

# intl me this, intl me that

Andrei Zmievski  
Chief Architect  
Outspark

OSCON ~ July 24, 2008

Why localize?

[главная](#)[новости](#)[модели](#)[агентства](#)[рейтинги](#)[статьи](#)[стиль](#)[форум](#)[о проекте](#)

**модель месяца**  
июль 2006

**Анна Жукова** >>>



Агентство: Президент



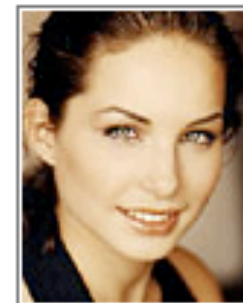
**лауреаты**

**Римма Агафошина** >>>>

Red Stars

**Вера Сальникова** >>>>

SieS



**Твоя электронная визитка  
в мире модельного бизнеса**

One example.



Most important thing...

No assumptions!

# No assumptions!

- ✧ English is just another language
- ✧ USA is just another country
- ✧ Earth is just another planet (eventually)

# i18n

- ✦ PHP 5.3 or PHP 6
- ✦ **intl** extension
- ✦ Consider all data processing and output points

# Locale data

- ✦ Common Locale Data Repository (CLDR)
- ✦ 374 locales: 137 languages and 140 territories
- ✦ Updated regularly
- ✦ Used by **intl** extension



# Translation

- ✧ Identifying what to translate
- ✧ Checking all sources
- ✧ Obtaining translation
- ✧ Iteration

# What to translate

- ✧ Translatable units  
Continue or There were 5 search results
- ✧ Approaches
  - ✧ Automatic “rippers”
  - ✧ Manual markup

# Sources: PHP

- ✦ Anything destined for output layer
  - ✦ single- and double-quoted strings
  - ✦ heredocs
  - ✦ error/exception messages (if seen by messages)
  - ✦ 404 pages, anyone?

# Sources: PHP

- ✦ Use output buffering to detect misses
- ✦ Consider templates to enforce separation
- ✦ Don't use extensions that cannot deal with UTF-8

# Sources: JS and CSS

- ✦ Text
- ✦ Images
- ✦ Position or alignment of elements may change
- ✦ Modularize locale-dependent code into separate files

```
<script src="/js/common.js" type="text/javascript"></script>  
<script src="/js/locale-<?php echo $LOCALE ?>.js"  
        type="text/javascript"></script>
```

