his document is part of the Network of Excellence “Multilingual Europe Technology Alliance (META-NET)”,
co- funded by the 7th Framework Programme of the European Commission through the T4ME grant agreement no.: 249119.

A Network of Excellence forging the

Multilingual Europe Technology Alliance

META-SHARE V2.1 Provider Manual

Author(s): Victoria Arranz, Christian Federmann, Ioanna Giannopoulou, Olivier Hamon, Marc Schröder

Date: May 18th, 2012

Table of Contents

1. Executive Summary

This document is a guide for the users of META-SHARE interested in providing Language Resources (LRs) through META-SHARE.

2. Provider-oriented documentation

2.1. Upgrading a user account to a provider account

As per V2.1 of META-SHARE, only user accounts marked as “staff” users have the permission to create or edit LRs. For any given META-SHARE node, each “staff” user has the right to edit all LRs on that node. This means that only trusted individuals should be granted staff status.

Future versions of META-SHARE will provide a more fine-grained system of access rights.

In order to endow a user account with provider access, the following steps are necessary.

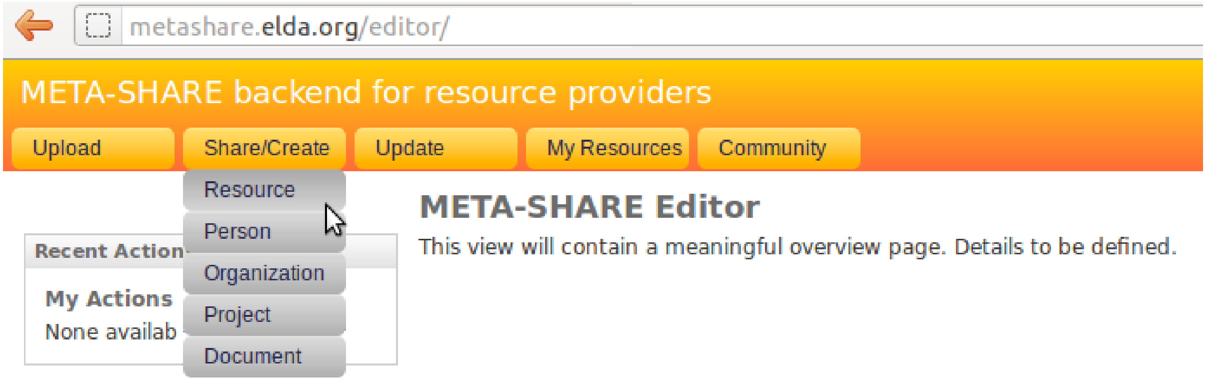
1. The user account should be created as documented in the general User Manual.
2. An administrative user must upgrade this account to a “staff” account. As described in the Installation Manual, at least one Administrator account is created as part of the installation process. This administrator can log into the administrative interface at <metashare-url>/admin. Under “Users”, go to the “Details” page for the user in question, tick the permissions box “Staff status”, and confirm with “Save”.
3. The staff user should now be able to log in to the META-SHARE portal as described in the User Manual and have the permissions required to add and edit LRs as described below.

2.2. Add a Language Resource

The provider can add a new LR by inserting the *metadata* that describes this LR. The provider must be registered and logged in so as to add new LRs, and then do as follows:

1. Click on the “Editor” button on the top right of the main page of META-SHARE.

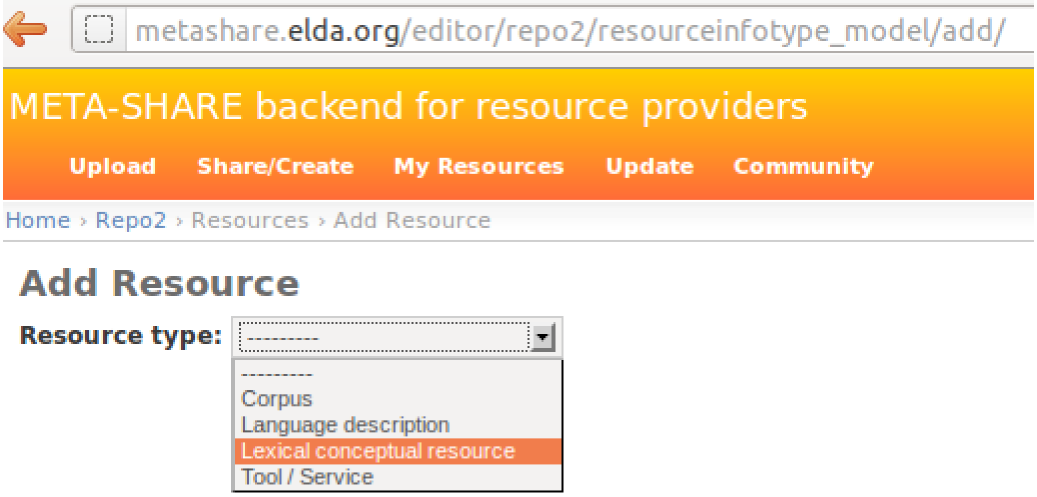
2. In the result page ("META-SHARE backend for resource providers"), click on the "Share/Create" link at the top left (see Figure 1). From the drop-down menu that appears, click on "Resource". The result page presents a drop-down menu, listing all the available *resource types* (see Figure 2).



META-SHARE Editor

This view will contain a meaningful overview page. Details to be defined.

Figure 1: META-SHARE backend for resource providers


metashare.elda.org/editor/repo2/resourceinfotype_model/add/
META-SHARE backend for resource providers
Upload
Share/Create
My Resources
Update
Community
Home › Repo2 › Resources › Add Resource

