
Europeana	year[7]		0..1
Europeana	hasObject		1..1
Europeana	country		1..1

In addition to the elements, Europeana Semantic Elements also define an attribute which can be used to specify the language of the element values, **xml:lang**. This attribute can be used for all text-based elements with multiple occurrences to state whether the language of the element value is e.g. English (en), French (fr) or German (de).

```
<dc:subject xml:lang="en">weather</dc:subject>
<dc:subject xml:lang="de">wetter</dc:subject>
```

The usage and content of Europeana Semantic Elements is described in greater detail in the two documents **Specification for the Europeana Semantic Elements** (Clayphan, et al., 02/2009) and **Metadata Mapping & Normalisation Guidelines for the Europeana Prototype** (Clayphan, Lupovici, Sugimoto, & Verleyen, 05/2009). Both of these documents are attached as appendices to the implementation manual, see Appendix 3: Europeana Semantic Elements v3.1 and Appendix 4: ESE Implementation Guidelines .

The OAI-PMH web service protocol

While the standards we have been looking at above are related to the metadata content and how it should be formatted, OAI-PMH provides us with a means of transporting metadata from an OAI-PMH compliant provider repository to an OAI-PMH aggregator repository by means of harvesting.
OAI-PMH is a simple web service protocol which can be invoked calling the URL of an installed repository with one out of six arguments – called verbs. Each of these will return a specifically formatted XML string to the requestor web service, making it possible to interpret the nature and quantity of the content held in the provider repository as well as downloading its metadata content. The six verbs are as follows:

---

[1] This element is only meant to store user tags contribute through the Europeana application and does not need to be included in the content providers' XML

[2] Include any metadata properties which doesn't fit into the other elements but which may be interesting to include for indexing purposes in the "unstored" element.

[3] The object element will be populated by Europeana if left out from the XML

[4] The provider element will be populated by Europeana if left out from the XML

[5] One of TEXT, IMAGE, SOUND or VIDEO, Europeana controlled vocabulary

[6] Will be added automatically by Europeana

[7] Four digit year. Currently no support for dates before year 0. Planned to be implemented shortly.

1. Identify
2. ListMetadataFormats
3. ListSets
4. ListIdentifier
5. ListRecords
6. GetRecord

The most important of these verbs is "ListRecords" as this is the one returning actual metadata records from the source repository. This is the verb requested by harvesters who connect to source repositories, EuropeanaLocal content providers must therefore make sure that the response is properly formatted in accordance with the XML format rules as shown below.

```
<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH>
<ListRecords>
<record>
<header>
      <identifier>
      <dateStamp>
      <setSpec>
</header>
<metadata>
      <!—ESE metadata elements here -->
</metadata>
<about>
      <rights>
      <provenance>
</about>
      </record>
</ListRecords>
</OAI-PMH>
```

Below you will find a description of the most important elements of the OAI-PMH ListRecords XML format.

9.1.1 Identifier

To be able to separate between updated and added records during incremental harvesting, it is necessary that a unique identifier for each item exists in the provider repository.
OAI-PMH allows for identifiers which are not directly related to physical locations, but instead uses a system of locally unique identifiers within a repository. The format of the unique identifier must correspond to that of the URI (Uniform Resource Identifier) syntax. The syntax of an identifier is:
oai-identifier = scheme : namespace-identifier : local-identifier

The scheme is always the literal string "oai". The namespace-identifier is usually related to the domain-name of the organization hosting the repository.
The local-identifier is some identifier which is unique in the context of this particular repository. An example of an identifier from ArXiv.org might look like this: oai:arXiv.org:hep-th/9901001.

Globally unique identifiers are useful in that they make it possible to trace a metadata record back to its source repository, and because they can be used to build rich and interlinked services based on metadata harvested from different repositories.

9.1.2 Timestamps

The from/until parameters of the harvesting verb ListRecords is related to the dateStamp element with granularity of day or second.

9.1.3 Sets

Sets are an optional feature of OAI-PMH offering setSpec in the header section to divide the records reflecting e.g. the organizational or thematic structure of the records. As these are optional, they are not always preserved by aggregator repositories and EuropeanaLocal technical partners cannot at this moment recommend their use for any purpose outside of the provider repository itself except to make subsets of the repository contents harvestable for different aggregators.

9.1.4 Metadata

The metadata section of the record holds the metadata in the actual format. The XML is for some repositories stored as is, in other as occurrences of an element table.

9.1.5 About (Provenance)

About is 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. About is mainly used as tracking feature in addition to the identifier.