# Sources: DB

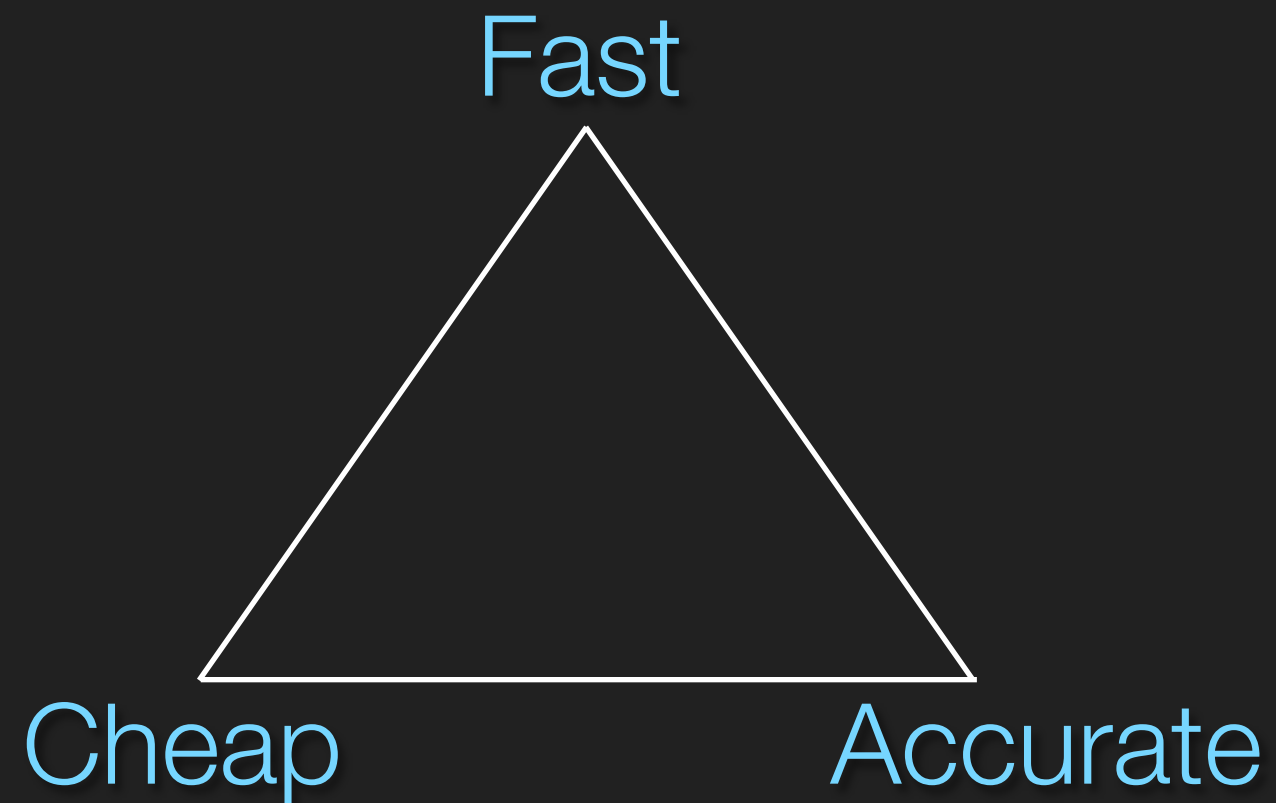
- ✦ Strings are fine, if they will never be displayed to users
- ✦ Consider using constants/identifiers, e.g. not `admin` or `user`, but `1` or `2`
- ✦ For things like product titles, keep separate table with translations and link against the main one

# Sources: external

- ✧ File-based content
- ✧ RSS Feeds
- ✧ Web services
- ✧ et al



# Obtaining translations



# Obtaining translations

- ✧ You
- ✧ (maybe) Fast and cheap - not accurate
  - "Not to perambulate the corridors during the hours of repose in the boots of ascension."  
(sign in an Austrian ski hotel)

# Obtaining translations

- ✦ Professionals
- ✦ (usually) Accurate and fast - not cheap

# Obtaining translations

- ✦ Community
- ✦ (fairly) Accurate and cheap - not fast

# Facebook approach

- ✦ Turn translation into a competitive activity
- ✦ Build it into the interface (just another app)
- ✦ Validation via voting

# Iteration

- ✧ Catching new units
  - ✧ mark up untranslated strings
  - ✧ use mnemonic identifiers,  
e.g. **MENU.NAV.HELP**
- ✧ Merge/update tools

# Using translations

- ✧ Self-contained pages (masochistic)
  - ✧ standalone per-locale pages with no common root
  - ✧ “quick-n-dirty”
  - ✧ iteration? not so much



# Using translations

- ✧ Runtime
  - ✧ uses translation storage and on-the-fly lookup
  - ✧ usually combined with caching

# Using translations

- ✦ Pre-generation (“baking”)
  - ✦ complete per-locale sites generated offline
  - ✦ no runtime lookups
  - ✦ may require runtime operations (sorting, etc)
  - ✦ could increase opcode cache memory requirements

# Considerations

- ✧ Fidelity
- ✧ Ease of use
- ✧ Performance
- ✧ Flexibility
- ✧ Portability

# Fidelity

- ✧ UTF-8
  - ✧ don't use tools that don't support it

# Fidelity

- ✧ How big should translatable units be?
  - ✧ “As large as possible, but not larger”
  - ✧ Avoid concatenation problem

**There are `<?php echo $nMesg ?>` unread messages  
in `<?php echo $nFolders ?>` folders.**

# Fidelity

- ✧ How big should translatable units be?
  - ✧ “As large as possible, but not larger”
  - ✧ Avoid concatenation problem

