

Demos@Work

Enable European-wide discussion between elected representatives and civil society

Project EP-07-01-035

Demos@Work

Installation Manual

Responsible : ScytI

Contributors: Gov2U, GFI, NTUA

Document Reference :

Dissemination Level : PU

Version : 1.0

Date : 23th October 2008



TABLE OF CONTENTS

I. REQUIREMENTS.....	3
II. INSTALLATION PROCESS.....	4
II.1. Installation and configuration of DemosMP.....	4
II.1.1. Portal:	4
II.1.1.a. Configuration file.....	6
II.1.1.b. Language Configuration.....	8
II.1.1.c. Language Selection	8
II.1.2. eRepresentative directory web service.....	9
II.1.2.a. Directory web service.....	9
II.1.2.a.1. Web service objects.....	9
II.1.3. LDAP	15
II.1.3.a. Directory objects.....	16
II.1.4. eRepRepository Webservice	17
II.1.4.a. eRepAddDocument	17
II.1.4.b. PHP5 webservice client example	23
II.1.5. DSPACE	25
II.2. Installation and configuration of DemosCitizen	26
II.2.1. Portal:	26
II.2.1.a. Language Configuration.....	27
II.2.1.b. Language Selection.....	27
II.2.2. Semantics Module Initializer & Semantics Module	28

I. REQUIREMENTS

Demos@Work is composed of two main components which have well defined boundaries and objectives: DemosCitizen & DemosMP. Each consists of several modules with several different requirements as detailed below.

DemosCitizen requires:

- **Apache 2** web server with support for **PHP 5.2** and **SOAP module** active for Apache.
- A **MySQL 5** or later database server.
- A **Glassfish 2** server for running the Java Web Applications: Semantic Module and its Initializer.
- An **Ontology File** (.owl File) containing the ontology terms that will be imported to MySQL Database via the Semantics Module Initializer.

DemosMP requirements are:

- **Apache 2** web server with support to **PHP 5.2** and **SOAP module** active for Apache and **phpCAS** active for PHP.
- **CAS** server.
- A **MySQL 5** or later database server.
- Java Web App server **Tomcat 5.5.27** and **PostgreSQL 8.1**
- **OpenLDAP**
- **OpenSSL**

II. INSTALLATION PROCESS

First of all, the server environment must be set up and properly configured for running PHP and Java Web Applications on the software described above in the requirements section, detailed documentation can be found on the software's projects homepages and in many independent online websites as well.

Homepages:

Product	Homepage
Apache	http://httpd.apache.org
PHP	http://www.php.net
SOAP Module	http://www.php.net/manual/en/book.soap.php
MySQL	http://dev.mysql.com/
Glassfish	http://www.glassfish.org
phpCAS	http://www.ja-sig.org/wiki/display/CASC/phpCAS
Tomcat	http://tomcat.apache.org/
PostgreSQL	http://www.postgresql.org/
OpenLDAP	http://www.openldap.org/
OpenSSL	http://www.openssl.org/
Joomla!	http://www.joomla.org/
CAS server	http://www.jasig.org/cas
DSpace	http://www.dspace.org/

II.1. Installation and configuration of DemosMP

As this component is composed of several different elements it will be detailed how to install and configure them separately.

II.1.1. Portal:

To install the portal please copy the portal files to the defined Apache virtual directory (by default is /var/www/). Concerning its configuration two files are used:

- configuration.php
- forum/Settings.php

These files must be present in Apache virtual directory. Their content variables are defined as follows:

