

German	de	X			X	X	X	X	X	X	X	X	X	X	X	X	X
Greek	el	X			X												
Hungarian	hu	X															
Indonesian	id	X															
Italian	it	X		X	X				X								
Japanese	ja	X		X	X			X					X				
Norwegian	no	X		X			X		ToDo								
Polish	pl	X		X	X												
Portuguese	pt							X	X								
Portuguese PT	pt_PT	X			X			X	X	X							
Portuguese BR	pt_BR	X		X				X									
Russian	ru	X		X	X												
Spanish	es	X				X			X								
Swedish	sv	X			X			X	X								
Thai	th						X		X								
Ukrainian	uk					X			ToDo		x						
Welsh	cy					X	X	X	X								

Available translations of messages.xml for XML-UI

Language	Locale	1.5	1.5.1	1.5.2	1.6.0	1.6.1	1.6.2	1.7.0	1.7.1	1.8.0	1.8.1	1.8.2	3.0	3.1	3.2	4.0	4.1	4.2	
German	de	x	x	x	x	x	x	x	x	x	x								
Greek	el		x																
Catalan	ca	x	x							x									
Catalan (Central)	ca_ES			x															
Czech	cs			x	x	x		x	x	x									
Italian	it	x		x															
Japanese	ja		x					x	x										
Portuguese	pt_BR		x				x	x	x	x	x	x	x	x	x	x	x	x	x
Spanish	es	x	x					x	x	x	x	x	x	x	x	x	x		
Galician	gl											x							

Localized parts of DSpace not yet managed on Github

Messages.properties not corresponding to a valid Locale

Language (code)	Provided by	Status	Files	Comments
Simplified Chinese (chs)	JiehHsiang	open to review and extention	Messages_chs_properties.mht	Based on v1.3.1 of Messages.properties
Traditional Chinese (cht)	JiehHsiang	open to review and extention	Messages_cht_properties.mht	Based on v1.3.1 of Messages.properties

Other parts of DSpace which can be translated

DSpace 1.6.0

- [Set of localized files for Ukrainian](#)

DSpace 1.5.2

- [Set of localized files for Ukrainian](#)

DSpace 1.5.1

- [Japanese-language-packs-1.5.1.tar.gz](#)
- [Set of localized file for Spanish](#)

DSpace 1.4.2

File	French	German	Italian	Russian
all available files	French.zip	German.zip	Italian.zip	Russian.zip
dspace-source/config/emails/change_password	Change_password_fr.txt	Change_password_de.txt		Change_password_ru.txt
dspace-source/config/emails/feedback	Feedback_fr.txt	Feedback_de.txt		Feedback_ru.txt
dspace-source/config/emails/internal_error		Internal_error_de.txt		Internal_error_ru.txt
dspace-source/config/emails/register	Register_fr.txt	Register_de.txt		Register_ru.txt
dspace-source/config/emails/registration_notify		Registration_notify_de.txt		
dspace-source/config/emails/submit_archive	Submit_archive_fr.txt	Submit_archive_de.txt		Submit_archive_ru.txt
dspace-source/config/emails/submit_reject	Submit_reject_fr.txt	Submit_reject_de.txt		Submit_reject_ru.txt
dspace-source/config/emails/submit_task	Submit_task_fr.txt	Submit_task_de.txt		Submit_task_ru.txt
dspace-source/config/emails/subscription	Subscription_fr.txt	Subscription_de.txt		
dspace-source/config/emails/suggest	Suggest_fr.txt	Suggest_de.txt		Suggest_ru.txt
dspace-source/config/input-forms.xml		Input-forms_de.xml		
dspace-source/config/default.license		Default_de_license.mht		
dspace-source/config/news-side.html		News-side_de.zip		
dspace-source/config//news-top.html		News-top_de.zip		
dspace-source/jsp/help/collection-admin.html		Collection-admin_de.zip	Collection-admin_it.zip	
dspace-source/jsp/help/index.html		Index_de.zip	Index_it.zip	
dspace-source/jsp/help/site-admin.html		Site-admin_de.zip	Site-admin_it.zip	

Misc resources not part of DSpace

Language (code)	Provided by	Content	Files	Comments
Russian (ru)	Andrew Barhatov	State rubricator of scientific and technical information	Grnti.zip	

JSPUI Basic Technical Considerations

The [Java Standard Tag Library v1.0](#) is used.