**There are <?php echo \$nMesg ?> unread messages  
in <?php echo \$nFolders ?> folders.**

# Fidelity

- ✧ How big should translatable units be?
  - ✧ “As large as possible, but not larger”
  - ✧ Avoid concatenation problem

**There are `<?php echo $nMesg ?>` unread messages  
in `<?php echo $nFolders ?>` folders.**



# Fidelity

- ✦ Sometimes the largest possible unit is a word
- ✦ Context is important  
chinese (person) vs. chinese (language)
- ✦ Add context as part of the unit  
chinese-person or CHINESE.PERSON

# Fidelity

- ✧ Combining translations with runtime data (parametrization)

**There are %1 unread messages in %2 folders.**

- ✧ **sprintf()** - works for simple things
- ✧ **gettext()** - can help with plurals
- ✧ **MessageFormat** + **ChoiceFormat** is better

# Ease of use

- ✧ Intuitive tools (or good documentation)
- ✧ Transparent formats
- ✧ Translation memory
  - ✧ useful for short, precise matches, not fuzzy
  - ✧ use in testing and first pass, not in production

# Performance

- ✦ Caching
  - ✦ translation units
  - ✦ translated pages
- ✦ APC, memcache, etc
- ✦ Reduce runtime overhead

# Flexibility

- ✦ Adding new languages/locales quickly
- ✦ Translation inheritance

# Portability

- ✧ Moving between tools
- ✧ Translations, most importantly
- ✧ XLIFF
  - ✧ <http://en.wikipedia.org/wiki/Xliff>

# Tools: gettext

- ✧ Developed for C/C++ originally
- ✧ Somewhat obscure format
- ✧ Translations on disk
- ✧ Have to compile translations with every change
- ✧ Proper markup not always possible
- ✧ POedit is a decent translation editor



# Tools: ezTranslation (et al)

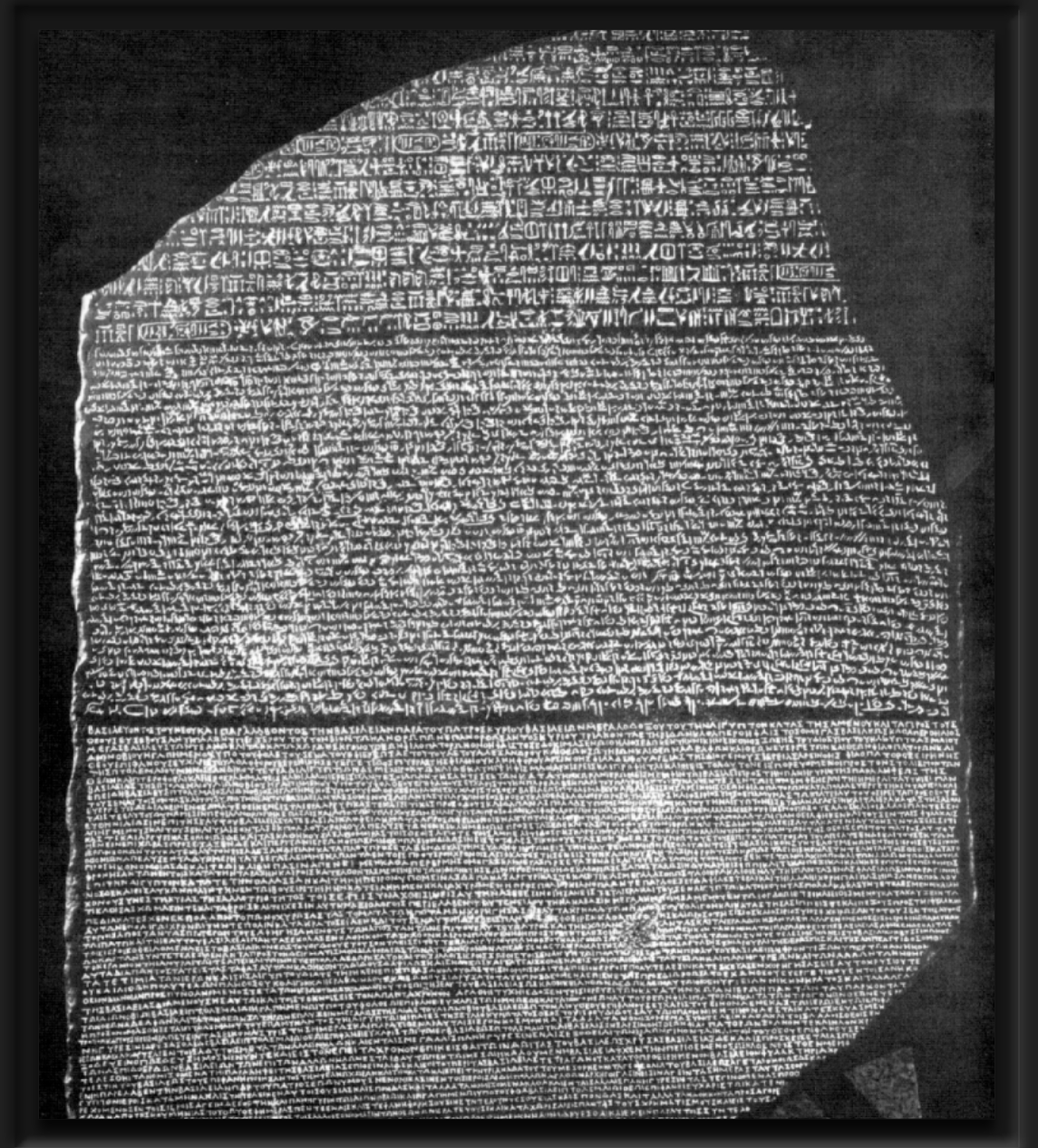
- ✦ More of a translation look-up tool
- ✦ Can support various backends for translation storage and caching (QT Linguist format by default)
- ✦ Supports parametrized strings
- ✦ Bork/I33t filters for marking untranslated strings

