Alchemer Mobile Translations

To enable full translation support on an application for an Enterprise account, please contact Alchemer directly to request translations be enabled.

Translations can be configured for all Alchemer Mobile interactions.

Import and Export

Translations may be exported and imported in the XLIFF format (version 1.2). The system will export XLIFF files bundled into a ZIP archive for convenience, and can export all strings or just the set of missing/out-of-date strings. The UI does not support importing ZIP archives, but does support importing multiple XLIFF files simultaneously.

You will need to export the bundle of out-of-date (i.e. *stale*) translations, deliver them to a translation service to add or update the translated strings, and, once they are returned to you, import the updated files back to your Alchemer Mobile dashboard.

The XLIFF files contain references to particular Interactions (i.e. by database ID), so it is not possible to import a file that was not previously exported from the site. Also, some operations (i.e. modifying a Survey) may generate a new Interaction with a new ID, invalidating any translations files that refer to the old ID. In this case, the import process will silently ignore unknown Interactions.

Translation Status

The Translation system tracks the status of translated strings for each locale:

- *Good* - all translation strings are present and up-to-date
- *Missing* - some or all of the translation strings are not present
- *Out-of-date* (stale) - some or all of the translation strings are present, but are out-of-date relative to the default locale. Translations will appear as out-of-date if you update the standard string but do not update the corresponding translation.

Changing a string in the default locale will mark that string as out-of-date in all other locales, to indicate that they need re-translation.

Fallback Policy

You can configure different behavior for cases where the translation strings in a particular locale are missing or out-of-date.

- *Fallback* (default) - deliver strings in the default locale if the requested locale is missing or

out-of-date

- *Exclude* - deliver strings in the default locale if the requested locale is out-of-date, don't deliver Interactions with missing translations
- *Strict* - only deliver an Interaction if the strings in the requested locale are complete and up-to-date

Application Settings

Each Application has a number of translation-related settings, some of which may be modified through the UI:

- *locales* - a list of locales supported by the application (by language code with optional dialect) . By default, these are all the locales for which default strings are available (currently: ar, da, de, en, es, fr, fr-CA, it, ja, ko, nl, pt, ru, sv, zh, zh-TW). Adding a new locale to this list will automatically add the default strings for that locale.
- *default*locale_ - the primary locale of the app (defaults to "en"). Changes to strings in this locale will mark those strings as out-of-date in all other locales (see Translation Status). This locale may be used if translation strings are not available in a requested locale (see Fallback Policy).
- *locale*policy_ - specifies that behavior when a particular locale doesn't have all translations up-to-date (see Fallback Policy)

Languages and Dialects

The only assumption made about dialects is that an unknown dialect will fallback to the base language (e.g. "en-XX" will fallback to "en"). We make no assumptions regarding which dialect is the "correct" default for a language, as this is largely a matter of opinion. You are free to rename dialects to languages to change the base language. For example, the default "en" translations are in the "US" dialect. If you are a business in the UK, you might want to make "en-GB" the default dialect for "en". You could do by changing "en-GB" to "en" in the XLIFF file before importing. You can also export our default "en" translations, rename them as "en-US", and re-import.

Note that locale names are automatically normalized to the form seen above, i.e. a lowercase language code and uppercase region code separated by a hyphen. Chinese dialects that specify "hans" or "hant" (e.g. "zh-hans-HK") are normalized to "zh-CN" and "zh-TW", respectively, and the region code is ignored.

Video: Using Translations

Important Details for iOS

In order for translations to show properly in your iOS app, you must specify that the available languages in Xcode prior to release. To do so, open your project in Xcode, then go to the localization settings and add the target language.

If an app is not provisioned for the language your device is using, it will fall back to the highest priority language that the app is provisioned for.

Related Articles