Corpus
Language description
Lexical conceptual resource
Tool / Service

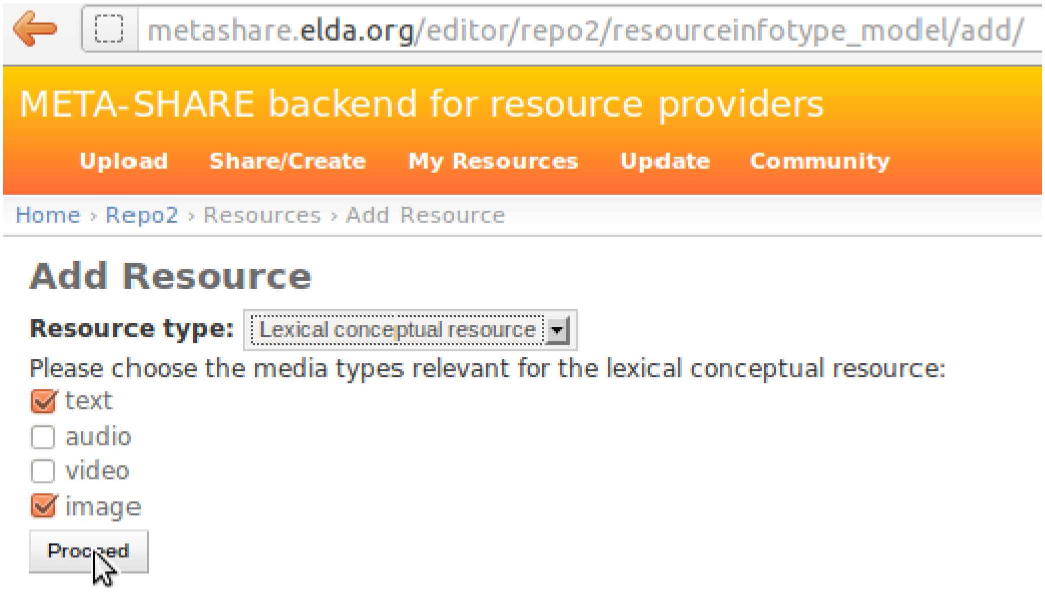
Add Resource

Resource type: ---------

Figure 2: Choose one of the available resource types

3. Choose a resource type for the LR. (A LR can only have one resource type). If your choice is "tool / service", go directly to step 4.
   If your choice is of the other three resource types, select from the appearing check-box list, the *media type(s)* for the LR (see figure 3).
   Note: In the case of "lexical conceptual resource" and "language description" resource types, "text" is a required media type and, as such, it is selected by default and cannot be unselected.
4. Click on the "proceed" button.

In the result page (see Figures 4 & 5), all the metadata information to be filled in, is summarized under a left hand side menu (see section 2.4 for the description of the metadata fields). On the top of this menu, there is the administrative information categorized as required, recommended and optional, if any. A click on any of these options, will lead in expanding the appropriate metadata block on the right hand side.



metashare.elda.org/editor/repo2/resourceinfotype_model/add/

META-SHARE backend for resource providers

Home › Repo2 › Resources › Add Resource

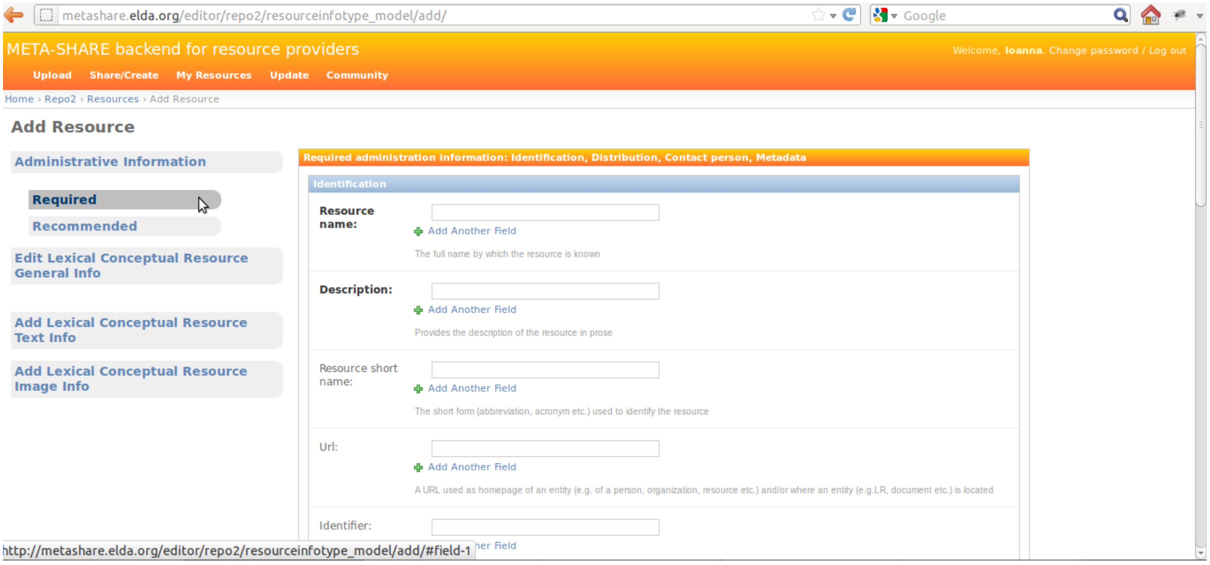
Add Resource

Resource type: Lexical conceptual resource

Please choose the media types relevant for the lexical conceptual resource:

☑ text
☐ audio
☐ video
☑ image

Figure 3: Choose media types for the selected resource type



metashare.elda.org/editor/repo2/resourceinfotype_model/add/

META-SHARE backend for resource providers

Welcome, **Ioanna**. Change password / Log out

Upload
Share/Create
My Resources
Update
Community

Home › Repo2 › Resources › Add Resource

Add Resource

Administrative Information
Required
Recommended
Edit Lexical Conceptual Resource General Info
Add Lexical Conceptual Resource Text Info
Add Lexical Conceptual Resource Image Info

Required administration information: Identification, Distribution, Contact person, Metadata

Identification

Resource name:
Add Another Field
The full name by which the resource is known

Description:
Add Another Field
Provides the description of the resource in prose

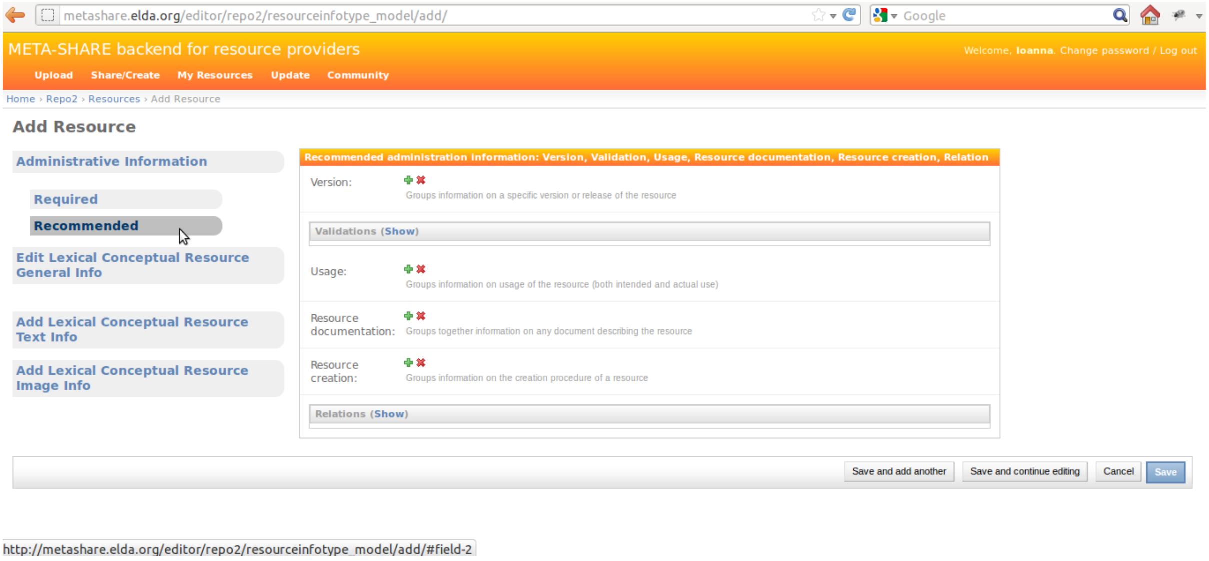
Resource short name:
Add Another Field
The short form (abbreviation, acronym etc.) used to identify the resource

