How to Translate Sakai

Contents

How to Start

Anyone interested in localizing/translating Sakai is strongly encouraged to join the sakai-dev apereo email list for questions (prefix the subject line with [i18n]). **I would recommend that you read this entire page before proceeding.**

If you are planning on working on a Sakai translation, please send an email to Gao Jun (gaojun.fudan at apereo.org) or sakai-dev@apereo.org. There may be someone else already working on translating Sakai to your target language.

Sun provides some general guides to translation at the Open Translation Website.

Before you can begin with the translation process, you will need to install and build Sakai from source. Directions can be found at Sakai Release Page (look for the Installation Guide in the Documentation section).

IMPORTANT: Sakai source code is no longer maintained in subversion, but in github. You'll need to create a fork of the Sakai github repository and clone it locally for a full checkout of the source for all core Sakai tools:

- https://github.com/sakaiproject/sakai
- https://help.github.com/articles/fork-a-repo/

Translating is a tedious job, and perhaps best to translate tool-by-tool. Try to do a cursory look over as many of the tool pages as possible to verify the correctness of the translation in context. If a translation doesn't look right, you can identify which property is mis-translated by selecting the 'en_DEBUG' language locale in user preferences (see below, Translation Help). Most tools (though not components) can be re-loaded without needing to restart tomcat, which can help with turn-around time when fixing up translations.

What to Translate

All of the localized text (ready for translation) resides in either HTML files or source-code "properties" files containing key=value pairs.

- Properties files values should be translated into locale-specific files, for example a Spanish language version of a properties file would be calendar_es.properties.
- Not all properties files contain localized text (some are configuration files). A nice list of configuration files to *exclude* is at http://qa1-nl.sakaiproject.org/international/trunk/de.html#excludedfiles
- Help text:
  - Sakai 2.9 (and earlier) help text is referenced from help.xml files located with each tool. In order to translate, a locale-specific help.xml file should be created (e.g. help_fr_CA.xml), which references the translated HTML help text. Note that a complete set of help.xml files must be created for the translated locale.
  - Sakai 10.0 (and later) help text can be converted to the correct format using the sakai-help-conversion tool. Please email the sakai-dev apereo email list for assistance.
- Alternate HTML content locations can be configured in sakai.properties:

> *server.info.url=/library/content/server_info_LOCALE.html*
> *myworkspace.info.url=/library/content/myworkspace_info_LOCALE.html*
> *webcontent.instructions.url=/library/content/webcontent_instructions_LOCALE.html*
> *webdav.instructions.url=/library/content/webdav_instructions_LOCALE.html*

Sun keeps a list of supported locales at http://java.sun.com/j2se/1.5.0/docs/guide/intl/locale.doc.html (if your locale isn't listed, we can create one for you). All translated text must be saved as ascii-encoded UTF-8 characters; the suggested tools below will do this for you.

Page and Tools titles are localized in the tools.properties file:

- `sakai/tool/tool-impl/impl/src/bundle/tools.properties` *(Sakai 2.5.x)*
- `sakai/config/localization/bundles/src/bundle/org/sakaiproject/localization/bundle/tool/tools.properties` *(Sakai 2.6.x)*

You might also want to translate the New User Notification email in:

- `sakai/site-manage/site-manage-impl/impl/src/bundle/notifyNewuser.xml`

There is still a small amount of text that has not been localized (e.g. permissions), but these are being worked on.

Beginning in Sakai 2.9, the citations helper will use a new way of configuring search sources in its config.xml file (which is loaded dynamically from a resources folder in the citations-admin site). This is done so changes can be made at run-time rather than at compile-time or startup-time. The config.xml file includes strings that should be internationalized. The format of that file is described in the I18N for Citations Helper in Sakai 2.9. A sample of the config.xml file will be included in the Sakai 2.9 release with English and Spanish translations for a couple entries. Once the form of that file is finalized, we would invite additional translations.

Independent Release Projects

Starting with the Sakai 2.7 release, several Sakai projects (e.g. sitestats) are being released as "*independent releases*" or "*indie*" projects, which means they are released separately and then bundled into the Sakai release. Because of this, you will not see the source code for certain projects when you "check out" Sakai from subversion. These projects may need to be checked out separately, and are listed at Releases.

Suggested Translation Tools

1. Crowdin.net and https://www.transifex.com/ - Collaborative online translation utilities

2. The ResourceBundle Editor 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...

3. The Benten Community Translation Platform provides broad functionality as detailed in the 2012 Sakai/JaSig presentations.

4. Auto-I18N - A simple utility to automatically translate a Java properties file into any number of other languages.

5. The **ResourceProperties Editor** is a platform-independent, downloadable tool which supports editing and translating properties files. Its available from http://sourceforge.net/projects/i18nedit with documentation at http://i18nedit.sourceforge.net/userdoc/. This editor keeps track of changes, additions, and deletions to the default bundle, allowing translations to easily be kept up to date. UTF-8 characters can be typed directly into the editor, and they will be automatically asci-encoded when the file is saved. ***NOTE: This tools does not appear to work with Java 1.6*** . *Note: You may need to create an i18n.properties file in the root of your Sakai directory for this tool to work. The file should contain the following two lines (the second line should include the language/locale you will be translating):*

> *locale.default=en*
> *locales=de fr ru (or similar))*

6. The **XLIFF Translation Editor** is also a platform-independent, stand-alone tool which supports editing and translating a wide variety of file types. While it doesn't track changes like the ResourceProperties Editor, it does provide many other features which makes it a powerful tool for translating help pages and other HTML-based pages within Sakai.