# Tools: template engines

- ✦ Smarty
- ✦ 3rd party solutions based on pre- and post-filters
- ✦ Translations in config files or gettext mainly, could be in DB
- ✦ Mark-up approaches vary
- ✦ Parametrized strings are possible (depends on plugin)

# Tools: r3

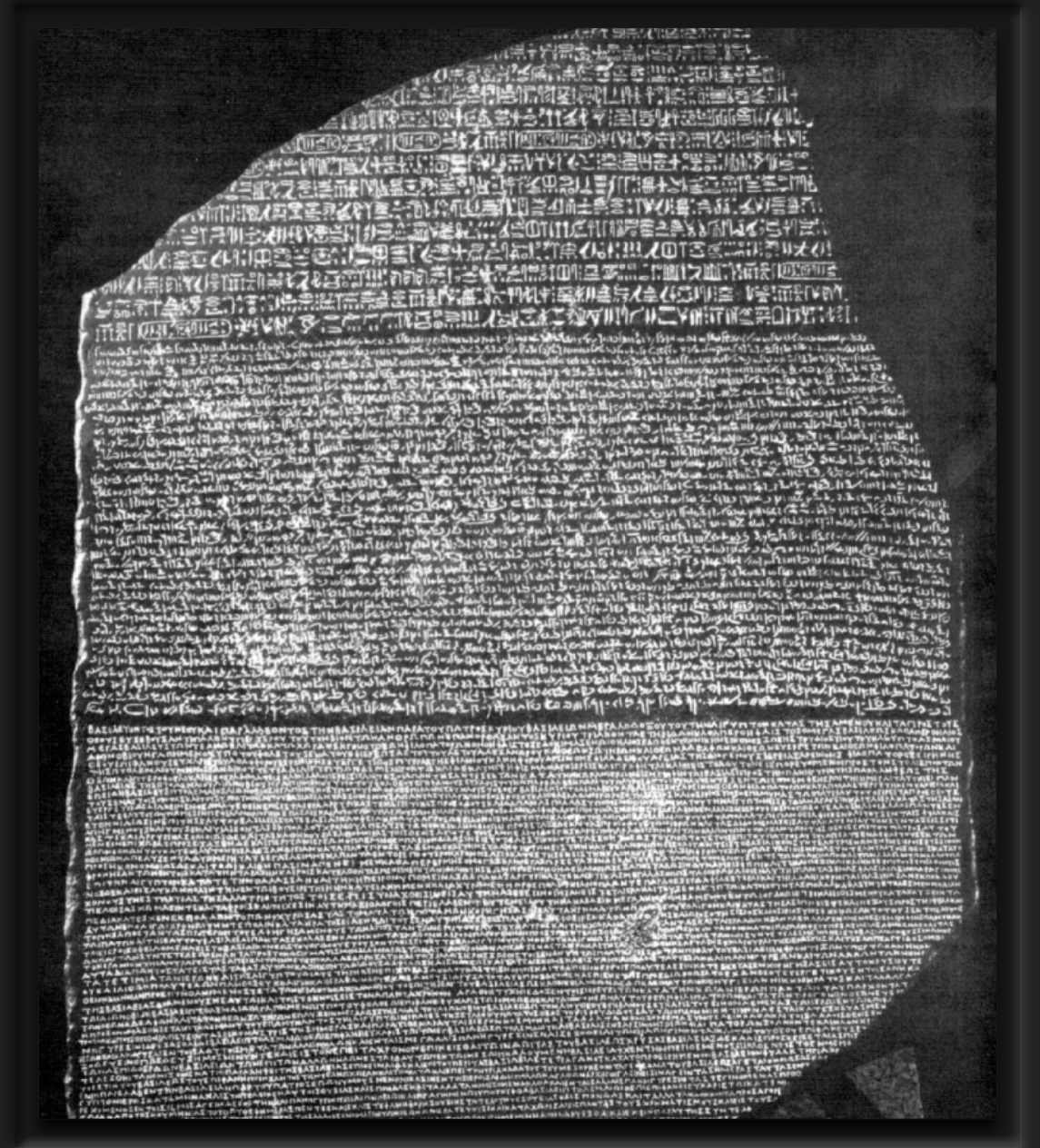
- Developed and supported by Yahoo!
- Very flexible and powerful, but a bit of a learning curve
- Translations are a subset of “site variations”





# Tools: r3

- ✦ Inheritance everywhere
- ✦ Translations in DB (MySQL or SQLite)
- ✦ Has basic GUI for some operations



# Tools: intl

- ✧ Available for PHP 5.3 and PHP 6
- ✧ Access to locale data
- ✧ Formatters/parsers
  - ✧ Number, date, time, message, choice, etc
- ✧ Collation (sorting)
- ✧ More coming

# r3: setup

```
% sudo sudo pear install -f stickleback-[version].tgz
% sudo pear install -f --alldeps r3-[version].tgz

% mkdir ~/r3
% r3 setup setuphome ~/r3
% export R3HOME=~/r3

% r3 setup installdb
```