- Configuration.php: this file can be found at the root of the portal and the following variables must be changed according to the server specifications:
 - \$mosConfig_absolute_path – absolute path to the portal root
 - \$mosConfig_db – the name of the database that the portal is stored in
 - \$mosConfig_host – the database host
 - \$mosConfig_live_site – the URL of the live site
 - \$mosConfig_locale – the locale has to be changed according to the language of the portal (e.g. ca_CA for Catalan)
 - \$mosConfig_mailfrom – the email address from which emails will be sent
 - \$mosConfig_sitename – the name of the site (default: Demos@Work)
 - \$mosConfig_smtphost – The smtp host for sending emails
 - \$mosConfig_smtpuser – The username for the smtp host
 - \$mosConfig_smtppass – The password for the smtp host
 - \$mosConfig_user - The database username
 - \$mosConfig_upload_agenda – The folder that agendas are saved in according to the agenda upload web service (default: /collaboration/agenda/)
 - \$mosConfig_upload_minutes – The folder that meeting minutes are saved in according to the meeting minutes upload web service (default: /collaboration/minutes/)
 - \$eRepLDAPServer - the host of the LDAP server (default: demosatworkmp)
 - gov2demossIntegration – a flag that indicates if the integration with Gov2DemOSS is active
 - \$gov2demossDBHost – the host where Gov2DemOSS resides
 - \$gov2demossDBName – the name of Gov2DemOSS' database
 - \$gov2demossDBUsername - the username of the Gov2DemOSS' database
 - \$gov2demossDBPassword – the password of Gov2DemOSS' database
 - \$gov2demossSSH2Login – the username of SSH2 access to Gov2DemOSS
 - \$gov2demossSSH2Password – the password of the user for SSH2 access to Gov2DemOSS
 - \$gov2demossSSH2RemoteDirectory – the remote directory where the documents published from DemosAtWorkMP are saved to Gov2DemOSS
- Settings.php: this file contains properties for the forum:
 - \$mbname – Name for the forum (default: Discussion space)
 - \$language – forum language (catalan for Catalan and so on)
 - \$boardurl - the URL under which the forum will be situated
 - \$webmaster_email - the webmaster's email
 - \$db_server – the host of the database
 - \$db_name – database name
 - \$db_user – the database username
 - \$db_passwd – the database password
 - \$boarddir - the absolute path to the forum
 - \$sourcedir – the absolute path to the forum source files

Additionally, the following database tables should be edited:

- jos_desitemanager_site - for the 1st, 2nd and 3rd entry the following fields must be changed:

- path – absolute portal path
- url – URL to live portal
- database host – database host (localhost)
- database_username – database username
- database_password – database password
- jospda_desitemanager_site – same as above
- smf_settings: the following settings can be configured from the forum administration as a system administrator
 - smileys_url – URL to the live portal smileys folder
 - avatar_url – URL to the live portal avatar folder
 - attachmentUploadDir – absolute path to the attachment uploading folder
 - smileys_dir – absolute path to the smileys folder
 - avatar_directory – absolute path to the avatar folder
- smf_themes: the following settings can be configured from the forum administration as a system administrator
 - theme_url – URL to live portal theme folder
 - images_url – URL to live portal images folder
 - theme_dir – absolute path to the portal theme folder
- jos_jsmf_config: in the 1st row of the table, the config value must be changed to the absolute path of the forum

II.1.1.a. Configuration file

For the configuration of the Document Repository part of Demos@Work a single configuration file is used: 'erep.cfg'. This file has to be available in the directory /home/erep/config on the 'DemosAtWorkMP' machine.

The comments in this configuration file describe the purpose of each configuration item:

```
# The host name of the eurovoc database
eurovoc_host = localhost

# The name of the eurovoc database
eurovoc_db = eurovoc

# The user name to access the eurovoc database
eurovoc_usr = eurovoc

# The password to access the eurovoc database
eurovoc_pwd = eurovoc

# The host name for the repository web services
repos_host = demosatworkmp

# The port number for the repository web services
repos_port = 8443

# The user name to access the repository web services
repos_usr = erepclient

# The password to access the repository web services
```

```
repos_pwd
# The ldap host connect string
ldap_connect = erepldap
# The connect string to bind the ldap server
ldap_bind = uid=eRepHP,ou=Managment,dc=eRep,dc=com
# The password to access the ldap server
ldap_pwd
# The maximum size for uploaded files
maxFileSize = 536870912
# The host name for the CAS services
cas_host = demosatworkmp
# The port number for the CAS services
cas_port = 8443
# Default collection name (DO NOT CHANGE)
default.collection.name = eRepresentative
# Default community name
default.community.name = eRepresentative
# Default format to use for date values
default.format.date = yyyyMMdd
# Default format to use for timestamp values
default.format.datetime = yyyyMMdd HH:mm:ss
# Default format to use for time values
default.format.time = HH:mm:ss
# Default language to use for DC values
default.language = en
# DSpace admin email address
dspace.admin.email = erep.admin@localhost
# DSpace admin password
dspace.admin.password
# Maximum number of documents to be logged
max.logged.documents = 10
# Maximum length of a status message
max.status.message.length = 255
```

On the 'DemosAtWorkMP' machine the erep.cfg file has to be available from the classpath in the Tomcat webservices application. This is done by modifying the file 'catalina.sh' in the Tomcat bin directory with the addition of the following line:

```
CLASSPATH=/home/erep/config:"$CLASSPATH"
```



On the 'DemosAtWork' machine the /home/erep/config directory contains two additional configuration files:

- dspace.cfg - A standard Dspace configuration file that defines the location of the stored documents and the credentials of the PostgreSQL database that is used by Dspace.
- log4j.properties - A standard log4j file that defines the level of logging used by the DemosAtWorkMP webservices. The default output for logging is the file 'erepresentative.log' in the Tomcat log directory.