7. OmegaT is intended for professional translators. Its features include customisable segmentation using regular expressions, translation memory with fuzzy matching and match propagation, glossary matching, dictionary matching, translation memory and reference material searching, and inline spell-checking using Hunspell spelling dictionaries. (cf. http://en.wikipedia.org/wiki/OmegaT)

8. The **PropertyTransferer** is a web-based GUI utility from Smolny College of Liberal Arts and Sciences (St. Petersburg, Russia) to transfer property values from one property list to another, for updating localization files to the newer version. The source is available for a web-based tool at https://source.sakaiproject.org/contrib/tools/i18n/property-transfer/.

9. **Transformable and PreferAble:** The University of Toronto has developed a preferences tool that allows users to configure a personal stylesheet, which, for example, would allow users with a language preference such as Arabic or Hebrew to use a right-to-left stylesheet or skin. More information is at http://transformable.atrc.utoronto.ca/

10. **The Properties Comparator** : The Public University of Navarra (Spain) has developed a web tool for easily comparing two different properties files, selecting or editing the preferred translation of each property and generating a new file with the results. More info here. The source code is GPL, but is not still available in our repository as we want to test it first in our future migration to Sakai 2.9. If you want to check the source code, just write me at: daniel.merino AT unavarra DOT es.

11. There are (at least) four possible options for setting up a translation server:

- Pootle
- Eclipse Babel
- TranslateWiki
- Sakai Translation Server (Beta)

Translation Help

One way to discover the context of properties keys (i.e. where is the property key displayed on the page) is to enable a special debugging (en_US_DEBUG) locale by adding the following to sakai.properties

```
locales.more = en_US_DEBUG
```

Once this has been added and Sakai restarted, you can select the en_US_DEBUG locale within your user preferences to display the actual key value in place of the translated text.

Another option is to enable debugging in the ResourceLoader by adding the following to sakai.properties

```
log.config.count = 1
log.config.1 = DEBUG.org.sakaiproject.util.ResourceLoader
```

Logging message examples (warning: this is noisy, but useful for translation/context information):

> *DEBUG: getString(key) bundle name=announcement, locale=en_US, key=view.all, value=All (2008-01-15 14:55:04,546 http-8080Processor25_org.sakaiproject.util.ResourceLoader)*
> *DEBUG: getString(key) bundle name=announcement, locale=en_US, key=view.public, value=Public (2008-01-15 14:55:04,546 http-8080Processor25_org.sakaiproject.util.ResourceLoader)*
> *DEBUG: getString(key) bundle name=announcement, locale=en_US, key=gen.thereare, value=There are currently no announcements at this location.(2008-01-15 14:55:04,550 http-8080-Processor25_org.sakaiproject.util.ResourceLoader)*

After Translating

1. The translated properties files need to be in the source directories with the appropriate language extension (this is important). For example, sakai/discussion/discussion-tool/tool/src/bundle/discussion.properties translated into Japanese wouldbe copied to sakai/discussion/discussion-tool/tool/src/bundle/discussion_ja.properties. If you use the above ResourceProperties Editor, this will be done for you.
2. Next step would be to fully build and deploy Sakai, as described in the Source Installation Guide.
3. The default language locale for Sakai should be set either in Tomcat's catalina.sh (or catalina.bat) file or using "Preferences" after Sakai has started.

- The JAVA_OPTS environment variable specifies the language locale at startup. Following is an example that sets Japanese as the default language in the catalina.sh file:
  `JAVA_OPTS="$JAVA_OPTS -Duser.language=ja -Duser.region=JP"`
- The locales variable in the sakai.properties file specifies which locales are available in the "Preferences" tool. The following is the default value:
  `locales = en, ja_JP, ko_KR, nl_NL, zh_CN, es_ES, fr_CA, ca_ES, en_GB, ar`

Now you can start up Sakai! Any strings which have not been translated will use the default Resource Bundles (which are in English).

Adding Translations to Sakai Source

Once you've verified the translation, you'll need to create a JIRA and submit the changes to github to contribute them back to the Sakai community and future releases:

1. Create a Sakai JIRA with the component set to "Translation" https://jira.sakaiproject.org/secure/CreateIssue!default.jspa
2. Submit a Pull Request for your changes, which references the above JIRA

If you haven't signed a CCLA/CLA license for Sakai (https://www.apereo.org/licensing/agreements), please include the following at the top of every properties file:

```
#
# Copyright (c) 2015 Apereo Foundation
#
# Licensed under the Educational Community License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
# http://www.opensource.org/licenses/ECL-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
#
```

Alternately, those who have signed (open source) license agreements with the Sakai Foundation, may be given **Subversion Commit Privilege** for properties bundles.