Url:
Add Another Field
A URL used as homepage of an entity (e.g. of a person, organization, resource etc.) and/or where an entity (e.g.LR, document etc.) is located

Identifier:
her Field

http://metashare.elda.org/editor/repo2/resourceinfotype_model/add/#field-1

Figure 4: Administrative Information metadata block (Required)



metashare.elda.org/editor/repo2/resourceinfotype_model/add/

META-SHARE backend for resource providers

Welcome, **Ioanna**. Change password / Log out

Upload · Share/Create · My Resources · Update · Community

Home › Repo2 › Resources › Add Resource

Add Resource

Administrative Information
- Required
- Recommended

Edit Lexical Conceptual Resource General Info

Add Lexical Conceptual Resource Text Info

Add Lexical Conceptual Resource Image Info

Recommended administration information: Version, Validation, Usage, Resource documentation, Resource creation, Relation

Version: Groups information on a specific version or release of the resource

Validations (Show)

Usage: Groups information on usage of the resource (both intended and actual use)

Resource documentation: Groups together information on any document describing the resource

Resource creation: Groups information on the creation procedure of a resource

Relations (Show)

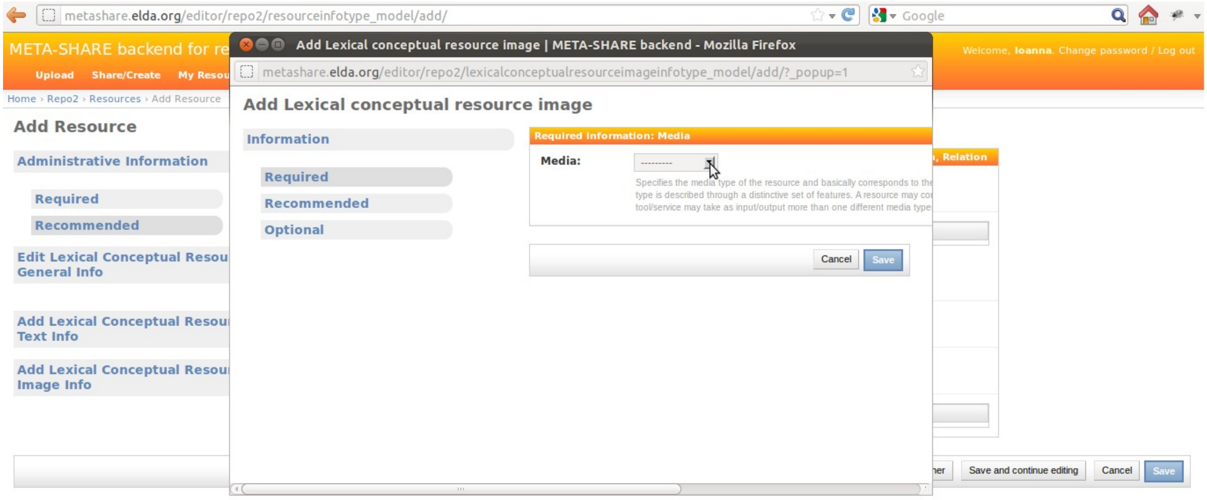
Save and add another · Save and continue editing · Cancel · Save

http://metashare.elda.org/editor/repo2/resourceinfotype_model/add/#field-2

Figure 5: Administrative Information metadata block (Recommended)

5. Fill in the administrative metadata fields (Only the required information -appearing as bold- is necessary for a LR to be successfully saved).

On the same level of the left hand side menu as the administrative information, there are the resource-type-specific information. Clicking on one of these, will lead in opening a pop-up window, containing the blocks of metadata to be filled in (see Figure 6 for an example). These pop-up windows follow a similar menu structure (required / recommended / optional) as the main page.



Add Lexical conceptual resource image | META-SHARE backend - Mozilla Firefox

metashare.elda.org/editor/repo2/lexicalconceptualresourceimageinfotype_model/add/?_popup=1

Add Lexical conceptual resource image

Information
- Required
- Recommended
- Optional

Required information: Media

Media: ---------

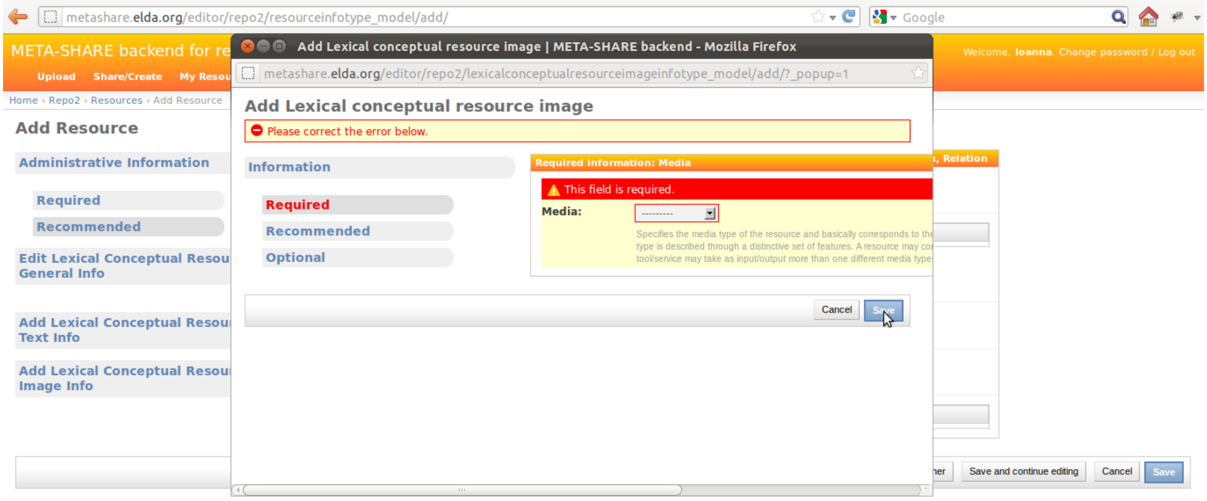
Specifies the media type of the resource and basically corresponds to the
type is described through a distinctive set of features. A resource may co
tool/service may take as input/output more than one different media type

Cancel · Save

Figure 6: Resource-type-specific pop-up window (here for a Lexical-Conceptual Resource)

6. For each one of the pop-up windows, fill in the metadata fields (again, only the required information is necessary for a LR to be successfully saved). Once done with filling in information within a pop-up window, click on the

"Save" button at the bottom of the page. In case of errors, a message will be displayed on the top of the page, the errors will be reflected on the menu options and the problematic fields (i.e. fields which either contain an error or are empty) will be highlighted in red (see Figure 7 for an example). If there are no errors, the "Save" button will close the pop-up window, returning to the previous page.