9.1.6 Flow control

The List verb of the OAI-PMH protocol offers use of resumption tokens to split the transfer of metadata in smaller chunks. This feature is optional for the provider, but mandatory for the harvester.

9.1.7 Deleted records

OAI recommends to handle deleted record marked as attribute status=deleted of the header tag. The Identify verb tells if a repository offers this service or not.
Alternative solutions for this service:

1. The proprietary system marks deleted records as deleted instead of physically removing them.
2. The proprietary system logs deleted records.
3. An additional system is added on top of the proprietary system to keep track of deletions.

OAI-PMH is described in greater detail in the **Implementation Guidelines for the Open Archives Initiative Protocol for Metadata Harvesting** which can be found at the Open Archive Initiative's web site. (Lagoze, Van de Sompel, Nelson, & Warner, 2005).

10 Metadata Extraction and Normalization

Once the repository software has been successfully installed, the next step of the process is to populate the repository with data extracted from local collection management systems. Should your local collection management system have an export feature to a well-known format such as MARC or Dublin Core – or an export to some XML-based format, your best bet might be to go for this. However, most of the data held by regional and local providers originates from legacy databases and systems without any such features. For such systems there are no other ways than writing a custom extraction script to create an ESE formatted XML representation of the source data.

Metadata extraction

While most of the legacy systems and databases lack any harmonized structure, they are mostly accessible via an ODBC driver which enables programmers to connect to them and read content values.
It is too comprehensive a task for this manual to go into the details of the SQL language and it is assumed that technical personnel performing the metadata extraction is acquainted with this language and technology. A good references for SQL is to be found at w3schools.com, (w3schools.com, 2009)
Extraction scripts can be written in any language but most likely one of the web scripting languages such as PERL, PHP, ASP, Python, VB.NET, C# or Java will be the easiest solution. After having connected to the database, the extraction script has to do three things:

- Map metadata fields from the source system to ESE
- Normalize any non-standard metadata values
- Create an OAI ListRecords embedded ESE XML file

Metadata Field Mapping

The script must interpret which fields in the source database corresponds t which fields in the target metadata application profile Europeana Semantic Elements. This may be a task that requires considerable manual efforts and experimentation in order to understand the practical consequences of the mapping decisions. In this respect, the ESE metadata elements description and implementation guidelines will be of great value. See: Appendix 3: Europeana Semantic Elements v3.1 and Appendix 4: ESE Implementation Guidelines

Metadata Normalization

In addition to reading raw element values from the source database and putting them into corresponding ESE fields in a text file, the script must also normalize values. The example of normalization is the adaption of the date format used in the source database to the one specified by ESE.
Dates can be expressed in a multitude of different ways including the following, which are all different ways of specifying the 1st of December 2009:

- DD/MM/YYYY – 01/12/2009
- YYYY/DD/MM – 2009/01/12
- YYYY/DD/MM – 2009/12/01
- MM/DD/YY – 12/01/09

In order for a source system to successfully use dates for visualization and retrieval purposes, it is a strict requirement that date values are normalized.
Another important thing to normalize is the codes used to reference languages. Here Europeana recommends the use of RFC 4646 (Phillips & Davis, 2006) ISO 639 (Library of Congress, 2008) which defines complete two and three letter codes for each country and language..
Normalization extends far beyond dates, countries and languages, but once we enter this territory, it is a requirement that quality assured authority files and thesauri exist from which we can pick the "normal" value to which other references should be normalized.
This include person names, names of time periods, geographical names and more. All of these issues will be covered in greater detail in later revisions of this manual. For the time being, it is required that
No thesauri has been agreed upon for use within Europeana but out of those in existence the Getty vocabularies (The J. Paul Getty Trust, 2009)are possibly the most promising and comprehensive resource available and thus a possible starting point. The time and resources required to perform high quality mapping of content values is however vast and it is beyond the scope of EuropeanaLocal to embark on this task. The issue will however be covered by EuropeanaLocal technical workshops.