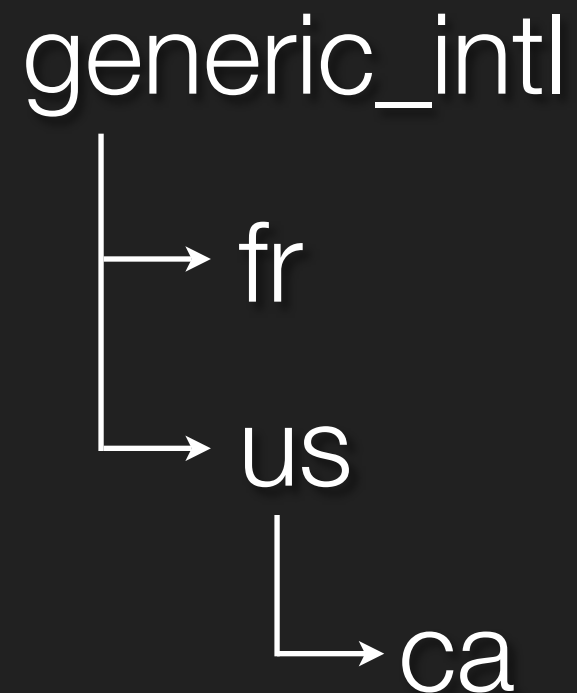
# r3: setup

```
% r3 dim product create wine
% r3 dim intl create generic_intl
% r3 dim intl create -p generic_intl us
% r3 dim intl create -p generic_intl fr
% r3 dim intl create -p us ca

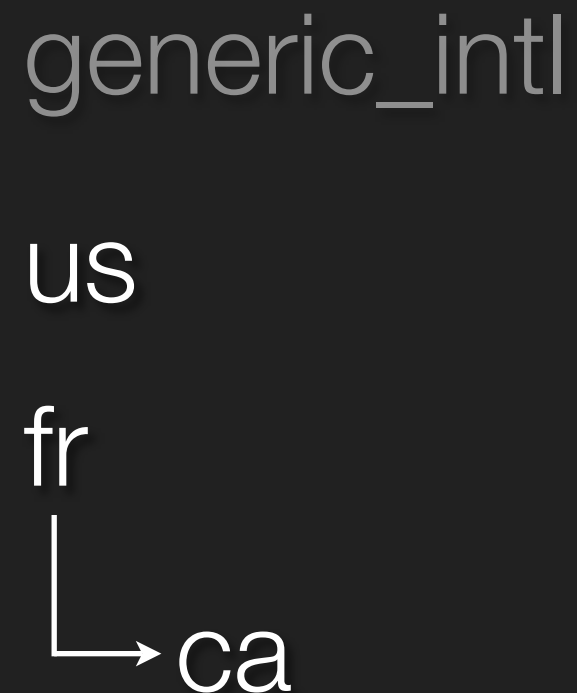
% r3 dim intl parent ca set fr -d translation
% r3 dimension intl parent fr unset -f -d translation
...
```

# r3: inheritance

## templates



## translations





# r3: make a page

```
% r3 target create wine/generic_intl/index.php
% r3 template edit wine/generic_intl/index.php \
    index.php.ros
...
% r3 generate -av
```

# r3: markup

```
<r3:trans>The Wine Source</r3:trans>
```

```
% r3 translation list  
% r3 translation set wine/fr 'The Wine Source' \  
    'La Source de Vin'  
% r3 generate wine/fr
```

# r3: translation

```
% r3 translation save all fr.xml
...
% r3 translation merge fr.xml
```

```
<file original='wine/fr/generic'
      source-language='en'
      target-language='fr'
      datatype='plaintext'>
  <body>
    <trans-unit id='26'>
      approved='yes'>
        <source>The Wine Source</source>
        <target>La Source de Vin</target>
      </trans-unit>
```

```
...
```

# r3: compile-time PHP

test.html.ros

```
<div>
<r3:cphp>
foreach (range(1, 5) as $i)
{
    echo '$i,' <br/>';
}
</r3:cphp>
</div>
```

test.html

```
<div>
1
2
3
4
5
</div>
```

# r3: parameterized strings

test.php.ros

```
$message = "<r3:trans>You have {0,number} messages as  
           of {1,date,full}</r3:trans>";  
$args = array(1234, time());  
echo MessageFormatter::formatMessage(  
    $LOCALE, $message, $args  
);
```

**fr** translation

Au {1,date,full} vous avez {0,number} messages.

**fr** output

Au mardi 22 juillet 2008 vous avez 1 234 messages.

# Resources

- ✦ **r3**

<http://developer.yahoo.com/r3/>

- ✦ **gettext**

<http://zez.org/article/articleview/42/>

<http://www.poedit.net/>

- ✦ **Smarty**

Chapter 12 of Smarty book

<http://smarty.incutio.com/?page=SmartyMultiLanguageSupport>

<http://bit.ly/2q2XM1>

- ✦ **ezTranslation**

<http://ezcomponents.org/docs/tutorials/Translation>

- ✦ **intl**

<http://php.net/intl>

thanks

спасибо

<http://gravitonic.com/talks>