META-SHARE backend for re
Upload
Share/Create
My Resou
Home › Repo2 › Resources › Add Resource
Add Resource
Administrative Information
Required
Recommended
Edit Lexical Conceptual Resou General Info
Add Lexical Conceptual Resou Text Info
Add Lexical Conceptual Resou Image Info
Welcome, Ioanna. Change password / Log out
, Relation
Information
Required
Recommended
Optional
Cancel
Save
Save and continue editing

metashare.elda.org/editor/repo2/resourceinfotype_model/add/

Add Lexical conceptual resource image | META-SHARE backend - Mozilla Firefox

metashare.elda.org/editor/repo2/lexicalconceptualresourceimageinfotype_model/add/?_popup=1

Add Lexical conceptual resource image

Please correct the error below.

Required information: Media

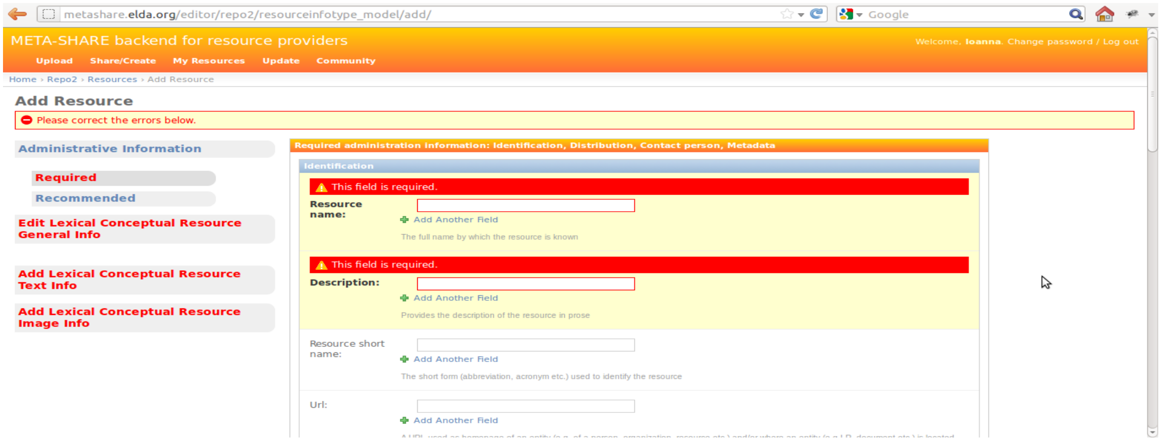
This field is required.

Media: ---------

Specifies the media type of the resource and basically corresponds to the type is described through a distinctive set of features. A resource may co tool/service may take as input/output more than one different media type

Figure 7: Errors in pop-up window: the media field has not been filled-in

7. Once done with filling in information both in the main page and in all the pop-up windows, click on the "Save" button at the bottom of the main page. If there are errors, they will be indicated in a similar way as in the pop-up windows (see Figure 8 for an example). If there are no errors, clicking on "Save" will result in a new page, displaying the message "The Resource 'resource name' was added successfully", and listing all the LRs added in META-SHARE, included the newly created one (see Figure 9).


META-SHARE backend for resource providers
Welcome, Ioanna. Change password / Log out
Upload
Share/Create
My Resources
Update
Community
Home › Repo2 › Resources › Add Resource
Administrative Information
Required
Recommended
Edit Lexical Conceptual Resource General Info
Add Lexical Conceptual Resource Text Info
Add Lexical Conceptual Resource Image Info
Add Another Field

metashare.elda.org/editor/repo2/resourceinfotype_model/add/

Add Resource

Please correct the errors below.

Required administration information: Identification, Distribution, Contact person, Metadata

Identification

This field is required.

Resource name:

The full name by which the resource is known

This field is required.

Description:

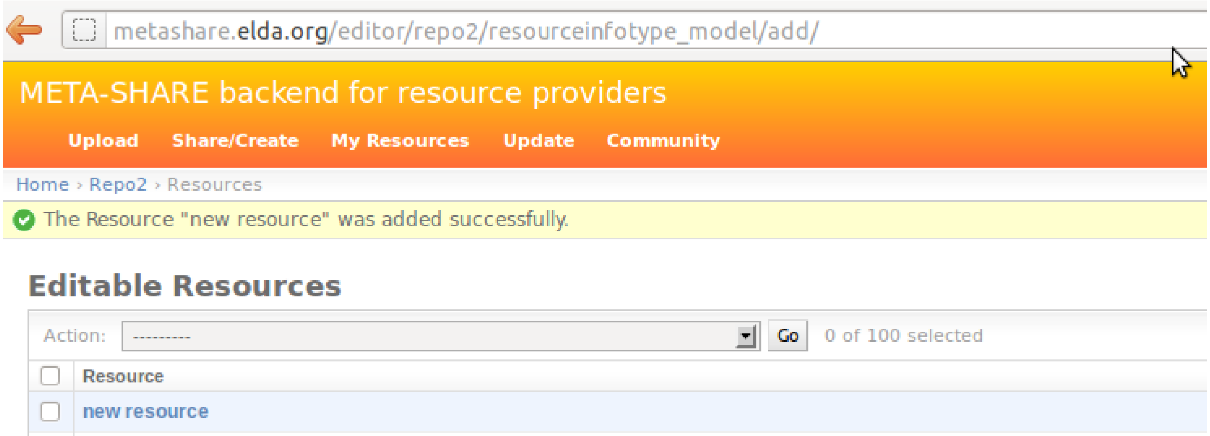
Provides the description of the resource in prose

Resource short name:

The short form (abbreviation, acronym etc.) used to identify the resource

Url:

Figure 8: Errors in main page: the resource name and the description have not been filled-in, errors also in General Info and Text Info tabs



metashare.elda.org/editor/repo2/resourceinfotype_model/add/

META-SHARE backend for resource providers

The Resource "new resource" was added successfully.

Editable Resources

Action: --------- Go 0 of 100 selected

□	Resource
□	new resource

Figure 9: Successfully saved LR

2.2.1. Add other metadata entities

As shown in figure 1, the provider has a choice of creating other entities as well. The point of creating these types of metadata (namely Person, Organization, Project, Document) apart / independently from a resource, is that they can be reused in many LR descriptions. So, if for example a provider creates a project description first, then all he has to do when creating a LR description, is to select the project from the list of existing projects, instead of creating it from scratch.

