i18n

Table of contents

The Nutch search pages are easy to internationalize.

For each language, there are three kinds things which must be translated:

1. page header: This is a list of anchors included at the top of every page.
2. static pages: These include the "about" page, the "search" page and the "help" page.
3. dynamic page text: These are strings used when constructing search result pages.

Each of the above is described in more detail below.

1. Getting Started

The things to translate are:

1. the page header
2. the "about" page (src/web/pages/lang/about.xml)
3. the "search" page (src/web/pages/lang/search.xml)
4. the "help" page (src/web/pages/lang/help.xml)
5. text for search results (src/web/locale/org/nutch/jsp/search_lang.properties)

If you'd like to provide a translation, simply post translations of these five files to nutch-dev@lucene.apache.org as an attachment.

2. Page Header

The Nutch page header is included at the top of every page.

The header is filed as src/web/include/language/header.xml where language is the IS0639 language code.

The format of the header file is:

<header-menu> <item> ... </item> <item> ... </item> </header-menu>

Each item typically includes an HTML anchor, one for each of the top-level pages in the translation.

For example, the header file for an English translation is filed as src/web/include/en/header.xml.

3. Static Page Content

Static pages compose most of the Nutch website, and are also used for project documentation. These are HTML generated from XML files by XSLT. This process is used to include a standard header and footer, and optionally a menu of sub-pages.

Static page content is filed as src/web/pages/language/page.xml where language is the IS0639 language code, as above, and page determines the name of the page generated: docs/page.html.

The format of a static page xml file is:

<page> <title> ... </title> <menu> <item> ... </item> <item> ... </item> </menu> <body> ... </body> </page> The <menu> item is optional.

Note that if you use an encoding other than UTF-8 (the default for XML data) then you need to declare that. Also, if you use HTML entities in your data, you'll need to declare these too. Look at existing translations for examples of this.

For example, the English language "about" page is filed as src/web/pages/en/about.xml.

4. Dynamic Page Content

Java Server Pages (JSP) is used to generate Nutch search results, and a few other dynamic pages (cached content, score explanations, etc.).

These use Java's Locale mechanism for internationalization. For each page/language pair, there is a Java property file containing the translated text of that page.

These property files are filed as src/web/locale/org/nutch/jsp/page_language.xml where page is the name of the JSP page in src/web/jsp/ and language is the IS0639 language code, as above.

For example, text for the English language search results page is filed as src/web/locale/org/nutch/jsp/search_en.properties. This contains something like:

title = search results search = Search hits = Hits {0}-{1} (out of {2} total matching documents): cached = cached explain = explain anchors = anchors next = Next

Each entry corresponds to a text fragment on the search results page. The "hits" entry uses Java's MessageFormat.

Note that property files must use the ISO 8859-1 encoding with unicode escapes. If you author them in a different encoding, please use Java's native2ascii tool to convert them to this encoding.

5. Generating Static Pages

To generate the static pages you must have Java, Ant and Nutch installed. To install Nutch, either download and unpack the latest release, or check it out from Subversion.

Then give the command:

ant generate-docs This documentation needs more detail. Could someone please submit a list of the actual steps required here?

Once this is working, try adding directories and files to make your own translation of the header and a few of the static pages.

6. Testing Dynamic Pages

To test the dynamic pages you must also have Tomcat installed.

An index is also required. You can collect your own by working through the tutorial. Once you have an index, follow the steps outlined at the end of the tutorial for searching.

This documentation needs more detail. Could someone please submit a list of the actual steps required here?