The Use of Tags and Message Keys

Basic steps for using Tags in a JSP:

- include the Standard Tag Library in the JSP - `<%@ taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt" %>`
- define the key you want in `config/Messages.properties`
- use the key instead of the text in the JSP, see examples below
The messages may now be changed in the appropriate `config/Messages_xx.properties` file. This must be done at build-time, because it is placed in the `dspace.war`.

Example: Basic key use

- Old JSP

```
<h1>Search Results</h1>
```

- New JSP

```
<h1><fmt:message key="jsp.search.results.title" /></h1>
```

- Messages.properties

```
jsp.search.results.title = Search Results
```

Example: Use of parameters

Phrases may have parameters to be passed in, to make the job of translating easier, reduce the number of 'keys' and to allow translators to make the translated text flow more appropriately for the target language.

- Old JSP

```
<p>Results <%= r.getFirst() %> to <%= r.getLast() %> of <%= r.getTotal() %></p>
```

- New JSP

```
<fmt:message key="jsp.search.results.text">
<fmt:param><%= r.getFirst() %></fmt:param>
<fmt:param><%= r.getLast() %></fmt:param>
<fmt:param><%= r.getTotal() %></fmt:param>
</fmt:message>
```

- Messages.properties

```
<p>jsp.search.results.text = Results \{0\}-\{1\} of \{2\}</p>
```

(Note: JSTL 1.0 does not seem to allow JSP <%= %> expressions to be passed in as values of attribute in <fmt:param value="" />)

Configuring stop words for search

The list of [stop words](#) is configured in [DSAnalyzer.java](#). Like any other change in the DSpace source code, you are required to [rebuild DSpace](#) after modifying the list of stop words.

To Do

- In the servlet code:
 - In 'AuthorizeAdminServlet', some English is passed for direct display in the title through the attribute 'edit_title'. E.g. "Item".
- 'config/dstat.map'
- Support for easy customisation of UI without needing to fork language bundles.
- Display dates, numbers etc. according to Locale. (JSTL has tags for this)
- Searching in non-Western languages such as Chinese; Lucene cannot parse/separate words
- Multi-lingual metadata. Submission forms do not include language info for metadata. (The "language" DC field relates to the language of the *content*, i.e. the bitstreams.)
 - HPCLab, University of Patras, has submitted a new patch addressing this issue, against 1.4.2 (including the I18N patch) [here](#). Please test and verify in order to remove from to-do.
- Author and title ordering.
- Browse controls ("Jump to" 0-9, A-Z) – support for other characters
- Use of Collator fixes non-English language sort order problems. Not developed but small. No schema changes. Jim Downing
- I18n of non-Web UI – e.g. command line tools like the import/export tools. Currently the dictionaries are only stored in the .war file, so other parts of code won't have access to them.

Useful Tools

Python script to check for missing/extra strings

Given the English original (messages.xml or Messages.properties) and the corresponding translation file, this script will report both strings missing in the translation file and extra strings that shouldn't be in the translation. Tested on Python 2.7, requires ElementTree (wrapper for libxml2) for checking XML and /or jprops for testing .properties.

Here's also a slightly longer version which preserves the order of strings in the output: [dspace-i18n-check.py](#)

dspace-l10n-check.py

```
#!/usr/bin/env python

import sys

class MessagesXmlParser():
    def __init__(self, filename):
        import xml.etree.ElementTree as etree

        self.keys = []

        tree = etree.parse(filename)
        root = tree.getroot()
        for message in root:
            self.keys.append(message.attrib['key'])

class MessagesPropertiesParser():
    def __init__(self, filename):
        try:
            import jprops
        except:
            print('Error: jprops module for parsing .properties files is missing. Download and follow installation instructions from http://mgood.github.com/jprops/')
            sys.exit(2)

        self.keys = []

        with open(filename) as fp:
            for key, value in jprops.iter_properties(fp):
                self.keys.append(key)

if __name__ == "__main__":
    if len(sys.argv) != 3:
        print("Usage:")
        print("    %s messages.xml messages_XX.xml" % (sys.argv[0]))
        print("or")
        print("    %s Messages.properties Messages_XX.properties" % (sys.argv[0]))
        sys.exit(1)

    testfile = open(sys.argv[1], 'rb')
    if testfile.readline().find('<?xml') != -1:
        # xml file detected, assume messages.xml
        messages_tmpl = MessagesXmlParser(sys.argv[1])
        messages_in = MessagesXmlParser(sys.argv[2])
    else:
        # assume Messages.properties
        messages_tmpl = MessagesPropertiesParser(sys.argv[1])
        messages_in = MessagesPropertiesParser(sys.argv[2])

    print "Present in %s but missing in %s:" % (sys.argv[1], sys.argv[2])
    for i in set(messages_tmpl.keys) - set(messages_in.keys):
        print i
    print "\nPresent in %s but missing in %s:" % (sys.argv[2], sys.argv[1])
    for i in set(messages_in.keys) - set(messages_tmpl.keys):
        print i
```