2.3. Manage Language Resources

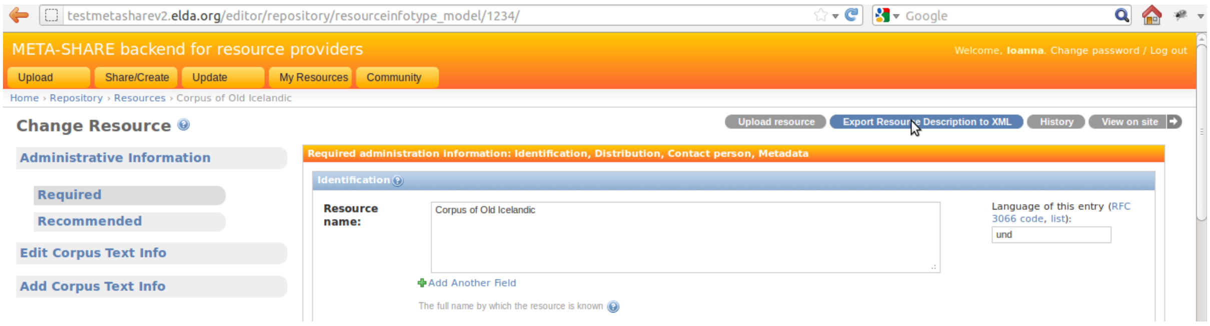
The provider is a “staff” member, which means, as mentioned above, that he can edit all the LRs within the specific METASHARE node. On the top left of the “META-SHARE backend for resource providers” page, the provider can click on the “Update” link, where he can view a list of all the LRs / People / Organizations / Projects / Documents for the specific META-SHARE node. He can also click on the “My Resources” link, where he can only view a list of the LRs he has provided on META-SHARE himself.

The provider can view, edit, delete, *publish* / *unpublish* / *ingest* all the listed LRs or export to XML the LRs' descriptions. He can select the LR(s) he wishes to apply an action to and then select an action from the drop-down menu of actions (delete / export / publish / unpublish / ingest).
Clicking on “Go” button will execute the selected action.

If the provider wishes to edit a LR, he can click on its name. This will result in a page where he can edit all the metadata fields for this resource, which will appear in exactly the same view as described in section 2.2.

2.3.1. Export LR descriptions to XML

The provider is able to export the description of a LR to META-SHARE XML format. One way of performing the exporting is through the Update page, for one or multiple LRs, as mentioned in the previous section. Another way of exporting is through the metadata editing page, by clicking on the relevant button on the top right of the page (see figure 10).


testmetasharev2.elda.org/editor/repository/resourceinfotype_model/1234/
Google
META-SHARE backend for resource providers
Welcome, Ioanna. Change password / Log out
Upload
Share/Create
Update
My Resources
Community
Home › Repository › Resources › Corpus of Old Icelandic
Change Resource
Upload resource
Export Resource Description to XML
History
View on site
Administrative Information
Required
Recommended
Edit Corpus Text Info
Add Corpus Text Info
Identification
Add Another Field

Required administration information: Identification, Distribution, Contact person, Metadata

Resource name: Corpus of Old Icelandic

Language of this entry (RFC 3066 code, list): und

The full name by which the resource is known

Figure 10: Exporting a LR description to XML

2.4. Add and edit metadata information

Metadata information for LRs are organised in two levels, the first being Administrative / Type-specific, and the second being required / recommended / optional.

The tabbed menu described in section 2.2 is making the two levels clear to the provider (see Figures 2, 3 & 4). A description of the metadata fields follows:

1. Administrative
   i. Required: Contains information on identification, distribution, contact person, metadata of the resource, etc.
   ii. Recommended: Contains information on version/release, validation, usage, documentation, creation, relations of the resource, etc.

➔ *Resource type: Corpus*

2. Corpus Text Info*: Information on the text component of the resource, such as the language, the text format, the encoding, the annotation, etc.

3. Corpus Audio Info*: Information on the audio component of the resource, such as the language, the audio format, the size, the recording, the capture, etc.

4. Corpus Video Info*: Information on the video component of the resource, such as the size, the video format, the content, the language, etc.

5. Corpus Image Info*: Information on the image component of the resource, such as the size, the image format, the content, etc.

6. Corpus Text Numerical Info*: Information on the numerical text component of the resource, such as the size, the content, the creation, etc.

7. Corpus Text n-gram Info*: Information on the text n-gram component of the resource, such as the base item (word, syllable, letter, etc.), the language, the size, the text format, the encoding, etc.

➔ *Resource type: Language Description*

2. Language Description General Info*: Information on the specific type of the resource (grammar, etc.), the encoding, the performance, etc.

3. Language Description Text Info*: Information on the text component of the resource, such as the language, the linguality (monolingual, bilingual, etc.), the text format, the encoding, the size, etc.

4. Language Description Video Info*: Information on the video component of the resource, such as the creation, the size, the content, the video format, the language, etc.

5. Language Description Image Info*: Information on the image component of the resource, such as the creation, the size, the content, the image format, etc.

➔ *Resource type: Lexical Conceptual Resource*

2. Lexical Conceptual Resource General Info*: Information on the specific type of the resource (thesaurus, terminology, etc.), the encoding, as well as further details about the creation of the resource.

3. Lexical Conceptual Resource Text Info*: Information on the text component of the resource, such as the languages, the linguality (monolingual, bilingual, etc.), the text format, the encoding, the size, etc.

4. Lexical Conceptual Resource Audio Info*: Information on the audio component of the resource, such as the size, the audio format, the content, the language, etc.

5. Lexical Conceptual Resource Video Info*: Information on the video component of the resource, such as the video content, the size, the video format, the language, etc.

6. Lexical Conceptual Resource Image Info*: Information on the image component of the resource, such as the image format, the size, the content, etc.

➔ *Resource type: Tool / Service*

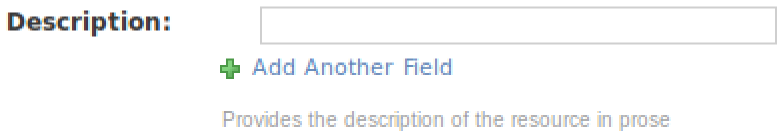
2. Tool / Service Info*: Information on the specific type of the resource (tool, service, platform, infrastructure, etc.), details about the input and output types, information on the operating systems and running environments, etc.

*All this information is divided in blocks: required / recommended / optional.