Writing the XML format

Having determined which values goes where and how they must be split, merged and normalized to correspond to Europeana Semantic Elements, the next step is to create the actual XML file which in turn will be ingested into the repository software package installed in chapter 7 above.
We do this by writing a format which embeds ESE within the OAI-PMH ListRecords format. The following template may be used and extended with any of the metadata fields listed in chapter 0 above.

```
<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH>
<ListRecords>
<record>
<header>
<identifier>oai:sffKultur.no:SFFkl-100445</identifier>
<dateStamp>2009-02-17</dateStamp>
<setSpec>Kultur</setSpec
</header>
<metadata>
<ese:europeana xmlns:ese="http://europeana.eu/terms"
xmlns:europeana="http://localhost/"
xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:dcterms="http://purl.org/dc/terms/"
xmlns:dcmitype="http://purl.org/dc/dcmitype/" schemaLocation="
http://europeana.eu/ese.xsd">
<dc:publisher xml:lang="no">Fylkesarkivet i Sogn og Fjordane</dc:publisher>
<dc:contributor>Hermund Kleppa</dc:contributor>
<dcterms:issued>2008-11-27</dcterms:issued>
<dcterms:modified>2009-02-17</dcterms:modified>
```

```
<dc:type>Text</dc:type>
<dc:identifier><![CDATA[http://www.sffarkiv.no/sffbasar/default.asp?p=result&db=dbatlas_leks&art_id=1280&spraak_id=1&ptype=single]]></dc:identifier>
<dc:source xml:lang="no">Fylkesarkivet i Sogn og Fjordane</dc:source>
<dc:language>no</dc:language>
<europeana:isShownAt>http://www.sffarkiv.no/sffbasar/default.asp?p=result&amp;db=dbatlas_leks&amp;art_id=1280&amp;spraak_id=1&amp;ptype=single</europeana:isShownAt>
<dcterms:spatial type="text">Fylke 1400</dcterms:spatial>
<dcterms:spatial type="text">Kommune 1430</dcterms:spatial>
<dcterms:spatial type="sfs-wkt">SRID=32633;POINT(8361 6822394)</dcterms:spatial>
<dc:rights xml:lang="no">Fylkesarkivet i Sogn og Fjordane</dc:rights>
<europeana:unstored xml:lang="no">Vegen forbi Matbjørhammaren - smal, svingete og rasfarleg.</europeana:unstored>
<europeana:type>TEXT</europeana:type>
</ese:europeana>
</metadata>
</record>
```

11 Testing your Repository

Having completed the previous steps, only one thing remains: to test that your repository is working. To do so, EuropeanaLocal technical partners provide a test harvester which you can use to test that your repository is providing valid responses.
The test harvester will connect to your supplied repository URL and issue harvest requests using the ListRecords verb. If no error messages are returned, the repository is valid and may be forwarded to Europeana for ingestion into the sand-box environment where the content provider can see how the exported metadata appears within the Europeana service.
Once satisfied, the repository will be moved to the Europeana prototype and the data store for the upcoming release of Europeana Version 1.0 scheduled for the spring of 2010.
The test harvesting environment is available at the following URL:
http://utvikling.avinet.no/europeanalocal/harvester2
Additionally, the Open Archives Initiative Repository Explorer (Department of Computer Science, University of Cape Town, 2006) may be used to verify the functionality of your provider repository – though this does not allow you to do searches in the harvested data – merely validate the output of your provider repository.
The harvester is using the PKP Harvester Repository Software Package. Data will be purged at regular intervals and EuropeanaLocal technical partners offer no particular service level agreement for this environment which in time will be replaced by the Europeana test environment once completed.

12 Troubleshooting and Support

Should partners upon attempt to implement the tasks described in this manual run into problems, EuropeanaLocal offers a number of support options described in greater detail below.

EuropeanaLocal Support Forum

The online support option for EuropeanaLocal is currently available at http://www.avinet.no/forum and will by August also be available through the europeanalocal.eu domain used by the project web site. All partners will receive usernames and passwords for the forum to the administrative and technical contact e-mail addresses supplied in the technical questionnaire.
The online support forum consists of five sections out of which two are open for posting from all users (shown in **bold**).

- Frequently asked questions: a moderator run section where recurring questions and issues are highlighted
- **Repository installation questions**: a community run section where content providers exchange knowledge among each other supported by the technical partners
- **Metadata extraction, normalization and enrichment questions**: a community run section where content providers exchange experience on extraction, normalization and enrichment issues supported by the technical partners
- Documentation and utilities: a moderator run section where EuropeanaLocal documentation (like this implementation manual and future revisions) will be made available for download.
- Information from Project Management Board: a non-technical section where content provider coordinators can give feedback to EuropeanaLocal participants on overall project progress and related issues.

Other support options

The web based support forum is preferable due to the fact that questions directed there will benefit not only the originator but also others who experience the same problem. However, certain issues may be of interest only to individual content providers, in such cases we offer additional support options.