Newer version of the script which allows comparison and dumping in columnar TSV of several messages files simulatenously: [dspace-i10n-check2.py](#). Usage example:

Usage examples

```
./dSPACE-i10n-check2.py --compare file1 file2
./dSPACE-i10n-check2.py --dump file1
./dSPACE-i10n-check2.py --dump-tsv-for-translation /dSPACE/webapps/xmlui/i18n/messages.xml /dSPACE/webapps/xmlui/i18n/messages_es.xml /dSPACE/webapps/xmlui/i18n/messages_it.xml
```

Python script to prepare a list of untranslated XMLUI strings when updating and older translation

A python script to generate a messages_xx.xml file by merging existing translations from an older committed translation to the latest messages.xml file. Thus, you get a file with the latest keys, where you only need to look for the untranslated strings.

<https://github.com/evelthor/dSPACE-xmlui-lang-util>

PHP script to check for missing/extra strings

Given the English original (messages.xml) and the corresponding translation file, this script will report both strings missing in the translation file and extra strings that shouldn't be in the translation.

checkkeys.php

```

<?php
if ($argc != 3) {
    echo "Usage: checkkeys.php <master> <tocheck>\n";
    exit(1);
}

$masterKeys = getKeys($argv[1]);
$toCheckKeys = getKeys($argv[2]);

print "IN $argv[1] BUT NOT IN $argv[2]:\n\n";
printMissing($masterKeys, $toCheckKeys);

print "\n\n\nIN $argv[2] BUT NOT IN $argv[1]:\n\n";
printMissing($toCheckKeys, $masterKeys);

function printMissing($reference, $test)
{
    foreach ($test as $value) {
        if (! in_array($value, $reference)) {
            echo "$value \n";
        }
    }
}

function readFileMessage($path)
{
    if (! $fileContent = @file($path)) {
        echo "Can't open $path \n";
        exit(1);
    }

    return $fileContent;
}

function getKeys($path)
{
    $fileContent = readFileMessage($path);

    return readKeys($fileContent);
}

function readKeys($file)
{
    $keys = [];

    foreach ($file as $key => $line) {
        if (strpos($line, '<message key=') === false) {
            continue;
        }

        $key = getKey($line);

        $keys[] = $key;
    }

    return $keys;
}

function getKey($line)
{
    $line = trim($line);

    $line = substr($line, 14);

    $charEnd = strpos($line, '>');

    $key = substr($line, 0, $charEnd - strlen($line));

    return $key;
}

```

Usage examples - checkkeys.php

```
php ./checkkeys.php file1 file2
php ./checkkeys.php /dspace/webapps/xmlui/i18n/messages.xml /dspace/webapps/xmlui/i18n/messages_es.xml /dspace
/webapps/xmlui/i18n/messages_it.xml
```

Perl script for unused keys

Here's a useful Perl script that finds all of the dictionary keys in the JSPs and the `Messages.properties` file, and prints out keys that are used in the JSPs but aren't in the dictionary, and vice versa. [Missing File: Find-unused-keys.pl](#) **Note: slightly out of date, can't detect message keys used in e.g. LocalizedMessage()** You'll need to change the variables at the top of the file to point to your DSpace source dir.

Eclipse Resource Bundle Editor

The [Eclipse Resource Bundle Editor](#) is a usefull tool to create and manage Messages.properties files. It is an "Eclipse plugin for editing Java resource bundles. Lets you manage all localized properties files in one screen. Some features: sorted keys, warning icons on missing keys/values, conversion to/from Unicode, hierarchical view of keys, more..."

Contributors

The University of Patras in Greece have done the initial hard work of separating out text from the JSPs into a dictionary file and linking to it. MARTON Jozsef has build on this by parameterising the UI text.