2.5. Input types

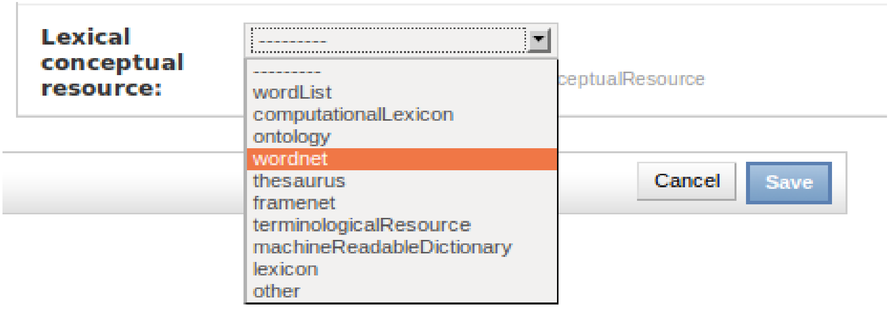
In order to insert or modify metadata, the provider has access to different kinds of input types:

- Text field: the provider may add/modify free text that corresponds to the description of the metadata elements;

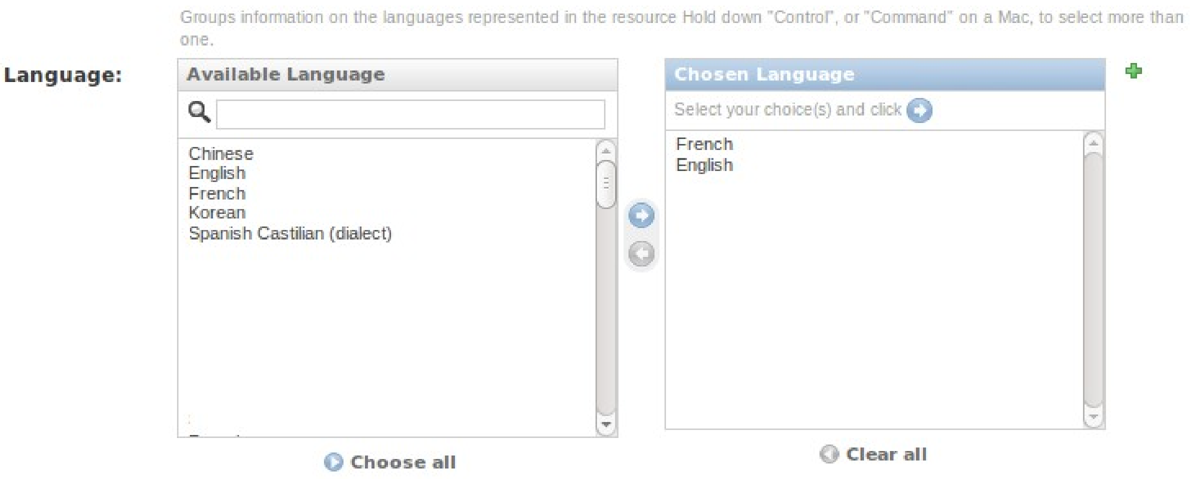


Provides the description of the resource in prose

- Drop-down list box: the provider may choose one item from the list provided;



- Multiple-choice Drop-down list box: the provider may chose one or several items from the list provided;

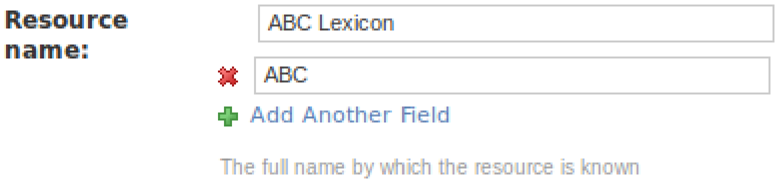


Groups information on the languages represented in the resource Hold down "Control", or "Command" on a Mac, to select more than one.

Language:

Available Language	Chosen Language
Chinese English French Korean Spanish Castilian (dialect)	French English

- Repeatable text field: the provider may add or delete all the text field he wants to by clicking on the “+” or “x“ buttons, respectively.

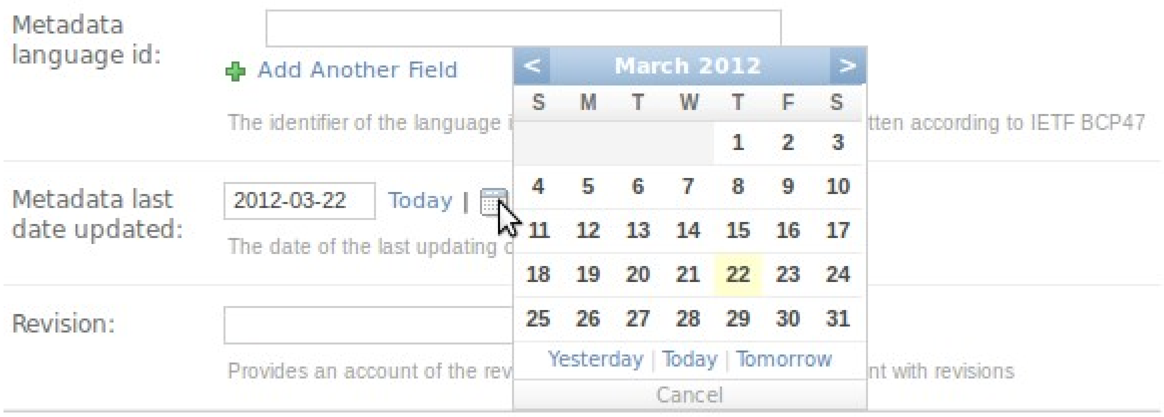


Resource name: ABC Lexicon

✖ ABC

The full name by which the resource is known

- Calendar text field: the provider may write a full date on his own or select a date from a pop-up calendar.



Metadata language id:

The identifier of the language i… …tten according to IETF BCP47

Metadata last date updated: 2012-03-22

The date of the last updating c…

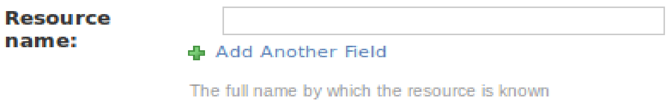
Revision:

Provides an account of the rev… …nt with revisions

< March 2012 >

S	M	T	W	T	F	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

All fields are accompanied by a short description/explanation line, as shown below:



Resource name:

The full name by which the resource is known

2.6. Import Language Resources

2.6.1. Prepare the import file

To be able to import new LRs, the provider must create XML *metadata files* for each LR, that follow the META-SHARE metadata. This compliance can be validated using the META-NET *XML schema (XSD)*, the current version being under the `misc/schema/v2.0` folder.

2.6.2. Import script

Importing XML metadata is also performed from the command line. It describes the import of existing LRs, but the process is the same for LRs converted from different representations as described below.

Importing of resources is possible using `import_xml.py` which can be found within the `metashare` installation folder.

Basic usage information is given when called without any arguments:

```
$ python import_xml.py
usage: import_xml.py <file.xml|archive.zip> \
         [<file.xml|archive.zip> ...]
```

The script reports errors and gives import statistics when finished.

A single xml file can be imported as follows:

```
$ python import_xml.py my_LR.xml
Done.  Successfully imported 1 files into the database, \
   errors occurred in 0 cases.
```

Multiple LRs can be imported using the “*” wildcard or by packaging the xml file into a zip file:

```
$ python import_xml.py metashare-v1-resources.zip
Done.  Successfully imported 1,277 files into the database, \
   errors occurred in 0 cases.
```

Importing a large zip file, or a large amount of XML files, takes some time; this is normal.

After successful import of these files, the provider should have the new resources available on his META-SHARE node.

The last line of the `import_xml.py` output should report no errors.