Please note that the default location of the erep.cfg file can be changed by a Joomla administrator

II.1.1.b. Language Configuration

The DemosAtWork uses different language files for its components. These language files are located in the "language" folder for each component. For example:

```
/language – main portal language file  
/components/com_extcalendar/language  
/components/com_letters/language  
/components/com_processes/language  
/administrator/components/com_docman/language  
/forum/Themes/default/languages...
```

In order for a language to function properly, a php file with the language as name has to be created. You can also duplicate the English file and rename it to catalan.php for Catalan. If a language file does not exist in one of the language folders, then the system will use the english.php file by default.

Demos@Work comes with English, Catalan and English language files and content.

II.1.1.c. Language Selection

From the Joomla control panel it is also possible to change the language used by the portal pages. Select the Language Manager from the control panel, then choose a language from the list and click Publish.

For further information on Joomla! configuration, please see Joomla! documentation at the project homepage: <http://docs.joomla.org/>

II.1.2. eRepresentative directory web service

It is necessary to deploy this webservice on Tomcat. Once deployed you may access it via HTTPS and the default port: 8443 (i.e. <https://localhost:8443>).

II.1.2.a. Directory web service

The management of the eRepresentative directory information is implemented through web service interfaces. It is not possible to change this information using the LDAP service and therefore any change must be implemented through this web service.

The interfaces provided are:

- eRepAddMember: Adds new elected representative members to the directory.
- eRepModifyMember: Modifies attributes (with the exception of the member UID) of existing elected representative members.
- eRepRemoveMember: Deletes members from the directory and his/her memberships in the committees.
- eRepValidateMember: Returns the information of elected representatives (with the exception of the password) stored in the directory.
- eRepModifyMemberPassword: Changes the password of existing elected representative members.
- eRepAddCommittee: Adds new committees to the directory.
- eRepAddCommitteeMember: Adds elected representatives to committees.
- eRepModifyCommitteeChairman: Adds or replaces chairmen of committees.
- eRepRemoveCommitteeMember: Deletes members of committees.
- eRepRemoveCommittee: Deletes committees.
- eRepValidateCommittee: Returns the information of the committee requested.

II.1.2.a.1. Web service objects

The web service mainly manages two specific eRepresentative objects related to the LDAP directory objects:

- eRepMember: Object that contains the attributes that identify elected representative members.
- eRepCommittee: Object that contains the attributes that identify the committees and their committee members and chairman.

The attributes contained in the objects are the following ones.

eRepMember

The attributes and types of the eRepMember object are:

Attribute	Type	Number	Description
eRepUID	String	1	Unique identifier used for login
userPassword	Password	1	Password used for login
eRepCn	String	1	Name of the elected representative
eRepRole	String	0..n	Role in the assembly (e.g., President, Secretary)
eRepMail	String	1	E-mail address
eRepTelephon eNumber	String	1	Mobile phone number (for SMS)
eRepMilitancy	String	1	Political party
userPKCS12	String	0..1	Base64 formatted PKCS#12 containing the private key
userCertificate	String	0..1	PEM formatted X509v3 digital certificate of the public key

The number field specifies if the object can contain one or more attributes of the same type (e.g., the object can contain 1 or more role attributes but only one UID attribute). If the number is 0..1 means that the attribute is not mandatory.

eRepCommittee

The attributes and types of the eRepCommittee object are:

Attribute	Type	Number	Description
eRepGID	String	1	Unique identifier for the committee
eRepCn	String	1	Name of the Committee
description	String	0..1	Main function of the committee
eRepChairman	String	1	UID of the representative that manages the committee
memberUid	String	1..n	UID of the representative that belongs to the

committee

This object can contain one or more member attributes. The committee member that has the chairman role of the committee also has a member attribute.

Web service Interfaces

The parameters that will be requested by these interfaces are the following ones.

The number field is related to the number of parameters that can be introduced in the same call to the interface (e.g., in one call to the eRepAddMember interface the client can send one or more eRepMember objects).

The type field can contain an attribute of the directory or a complete object. When an object is specified, it must be compliant with the object structure defined in the directory.

eRepAddMember

This interface generates a new eRepMember object in the LDAP directory. If the member exists (i.e., there is another eRepMember with the same eRepUID) an error is returned.

Parameter	Type	Number	Comments
eRepMember	eRepMember object	1	The object must have all the mandatory attributes specified.

eRepModifyMember

This interface allows the modification of one or more attributes (except the eRepUID) of an existing eRepMember object in the LDAP directory. If the member does not exist (i.e., there is no eRepMember with the eRepUID) an error is returned.

