



= Lost in Translation ?

现在什么？ _ _ _ هناك التطبيق لذلك!

Goal: Translating an App should not involve development, existing translations should be kept.

Problems, Tools, Best Practices.

== Problems

conflicting goals:

not in focus of english speaking developer/tester # Apple tools are a bit odd, e.g. [genstrings](#) (not incremental) # xibs even worse to keep up-to-date # OMG: images! # hard to verify: fringe screens, bilingual devs

== Interlude: Localisation != Translation

Localisation: **prepare for** translation (NSLocalizedString, **dev** task) # Translation: **do** translation (Localizable.strings, **non-dev** task) == Not in focus of english speaking developer/tester

most apps + devs english ⇒ no goal, # annoying NSLocalizedString(@„foo“, @„bar“) when **developing**, # but: expensive to hunt down when **releasing**,

```
# hard to find automatically - many technical strings
# 'NSLocalizedString' & friends mandatory to leverage 'genstrings'
```

== So,

make a habit to wrap **every** @„english text“ in NSLocalizedString(@„english text“, @„view hint“) whenever you meet one, # don't even bother about defining a mnemonic key but use just the **english** literal/format string. # (mnemonic: 2x the typing plus clueless translators)

== Apple tools are a bit odd

IMO there's no way around genstrings, despite it's shortcomings:

doesn't **update** but overwrite, # UTF-16 - not (git) diff friendly, # macro NSLocalizedString doesn't tell source file.

=== Solution to 1 + 2

[genstrings_incremental](#)

Nice: no xml/xliff, just plain text word lists for translation.

== NSLocalizedString

```
/* TBA */
"TBA" = "TBA";
```

or

```
/* ContributeVC
Feedback */
"Submit" = "Absenden";
```

which one would you choose?

```
LocalizedString(@"Submit", @"ContributeVC")
```

== Xcode project target membership

* we're not doing it 100% how Xcode likes it, * reason: define translations per target, **not** whole project * caution: keep Localizable.strings in sensible directory



== tl;dr

no literals in xibs, inject text via IBOutlet, # prefer **rendered text**, avoid images, # wrap **every** user visible string **whenever** you find one, # always localise the **format string**, # always **preserve** existing, used translated phrases: [genstrings_incremental](#).

== Thank you

for your attention.

* [ADC: Programming Resources](#) * [ADC: NSShowNonLocalizedString & NSShowNonLocalizableStrings](#)

```
[[NSUserDefaults standardUserDefaults] setBool:YES
 forKey:@"NSShowNonLocalizedString"];
[[NSUserDefaults standardUserDefaults] setBool:YES
 forKey:@"NSShowNonLocalizableStrings"];
```

* [genstrings_incremental](#) * [Google docs](#)

From:
<http://wiki.mro.name/> - □

Permanent link:
<http://wiki.mro.name/iphone/localisation>

Last update: **2013/01/29 23:50**