12.1.1 E-mail support

Where a content provider partner may not wish to expose his/her issues to other partners, e-mail support is an alternative to using the web support forum. Questions will be answered and dealt with in a similar fashion as for the support forum, but will not be public. Technical partners wish to publish questions deemed to be of general interest subject to consent from the originator, applying anonymization where required to protect privacy and/or security.
Questions may be asked in English, Slovakian, Norwegian, Danish or Swedish to the following e-mail address:

europeanalocal@avinet.no

E-mails sent to this address will be distributed to the support-team in Avinet and EEA and the reply will come directly from either of these organizations or be forwarded to the relevant source if appropriate.

12.1.2 Phone support

Phone support may be agreed upon between content providers and technical partners via e-mail and will be available from EEA and Avinet subject to limitations due to budget and resource constraints.

12.1.3 On-site support visits

While deemed a non-strategic support mechanism as it is confined to a very limited number of beneficiaries, on-site support visits remains an option. However, as an alternative, EuropeanaLocal has invited technical experts from across Europe to the technical workshops, thereby ensuring a wide-spread network of OAI-PMH and ESE experts throughout the partner regions. It is preferable that partners utilize these resources as this has a more long-term strategic aspect to it than one-of visits from technical partners.
In very particular cases, the project budget gives provisions for a limited number of onsite support visits. These must be sanctioned by the Project Management Board and will only be used most or all of the below criteria are satisfied:

1. A partner has un-resolved technical issues
2. The resolution of the issues are time-critical to the project
3. The resolution of the issues cannot be satisfied by any other support options

Bibliography

Clayphan, R., Dekkers, M., Fernie, K., Gradmann, S., Lupovici, C., Meghini, C., et al. (02/2009). *Specification for the Europeana Semantic Elements v3.1.* Den Haag: European Digital Library Foundation (Europeana).

Clayphan, R., Lupovici, C., Sugimoto, G., & Verleyen, J. (05/2009). *Metadata Mapping & Normalisation Guidelines for the Europeana Prototype.* Den Haag: European Digital Library Foundation (Europeana).

Department of Computer Science, University of Cape Town. (2006, 12). *Open Archives Initiative - Repository Explorer - Version - 1.46c.* Hentet 04 2009 fra OAI Repository Explorer: http://re.cs.uct.ac.za/

Lagoze, C., Van de Sompel, H., Nelson, M., & Warner, S. (2005, 05 03). *Implementation Guidelines for the Open Archives Initiative Protocol forMetadata Harvesting v2.0.* Hentet 11 04, 2008 fra OAI-PMH Implementation Guidelines: http://www.openarchives.org/OAI/2.0/guidelines.htm

Library of Congress. (2008, 11 07). *Codes for the Representation of Names of Languages.* Hentet fra ISO 639-2 Language Code List - Codes for the representation of names of languages: http://www.loc.gov/standards/iso639-2/php/code_list.php

Phillips, A., & Davis, M. (2006, 09). *RFC 4646 - Tags for Identifying Languages.* Hentet fra Network Working Group, Request for Comments: 4646: http://www.ietf.org/rfc/rfc4646.txt

Powel, A., & Johnston, P. (2003, 04 02). *Guidelines for implementing Dublin Core in XML.* Retrieved 04 10, 2009, from Dublin Core Metadata Initiative: http://dublincore.org/documents/dc-xml-guidelines/

Public Knowledge Project. (2009). *Open Archives Harvester .* Hentet fra Public Knowledge Project: http://pkp.sfu.ca/?q=harvester

TEL+ Project. (2009). *REPOX A Metadata Space Manager.* Hentet fra REPOX: http://repox.ist.utl.pt/index.html

The J. Paul Getty Trust. (2009). *Learn about the Getty Vocabularies .* Hentet 04 2009 fra Getty Vocabulary Program (Research at the Getty):: http://www.getty.edu/research/conducting_research/vocabularies/

w3schools.com. (2009). *SQL Tutorial.* Hentet 04 05, 2009 fra w3schools.com: http://www.w3schools.com/sql/default.asp

w3schools.com. (2009). *XML Tutorial*. Hentet 04 02, 2009 fra w3schools.com: http://www.w3schools.com/xml/default.asp

Tables and Diagrams

Table 1: Feature comparison matrix for assessed repository software packages.....15
Table 2: Advantages and drawbacks of repository software packages 16
Table 3: ESE elements 27

Appendices

Appendix 1: Dublin Core Metadata Elements

Appendix 2: OAI Protocol for Metadata Harvesting

Appendix 3: Europeana Semantic Elements v3.1

Appendix 4: ESE Implementation Guidelines