Parameter	Type	Number	Comments
eRepMember	eRepMember object	1	The object must have all the mandatory attributes specified. The eRepUID must correspond to the unique identifier of an existing member inside the LDAP



			directory, otherwise an error is returned.
--	--	--	--

eRepRemoveMember

This interface allows the deletion of an existing eRepMember in the LDAP directory. If the member does not exist (i.e., there no eRepMember with the eRepUID) an error is returned.

If the eRepMember belongs to one or more eRepCommitte objects, the memberId attribute with the eRepUID of this member is eliminated from these eRepCommitte objects. If the member is a chairman of the committee (i.e, the eRepChairman of a eRep Committee object has the same eRepUID), an error is returned and the deletion cancelled. The chairman of the committee must be first modified before accepting the elimination of the member.

Parameter	Type	Number	Comments
eRepUID	string	1	The eRepUID must correspond to the unique identifier of an existing member inside the LDAP directory, otherwise an error is returned.

eRepValidateMember

This interface returns the information, with the exception of the password, of an existing eRepMember in the LDAP directory. The password attributed is left blank to prevent any problem. If the member does not exist (i.e., there no eRepMember with the eRepUID) an error is returned.

Parameter	Type	Number	Comments
eRepUID	string	1	The eRepUID must correspond to the unique identifier of an existing member inside the LDAP directory, otherwise an error is returned.

eRepModifyMemberPassword



This interface modifies the password, of an existing eRepMember in the LDAP directory. If the member does not exist (i.e., there no eRepMember with the eRepUID) an error is returned.

Parameters	Parameters	Number	Comments
eRepUID	string	1	The eRepUID must correspond to the unique identifier of an existing member inside the LDAP directory, otherwise an error is returned.
Password	string	1	

eRepAddCommittee

This interface generates a new eRepCommittee object in the LDAP directory. If the committee exists (i.e., there is another eRepCommittee with the same eRepGID) an error is returned.

Parameter	Type	Number	Comments
eRepCommittee	eRepCommittee object	1	The object must have all the mandatory attributes specified. The eRepGID must correspond to the unique identifier of an existing committee inside the LDAP directory, otherwise an error is returned.

eRepAddCommitteeMember

This interface adds a new member to an existing eRepCommittee in the LDAP directory. This operation fails if:

- The committee does not exist (i.e., there no eRepCommittee with the eRepGID)
- There is no member in the LDAP directory with the specified memberId identifier (i.e., an eRepMember with an eRepUID equal to the memberId).
- The memberId already exists as a member of the committee.

When fails an error is returned specifying the cause.

Parameter	Type	Number	Comments
-----------	------	--------	----------



eRepGID	string	1	eRepGID and a eRepMember with a memberUid as eRepUID, must exist in the LDAP directory.
memberUid	string	1	

eRepModifyCommitteeChairman

This interface changes the chairman of an existing eRepCommittee in the LDAP directory. This operation fails if:

- The committee does not exist (i.e., there no eRepCommittee with the eRepGID)
- There is no member in the LDAP directory with the specified eRepChairman identifier (i.e., an eRepMember with an eRepUID equal to the eRepChairman).
- The eRepChairman is not an existing member of the committee (i.e., a memberUid equal to the eRepChairman identifier).

When fails an error is returned specifying the cause.

Parameter	Type	Number	Comments
eRepGID	string	1	The eRepChairman must be a member of the eRepGID committee.
eRepChairman	string	1	

eRepRemoveCommitteeMember

This interface removes a member of an existing eRepCommittee in the LDAP directory. This operation fails if:

- The committee does not exist (i.e., there no eRepCommittee with the eRepGID)
- The member is also the chairman of the committee (i.e., the memberUid and the eRepChairman of the committee have the same value).
- The memberUid does not exist as a member of the committee.

When fails an error is returned specifying the cause. When the member is also the chairman, the committee chairman must be changed (using the eRepRemoveCommitteeMember) before calling this interface. If memberUid is the last member of the committee this member only can be eliminated when deleting the committee (eRepRemoveCommittee) or after adding new members and assigning a new chairman.

Parameter	Type	Number	Comments
eRepGID	string	1	The memberUid cannot be also

memberUid	string	1	the eRepChairman of the eRepGID committee.
-----------	--------	---	--

eRepRemoveCommittee

This interface removes an existing eRepCommittee object of the LDAP directory. If the committee does not exist (i.e., there is no eRepCommittee with the same eRepGID) an error is returned.