2.6.3. Upload in META-SHARE

A provider can upload an XML metadata file by clicking on the “Upload” link at the top left of the “META-SHARE backend for resource providers” page. Uploading works as follows:

1. Choose a file by clicking on the “Browse button”;
2. Click on “Upload Terms” check-box to accept them;
3. Click on the “Upload” button.

2.6.4. Conversion of existing LRs

The conversion of the existing LRs from several different XML schemata to META-NET XML schema has been made with XSL transformations.

The process was the following:

- The first step was to map all the elements of the other XML schema to the elements of META-NET XML schema, one by one.
- Then, a XSL file was created, which contains the structure of META-NET XML schema, and XSLT lines harvesting the information from the files of the other XML schema format.
- The conversion process was executed for all given files.

Example:

Element *resource_periodofcoverage* from ELRA format has been mapped to *timeCoverage* element in META-SHARE schema format.

Line in ELRA XML file:

```
<resource_periodofcoverage>Between 1992 and 1999</resource_periodofcoverage>
```

XSL conversion file:

```
<xsl:variable name='time' select='preceding-sibling::resource_periodofcoverage'/>

        <xsl:if test='$time!=""'>
                <timeCoverageInfo>
                        <timeCoverage>
                                <xsl:value-of select='$time'/>
                        </timeCoverage>
                </timeCoverageInfo>
        </xsl:if>
```

Lines in final META-SHARE XML file:

```
	<timeCoverageInfo>
		<timeCoverage>Between 1992 and 1999</timeCoverage>
	</timeCoverageInfo>
```

In the example above, the element "*timeCoverage*" is part of the meta-share XML structure. The content inside the element is added by the line *<xsl:value-of select='$time'/>*, where *$time* is a XSL variable, defined before, as the (xpath) "*preceding-sibling*" with the name "*resource_periodofcoverage*". If the variable is empty, the element does not appear in the final XSL file.

This is the way to construct, little by little, a META-SHARE compliant XML file.

For the conversion, there are many things that need to be taken into account, such as mandatory fields, unique fields, repeated fields, elements that have the enumeration constraint (i.e. that can only contain a value from a set of acceptable values), etc. Thus, the conversion files may contain far more complicated transformations than the one displayed before.

Due the complexity of the META-SHARE schema and the significant differentiation between this and the other formats, some tools and mechanisms were used to simplify the conversion process.

In the case of ILSP and PSP resources conversion, the tool Altova-MapForce was used for the mapping between schemata and the production of the XSL files. Yet, there were quite a few cases where the automatically produced files had to be manually edited in order to fully comply with META-SHARE schema.

In the case of ELRA resources conversion, there was a pre-conversion stage, where the ELRA format files were converted in simple XML flat list files, with the following format:

```
<resource id="1">
	<resource_id>1</resource_id>
	<resource_reference>ELRA-S0148</resource_reference>
	…
	…
</resource>
<resource id="2">
	…
```

Files with this format were the input to the XSL file that were converted in META-SHARE format. The tool Altova-MapForce, facilitated here as well the mapping process.

2.6.5. Conversion from META-SHARE v.2.0 to v2.1

All the previously mentioned LRs were converted to META-SHARE XML schema version 2.0. As there have been some minor changes in the schema from version 2.0 to version 2.1, the LRs had to be changed accordingly. The conversion from LRs in version 2.0 to version 2.1 has been performed by XSL transformations as well. The process of converting LRs from v.2.0 to v.2.1 is described in the META-SHARE installation manual, section 6.4.2 “Upgrade the XML Files to the Latest Metadata Format ”.

2.7. Add downloadable content to resources

Each META-SHARE resource can contain *downloadable content*. At the moment, it is possible to add *exactly one* archive file to a resource. Its name is fixed to `archive.{'zip', 'tar.gz', 'gz', 'tgz', 'tar', 'bzip2'}`.

After upload, the archive.ext, where .ext will be the file extension of the original file, will be placed inside the dedicated *storage layer* of the resource: each `StorageObject` has a local folder attached to it which is used to store associated downloadable data.

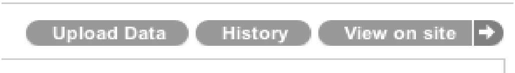
2.7.1. Adding downloadable content

There are two ways of adding downloadable content to an existing META-SHARE resource:

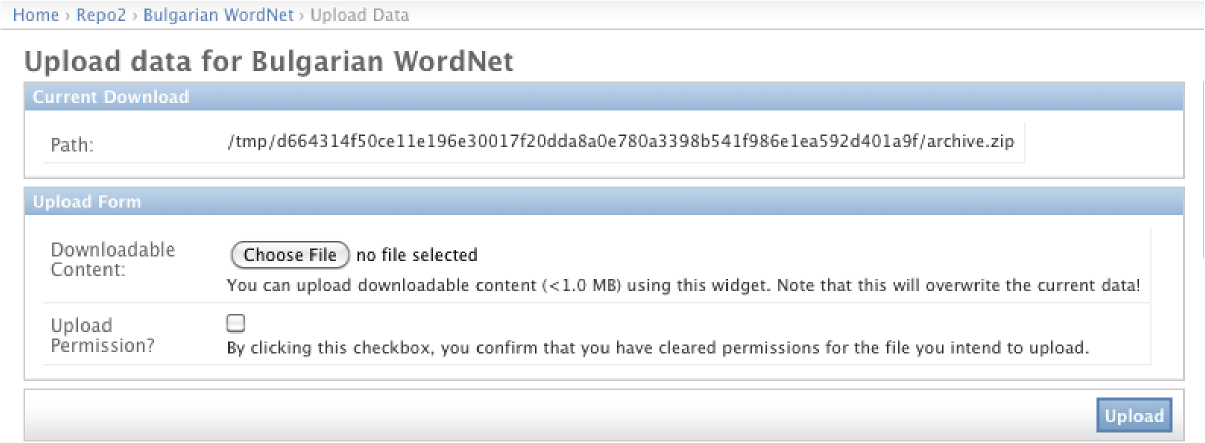
1. Upload via the Django admin backend, or
2. Upload using the command line `storage_admin.py` script

2.7.1.1. Upload via the Django admin backend

To upload data via the Django admin backend, the provider has to go to the Django admin editor page for *an existing resource*. It is *not* possible to upload files for new resources. On the top-right of the browser window, there is a button "Upload Data" next to "History".



Clicking the "Upload Data" button will load an upload page for the current resource.



Home › Repo2 › Bulgarian WordNet › Upload Data

Upload data for Bulgarian WordNet

Current Download

Path: /tmp/d664314f50ce11e196e30017f20dda8a0e780a3398b541f986e1ea592d401a9f/archive.zip

Upload Form

Downloadable Content: no file selected

You can upload downloadable content (<1.0 MB) using this widget. Note that this will overwrite the current data!

