

Localized Preferences

Note: This is not implemented yet

Context and Problem Statement

Currently, JabRef uses some localized preferences. Example: The email subject for sending references should be localized. A German user should use “Referenzen”, whereas the English default is “References”. In JabRef 5.x, it is implemented using `defaults.put(EMAIL_SUBJECT, Localization.lang("References"));` and `org.jabref.logic.preferences.JabRefCliPreferences#setLanguageDependentDefaultValues`.

We want to remove the localization-dependency from `JabRefPreferences`. The aim is to move the Localization to where the string is used.

The problems are:

- How to store default values?
- How to know if a user changed the string?
- What happens if the user changes the UI language? (If he configured a string, that should be kept. If the did not configure anything, the string should just change).

Considered Options

- Mark default value with `%`
- Localize defaults
- Store the preference only when it was changed by the user
- Store the unlocalized string

Decision Outcome

Chosen option: “Mark default value with `%`”, because it achieves goals without requiring too much refactoring and reuses a pattern already in use.

Pros and Cons of the Options

Mark default value with `%`

If user stores value, the value is stored as is. The default value is stored with `%` in front. A caller has to localize any stored value in the preferences if the string is ‘escaped’ by `%` as the

first character.

Code: If looked-up string starts with %, call `Localization.lang` of the substring (starting from 2nd character). Otherwise, use string as is.

- Good, because clear distinction between default value and user-supplied value.
- Good, because on update of JabRef's defaults, the string is not modified.
- Good, because already used in fxml files to indicate translatable strings.
- Good, because consistent to FXML.

Localize Defaults

Example: `defaults.put(EMAIL_SUBJECT, Localization.lang("References"));` in `JabRefGuiPreferences`.

- Good, because it is the current implementation
- Bad, because it does not allow for language switching

Store the preference only when it was changed by the user

If the preference was changed by the user, it should be stored. If there is no setting by the user, leave it empty. When a consumer gets such an empty preference, it knows that it needs to read the default and localize it.

- Good, because easy to implement.
- Bad, because this won't work if users actually want something to be empty.

Store the unlocalized string

Consumers then check the string they got as a preference against the defaults. If it matches, localize it. Otherwise, use it.