Parameter	Type	Number	Comments
eRepGID	string	1	The eRepGID must correspond to the unique identifier of an existing committee inside the LDAP directory.

eRepValidateCommittee

This interface returns the information of an existing eRepCommittee in the LDAP directory. If the committee does not exist (i.e., there no eRepCommittee with the eRepGID) an error is returned.

Parameter	Type	Number	Comments
eRepGID	string	1	The eRepGID must correspond to the unique identifier of an existing committee inside the LDAP directory.

II.1.3. LDAP

Access to the directory is twofold:

- LDAP read-only access to allow to request elected representative information and/or committee information
- Webservice access to allow update the information of committees and elected representative members

To access to both services, clients need to connect to IP address of the Tomcat server.

LDAP direct access is restricted read-only. This access is provided for the eRep internal modules that need to retrieve elected representative and committee information. Any change

or addition of information must be implemented using the interface provided by the web services.

Since access is encrypted using SSL, protocol used must be SLDAP. The port assigned to access the service is 636

II.1.3.a. Directory objects

The directory provides mainly access to two specific eRepresentative objects:

- eRepMember: Object that contains the attributes that identify elected representative members.
- eRepCommittee: Object that contains the attributes that identify the committees and their committee members and chairman.

The attributes contained in the objects are the following ones.

eRepMember

The attributes and types of the eRepMember object are:

Attribute	Type	Number	Description
eRepUID	String	1	Unique identifier used for login
userPassword	Password	1	Password used for login
eRepCn	String	1	Name of the elected representative
eRepRole	String	0..n	Role in the assembly (e.g., President, Secretary)
eRepMail	String	1	E-mail address
eRepTelephoneNumber	String	1	Mobile phone number (for SMS)
eRepMilitancy	String	1	Political party
userPKCS12	String	0..1	Base64 formatted PKCS#12 containing the private key
userCertificate	String	0..1	PEM formatted X509v3 digital certificate of the public key

The number field specifies if the object can contain one or more attributes of the same type (e.g., the object can contain 1 or more role attributes but only one UID attribute). If the number is 0..1 means that the attribute is not mandatory.

eRepCommittee

The attributes and types of the eRepCommittee object are:

Attribute	Type	Number	Description
eRepGID	String	1	Unique identifier for the committee
eRepCn	String	1	Name of the Committee
description	String	0..1	Main function of the committee
eRepChairman	String	1	UID of the representative that manages the committee
memberUid	String	1..n	UID of the representative that belongs to the committee

This object can contain one or more member attributes. The committee member that has the chairman role of the committee also has a member attribute.

For more detailed instructions in configuring LDAP please see the project documentation at its homepage (detailed in the beginig of the manual).

II.1.4. eRepRepository Webservice

It is necessary to deploy this webservice on Tomcat. The interface to the eRepresentative document repository for assemblies consists of a web service with the following methods:

- eRepAddDocument: upload a document into the repository.

Each method of the web service is described with an IDL-signature and a table describing the purpose of the parameters.

II.1.4.a. eRepAddDocument

One logical document may consist of multiple files. For example each chapter of a document may be stored in a separate .pdf or .word file, or maps and pictures for a document may be stored in separate files. Therefore a list of files can be specified when adding a document to

the repository. Each file can have its own file format and description. Only one of the files in the list can be designated as the main file of the document.

The purpose of this method is to submit a document with all its individual files to the document repository.

```
void eRepAddDocument (  
    in string assemblyId,  
    in string collection,  
    in string title,  
    in docIdentifier identifier optional,  
    in string language,  
    in string documentType,  
    in string abstract,  
    in stringList authors,  
    in stringList eurovocCodes optional,  
    in dcTagList dublinCoreTags optional,  
    in uploadFileList files,  
    out string documentUrl optional,  
    out short statusCode,  
    out string statusMessage optional);
```

```
struct docIdentifier {  
    string identifierType,  
    string value  
};
```

```
struct dcTag {  
    string tagName,  
    string value  
};
```

```
typedef sequence <dcTag> dcTagList;
```

```
struct uploadFile {  
    string fileName,  
    boolean main,  
    string description optional,  
    base64 signature optional,  
    string format,  
    base64 content  
};
```



typedef sequence <uploadFile> uploadFileList;

