i18n Improvements Proposal

Currently this is a "splash page" for various i18n (internationalization) and localization improvements that have been suggested recently on DSpace mailing lists. Some of these discussion threads include:

- "Localization in Config Files" thread on dspace-tech (began March 31, 2012)
- "Introduction / A localization framework" thread on dspace-devel (began March 23, 2012)

The Problem Statements & Brainstorms

Overarching Issue: The translation process/workflow is extremely complex as it stands. It's downright frustrating for translators. If it stays this way, we will likely have more and more problems finding translators.

Lack of Compatibility With Translation Tools/Software

Issue Summary: There are plenty of translation tools/software to make this process easier. But, unfortunately, most/all? don't work with our current DSpace Cocoon XML (messages.xml) files.

Known Tools:

- GetText (GetText-Commons available for Java)
- Pootle
- Transifex
- Launchpad
- http://translationproject.org
- http://en.wikipedia.org/wiki/Computer_Assisted_Translation
- http://omegat.org/en/omegat.html
- https://en.wikipedia.org/wiki/Translation_Memory_eXchange standard for exchange of translation strings.

Brainstorming potential solutions:

- Investigate auto-converting both *.properties files and *.xml files to a single standard format (*.pot or similar) for translators. (idea from helix84)
    - .properties and .xml should be treated separately, as they are different in content not only in format (Christian Völker)
    - .properties has another issue: it's a mix of JSPUI text and text used by dspace-api, they might need to be separated out (Mark Wood)
- OR investigate standardizing DSpace i18n on a single format, likely *.properties. Though this may mean writing our own i18NTranslator class for Cocoon (as I don't believe the default one supports *.properties, but I could be mistaken). (Tim Donohue)

Path to Submit a new Translation is Frustrating

Issue Summary: The workflow of submitting a translation is also frustrating. As developers move towards more collaborative development models (i.e. GitHub), can we find ways to ease the collaboration of translators?

Brainstorming potential solutions:

- Investigate creating a new i18n submission workflow via GitHub? Maybe have a separate i18n GitHub repo, which we can provide translators with more direct access to or similar. During DSpace releases we can either pull updates from this repo, or even potentially release directly from it. (Tim Donohue, original idea from helix84)

Difficult to Locate All the i18n Files

Issue Summary: The proliferation of i18n messages files also makes translation frustrating, as it is harder to find all the keys needed to be translated (and harder for users to install all the translated messages files). Obviously we need to find a balance here between allowing DSpace to become more modular (which requires separate i18n messages files per module) and making locating these files easier.

Brainstorming potential solutions:

- Perhaps we need to better communicate or "standardize" the location of these i18n files across all modules? Currently i18n files are in different locations (subfolders) based on whether you are looking at the dspace-api, dspace-xmlui-webapp, or a newer module like 'dspace-discovery-xmlui-api'. There is seemingly no "standard" location to find these messages files. (Tim Donohue)
- In the past, I vaguely recall ideas around having a generic place to "drop-in" localized i18n files. For example, if there was a [dspace]/config/i18n/ folder, you could potentially "drop-in" your own custom version of various messages files for each module. Although this may not help translators, it could be useful for users. Imagine if you could just drop-in all your customized messages files (for XMLUI, API, Discovery) into a single location and DSpace would just use those – it would mean no more searching for the proper "drop-in" folder for each module. (Tim Donohue)

Not all Text is Easily Translatable

Issue Summary: Not all text to be translated is actually in i18n messages files (as we all know). Some are unfortunately still elsewhere (email files, *.cfg files, etc.).

Known Text That Is Not as Easily Translatable:

- email templates (under 'config/email')
- dspace.cfg ('dspace.name')
- Other configs under 'config/modules/' (curate.cfg, etc.)
- input_forms.xml
- Community & Collection descriptions (this is a result of Communities & Collections not having full Metadata support, like Items) See also DS-1134
- Certain XMLUI Value pairs (Three pairs of button labels: "Add" and "Remove Selected") (See: DS-1159)

Brainstorming potential solutions:

- Obviously, we need to investigate ways to bring more of that text either into translatable metadata fields (like Item metadata) in the DB, or migrate them to *.properties files (perhaps similar to how "item-submission.xml" file can references keys in messages files for <heading> text). This may need to be something we slowly improve over several releases. (Tim Donohue)
- Storing and transporting strings in .xml files rather than *.properties files hugely reduces the changes that tools incorrectly guess the character encoding.

What are other open source platforms using for i18n?

Please add to this list!

- Wordpress: http://codex.wordpress.org/Translating_WordPress
- Drupal: http://drupal.org/node/11130