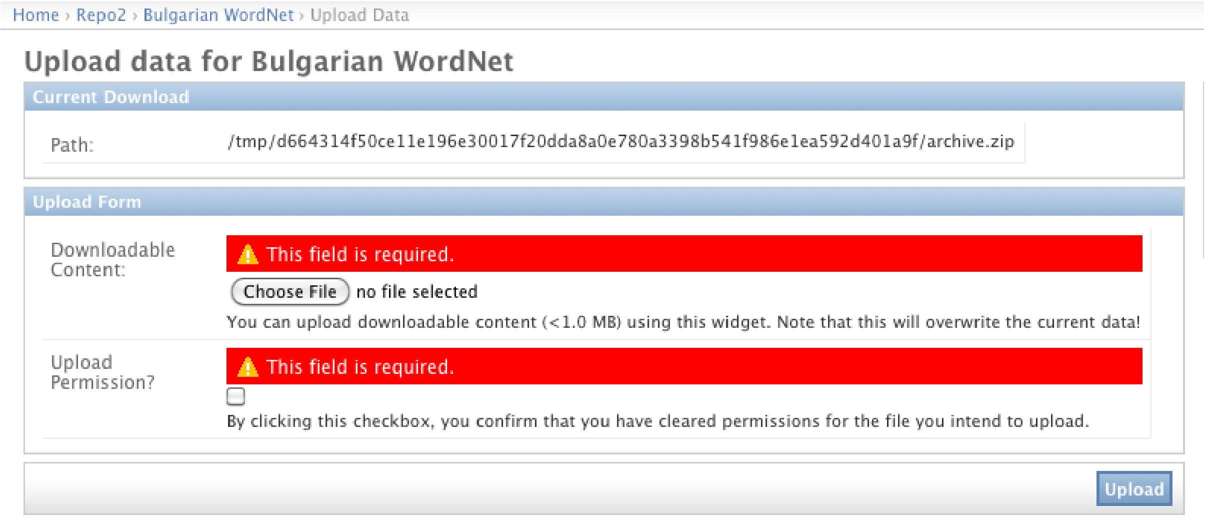
Upload Permission? □

By clicking this checkbox, you confirm that you have cleared permissions for the file you intend to upload.

The "Path" information is displayed for convenience and allows to quickly retrieve the file system path of the *StorageObject*'s downloadable content.

Note: It is required to click the "Upload Permission?" check-box to verify that the provider accepts liability for the file he intends to upload. The provider should only upload an archive to a META-SHARE resource if he is allowed to do so; if not, it is better to ask the original owner of the resource to do so.

Both fields are required. In case the provider attempts to upload without having filled them in, the fields are highlighted in red, with the message “This field is required”.



Home › Repo2 › Bulgarian WordNet › Upload Data

Upload data for Bulgarian WordNet

Current Download

Path: /tmp/d664314f50ce11e196e30017f20dda8a0e780a3398b541f986e1ea592d401a9f/archive.zip

Upload Form

Downloadable Content: ⚠ This field is required.

no file selected

You can upload downloadable content (<1.0 MB) using this widget. Note that this will overwrite the current data!

Upload Permission? ⚠ This field is required.

□

By clicking this checkbox, you confirm that you have cleared permissions for the file you intend to upload.

Once the upload is successful, the file is copied to the `StorageObject`'s storage folder as `archive.ext` (see above). The provider gets redirected to the editor page for the current resource, the page where he started the upload process.

Limitation: The admin backend upload page only allows files smaller than `MAXIMUM_UPLOAD_SIZE = 1 * 1024 * 1024` bytes to be uploaded. This is a robustness fix which ensures that admin page does not crash due to a very large file being uploaded. Section 2.7.1.2 offers a solution for uploading larger files.

2.7.1.2. Upload using the command line `storage_admin.py` script

When the provider wants to add a *large file* as downloadable content to an existing resource, he can use the `storage_admin.py` script, which is available within the `metashare/` folder.

If called without arguments, the script provides an overview on available options:

```
metashare $ python2.7 storage_admin.py
    usage: storage_admin.py MODE ARGS...

    MODE = [help, checksum, create, update, purge, folder, delete]
    Use storage_admin.py help MODE for detailed usage instructions.
```

To add a large file as downloadable content, the provider simply has to copy it over to the *storage folder* belonging to the resource. The *storage folder path* can be retrieved like this:

```
metashare $ python2.7 storage_admin.py folder <storage-object-id>
```

To retrieve the `<storage-object-id>` argument, the provider can go to the Django admin backend upload view (see above) and copy the last bit of the *storage folder* information. This is a 64-char hexadecimal String, e.g., `c0b9a36b50ce11e1a4570017f20dda8ac8066602350044549d62c1bdb750883c`.

Once having copied a file named archive.ext to the storage folder path for <storage-object-id>, the provider has to update the file checksum to "activate" the new downloadable content. This is the use of storage_admin.py:

```
metashare $ python2.7 storage_admin.py checksum <storage-object-id>
...
Checksum: d41d8cd98f00b204e9800998ecf8427e
```

If no file `archive.ext` can be found within the storage folder for `<storage-object-id>` the script will return `Checksum: None`.

2.7.2. Known Limitations

It is mandatory to follow the naming scheme. The provider has to make sure that the downloadable data is contained within *one* archive file. This file needs to have one of the following, allowed extensions (as defined in `storage/models.py`):

```
ALLOWED_ARCHIVE_EXTENSIONS = ('zip', 'tar.gz', 'gz', 'tgz', 'tar',
'bzip2')
```

3. Glossary

- ➔ Ingest: Within the META-SHARE context, ingesting a LR into the database is a one-time, irreversible step. It is preceded by a phase of editing an "internal" LR, visible only to the provider. Once the decision is taken that the metadata for a particular LR is "ready" to make the entry more widely visible, the provider "ingests" the LR into the database. An ingested LR is only visible to META-SHARE “staff” users.

- ➔ Media types: Within the META-SHARE context, the media types of a LR depend on its resource type. Thus, a Corpus LR can contain one or more of the following media types: text, audio, video, image, text-numerical and text-ngram. A Lexical-Conceptual Resource must contain a text media type and can also contain audio, video and image media types. A Language Description LR must contain a text media type and can also contain video and image media types. A Tool/Service does not contain any media types.

- ➔ Metadata: All the information that describes a given LR. Within the META-SHARE context, metadata can describe all different aspects of a LR, varying from details on the creation of the LR, to content or language or size information.

- ➔ Publish: Within the META-SHARE context, publishing a LR means making it visible to all META-SHARE users, registered and not registered.

- ➔ Resource type: Within the META-SHARE context, the resource type of a LR has to be strictly one of the following:

  - Corpus
  - Lexical-Conceptual Resource
  - Language Description

- Tool / Service

➔ Unpublish: Within the META-SHARE context, unpublishing a LR means making it visible only to META-SHARE “staff” users.

➔ XML metadata file: A XML format file that contains metadata on a LR.

➔ XML schema (XSD): An XML schema is a description of a specific XML type of files. It contains the structure that these XML files should have, along with all the constraints that should be respected so that the XML files are valid, such as the uniqueness of elements, the specific values that elements should contain etc.