Parameter	Direction	Description
assemblyId	in	Identifies the assembly. Will be used upon logging of method calls.
collection	in	The dspace collection to store the document in.
title	in	The title of the document.
identifier	in	The optional document identifier as a doclidentifier structure (e.g. identifierType = ISBN with value = xxxxxx)
language	in	The language of the document
documentType	in	The type of the document (e.g. Technical Report, Book, etc.)
abstract	in	The abstract of the document.
authors	in	List of author names that contributed to the document.
eurovocCodes	in	Optional list of Eurovoc codes applicable to the document. The codes of any ancestor tags for these listed tags will be assigned to the document by the web service.
dublinCoreTags	in	Optional list of dcTag structures with additional Dublin Core tags applicable to the document.
files	in	List of uploadFile structures with details and content of each individual file: <ul style="list-style-type: none"> ○ fileName: the name of the file ○ main: indicates if it is the main file of the document (true for just one file in the list) ○ description: optional description of the file ○ format: format of the file (Microsoft Word, etc.) ○ signature: optional digital signature of file ○ content: base64 encoded content of the file
documentUrl	in	The document URL of the added document in case of success.
statusCode	in	Code indicating success or failure of the method call.

statusMessage

in

Error text in case of failure.

Because the content of a file will most of the time be in a binary format the content is converted to text using base64 encoding so it can be used as a string value in the SOAP XML messages.

Recognized collection values:

- eRepresentative

Recognized document identifier names:

- ISSN
- Other
- ISMN
- Gov't Doc #
- URI
- ISBN

Recognized document type values:

- Animation
- Article
- Book
- Book chapter
- Dataset
- Learning Object
- Image
- Image, 3-D
- Map
- Musical Score
- Plan or blueprint
- Preprint
- Presentation
- Recording, acoustical
- Recording, musical
- Recording, oral
- Software
- Technical Report
- Thesis
- Video
- Working Paper
- Other

Recognized file format values:

- Adobe Portable Document Format
- Audio Interchange File Format
- Basic Audio
- Broadcast Wave Format
- Cascading Style Sheets
- Extensible Markup Language
- Filemaker Pro
- Graphics Interchange Format
- Hypertext Markup Language

- Item-specific license agreed upon to submission
- Joint Photographic Experts Group/JPEG File Interchange Format (JFIF)
- Kodak Photo CD image
- LaTeX document
- Machine-Readable Cataloging records
- Mathematica Notebook
- Microsoft Excel
- Microsoft Powerpoint
- Microsoft Project
- Microsoft Visio
- Microsoft Windows bitmap
- Microsoft Word
- Moving Picture Experts Group
- MPEG Audio
- Photoshop
- Plain Text
- Portable Network Graphics
- Postscript Files
- RealAudio file
- Rich Text Format
- SGML application (RFC 1874)
- Tag Image File Format
- TeX dvi format
- Tex/LateX document
- Unknown data format
- Video Quicktime

Recognized Dublin Core tag names (elements with optional qualifiers):

- dc.contributor.advisor
- dc.contributor.author
- dc.contributor.editor
- dc.contributor.illustrator
- dc.contributor.other
- dc.contributor
- dc.coverage.spatial
- dc.coverage.temporal
- dc.date.accessioned
- dc.date.available
- dc.date.copyright
- dc.date.created
- dc.date.issued
- dc.date.submitted
- dc.date
- dc.description.abstract
- dc.description.provenance
- dc.description.sponsorship
- dc.description.statementofresponsibility
- dc.description.tableofcontents
- dc.description.uri
- dc.description
- dc.format.extent
- dc.format.medium
- dc.format.mimetype

- dc.format
- dc.identifier.citation
- dc.identifier.govdoc
- dc.identifier.isbn
- dc.identifier.ismn
- dc.identifier.issn
- dc.identifier.other
- dc.identifier.sici
- dc.identifier.uri
- dc.identifier
- dc.language.iso
- dc.language
- dc.publisher
- dc.relation.haspart
- dc.relation.hasversion
- dc.relation.isbasedon
- dc.relation.isformatof
- dc.relation.ispartof
- dc.relation.ispartofseries
- dc.relation.isreferencedby
- dc.relation.isreplacedby
- dc.relation.isversionof
- dc.relation.replaces
- dc.relation.requires
- dc.relation.uri
- dc.relation
- dc.rights.uri
- dc.rights
- dc.subject.classification
- dc.subject.ddc
- dc.subject.lcc
- dc.subject.lcsh
- dc.subject.mesh
- dc.subject.other
- dc.subject
- dc.title.alternative
- dc.title
- dc.type

Returned statusCode values (the statusMessage values are not predefined):

- 1 = Successful
- 100 = General error in request
- 101 = Schema validation error in request
- 102 = Base64 decoding error in request
- 103 = Eurovoc code validation error in request
- 300 = General internal error on server
- 301 = Database error on server
- 302 = DSpace error on server

Stub webservice implementation

A stub webservice implementation is available for testing purposes. This stub implementation does not yet store documents into DSpace but it validates a received request against the XML schema definition from the WSDL file and uploaded files are base64 decoded and stored in the directory: /home/eRep/temp/upload.

For using a logical server name instead of an IP address in the webservice URL add a definition for it in the hosts file.

On Microsoft Windows this file can be found in:

`%Systemroot%\System32\Drivers\Etc`

On Linux this file can be found in:

`/etc`

Add a line like this as an alternative for the used IP address:

`192.168.93.128 eRepRepository`

To start the server, login at the eRepRepository Redhat server

Then open a terminal console and start the Tomcat server with the following command:

`./start_erep.sh`

(The Tomcat server can be stopped with: `./shutdown_erep.sh`)

When the Tomcat server has been started, the WSDL file of the webservice can be obtained with the following URL:

`https://ereprepository:8443/erep/services/RepositoryAssembly?wsdl`

The webservice can be invoked with the following URL:

`https://ereprepository:8443/erep/services/RepositoryAssembly`

These URLs access the SSL port of the Tomcat server. The webservice is secured by means of BASIC

For production usage the required heap size for the Tomcat server must be set to at least 500 MB. The stub webservice will be deployed in a VMWare virtual machine where such an amount of heap space is not available. Therefore the stub webservice will not be able to handle documents as large as the production service will be able to handle.

II.1.4.b. PHP5 webservice client example

This paragraph contains example code for a PHP5 page that acts as a SOAP client for the webservice. This example adds one document to the repository. The added document

consists of two files: one created from fixed text and one Microsoft word document read from disk.

```
<html>
<body>
<h3>RepositoryAssembly soapclient example</h3>
<?php
$client = new soapclient(
'https://erepository:8443/erep/services/RepositoryAssembly?wsdl',
array('login' => ' ', 'password' => ' '));
$content1 = base64_encode('Some dummy text');
$uploadFile1 = array(
'fileName' => 'dummy.txt',
'main' => 'true',
'format' => 'Plain Text',
'content' => $content1
);
$wordFile = "/home/eRep/temp/test.doc";
$fh = fopen($wordFile, "r");
$wordData = fread($fh, filesize($wordFile));
fclose($fh);
$content2 = base64_encode($wordData);
$uploadFile2 = array(
'fileName' => 'test.doc',
'main' => 'false',
'format' => 'Microsoft Word',
'content' => $content2
);
$files = array(
$uploadFile1,
$uploadFile2
);
$authors = array('HP employee');
$params = array(
'assemblyId' => 'testId',
'collection' => 'eRepresentative',
'title' => 'test title',
```

```
'language' => 'nl',
'documentType' => 'Other',
'abstract' => 'summary text',
'authors' => $authors,
'files' => $files
);
$response = $client->eRepAddDocument($param);
echo print_r($response);
echo '<br>';
echo 'Return code = ', $response->statusCode;
?>
</body>
</html>
```

II.1.5. DSPACE

The version of DSPACE proved to be fully compatible with Demos@Work is an older version than the current one of this server so it's highly recommended to the administrator to use the provided version rather than the latest one.

For complete installation instructions please see documentation at the project homepage detailed in the table at the beginning of this document.

II.2. Installation and configuration of DemosCitizen

II.2.1. Portal:

To install the portal please copy the portal files to the defined Apache virtual directory (by default is /var/www/). Concerning its configuration two files are used:

- configuration.php
- forum/Settings.php

These files must be present in Apache virtual directory. Their content variables are defined as follows:

- Configuration.php: this file can be found at the root of the portal and the following variables must be changed according to the server specifications:
 - \$mosConfig_absolute_path – absolute path to the portal root
 - \$mosConfig_db – the name of the database that the portal is stored in
 - \$mosConfig_host – the database host
 - \$mosConfig_live_site – the URL of the live site
 - \$mosConfig_locale – the locale has to be changed according to the language of the portal (e.g. ca_CA for Catalan)
 - \$mosConfig_mailfrom – the email address from which emails will be sent
 - \$mosConfig_sitename – the name of the site (default: Demos@Work)
 - \$mosConfig_smtphost – The smtp host for sending emails
 - \$mosConfig_smtpuser – The username for the smtp host
 - \$mosConfig_smtppass – The password for the smtp host
 - \$mosConfig_user - The database username
 - \$erepDBHost – host where the MP part of Demos@Work resides (eRepresentative)
 - \$erepDBName – name of the DemosAtWorkMP database
 - \$erepDBPassword – password for the DemosAtWorkMP database
 - \$erepDBUsername – username for the DemosAtWorkMP database
 - \$erepLiveSite – URL of the DemosAtWorkMP live site.
- Settings.php: this file contains properties for the forum:
 - \$mbname – Name for the forum (default: Forum)
 - \$language – forum language (catalan for Catalan and so on)
 - \$boardurl - the URL under which the forum will be situated
 - \$webmaster_email - the webmaster's email
 - \$db_server – the host of the database

- \$db_name – database name
- \$db_user – the database username
- \$db_passwd – the database password
- \$boarddir - the absolute path to the forum
- \$sourcedir – the absolute path to the forum source files

Additionally, the following settings can be configured either altering the tables on database or from the forum administration as system administrator

- smf_settings:
 - smileys_url – URL to the live portal smileys folder
 - avatar_url – URL to the live portal avatar folder
 - attachmentUploadDir – absolute path to the attachment uploading folder
 - smileys_dir – absolute path to the smileys folder
 - avatar_directory – absolute path to the avatar folder
- smf_themes:
 - theme_url – URL to live portal theme folder
 - images_url – URL to live portal images folder
 - theme_dir – absolute path to the portal theme folder
- jos_jsmf_config: in the 1st row of the table, the config value must be changed to the absolute path of the forum

II.2.1.a. Language Configuration

The DemosAtWork uses different language files for its components. These language files are located in the “language” in each component’s folder. For example:

/language – main portal language file
/components/com_extcalendar/language
/components/com_letters/language
/components/com_processes/language
/administrator/components/com_docman/language
/forum/Themes/default/languages...

In order for a language to function properly, a php file with the language as name has to be created. You can also duplicate the English file and rename it to catalan.php for Catalan. If a language file does not exist in one of the language folders, then the system will use the english.php file by default.

II.2.1.b. Language Selection



From the Joomla control panel it is also possible to change the language used by the portal pages and the Eurovoc descriptions. Select the Language Manager from the control panel, then choose a language from the list and click Publish.

For further information on Joomla! configuration, please see Joomla! documentation at the project homepage: <http://docs.joomla.org/>

II.2.2. Semantics Module_INITIALIZER & Semantics Module

Both java applications must be placed in the Glassfish's auto-deploy folder. First the initializer should be run to create the DB and then the other one. An .owl file is required.

The external configuration of the application must reside in a properties file which must be reached through the classpath (thus the containing directory should be added to the launch configuration of the JVM).

Here is a description of the different semantics module properties:

Property	Description
ontology.file	Absolute file path of the ontology .owl file.
ontology.db.url	URL for the jena framework database connection.
ontology.db.user	Username for the jena framework database connection.
ontology.db.passwd	Password for the jena framework database connection.
ontology.dbdriver.class	JDBC driver class for the jena framework database connection (for mysql it's com.mysql.jdbc.Driver).
ontology.db	Database type key for the jena framework database connection (for MySQL it's 'MySQL').
ontology.model.name	Unique ontology key for the jena framework database connection. For instance 'Ontology200610'.
hibernate.english.datasource	URL pointing to the schema for english language ontology data.
hibernate.italian.datasource	URL pointing to the schema for italian language ontology data.
hibernate.polish.datasource	URL pointing to the schema for polish language ontology data.
hibernate.french.datasource	URL pointing to the schema for french language ontology data.

	ontology data.
hibernate.dialect	Hibernate connection dialect (for mySQL use org.hibernate.dialect.MySQLDialect)
hibernate.connection.driver_class	Hibernate connection driver class (for mySQL use com.mysql.jdbc.Driver)
hibernate.connection.username	Hibernate connection user name.
hibernate.connection.password	Hibernate connection password.
hibernate.connection.url	
hibernate.connection.charSet	Set to UTF-8
hibernate.connection.useUnicode	Set to 'yes'
hibernate.connection.characterEncoding	Set to 'utf8'
hibernate.connection.connectionCollation	Set to 'utf8_polish_ci'
hibernate.hibernate.hbm2ddl.auto	Set to 'update'

An example of configuration file would be:

```
# Database connection config.
ontology.db.url=jdbc:mysql://localhost/schemaname
ontology.db.user=root
ontology.db.passwd=
ontology.dbdriver.class=com.mysql.jdbc.Driver
ontology.db=MySQL
ontology.model.name=Ontology200610
ontology.supported.languages=cat
hibernate.cat.datasource=jdbc:mysql://localhost:3306/kdbcat
hibernate.dialect=org.hibernate.dialect.MySQLDialect
hibernate.show_sql=true
hibernate.connection.charSet=UTF-8
hibernate.connection.useUnicode=yes
hibernate.connection.characterEncoding=utf8
hibernate.hibernate.hbm2ddl.auto=update
```

To run initialization process with a defined configuration file, please use the following command:

```
java -jar semanticModule.jar -classpath=[classpath values]
```

