



Serge

A Free, Open Source Platform
for Continuous Localization
from Evernote

Igor Afanasyev
Director of Localization, Evernote

text.your
questions: **650 443 85 26**



15,000 words/month

25%

software

Android, iOS
Mac OS, Windows
Web

75%

marketing

Website
Google Docs
Zendesk KB

35 projects
(components)



text.your questions: 650 443 85 26



Localization Team:



Director / L10N Engineer
(backend)



L10N Manager/Engineer



Software Engineer
(translation server)

Our localization process
is **fully automated**

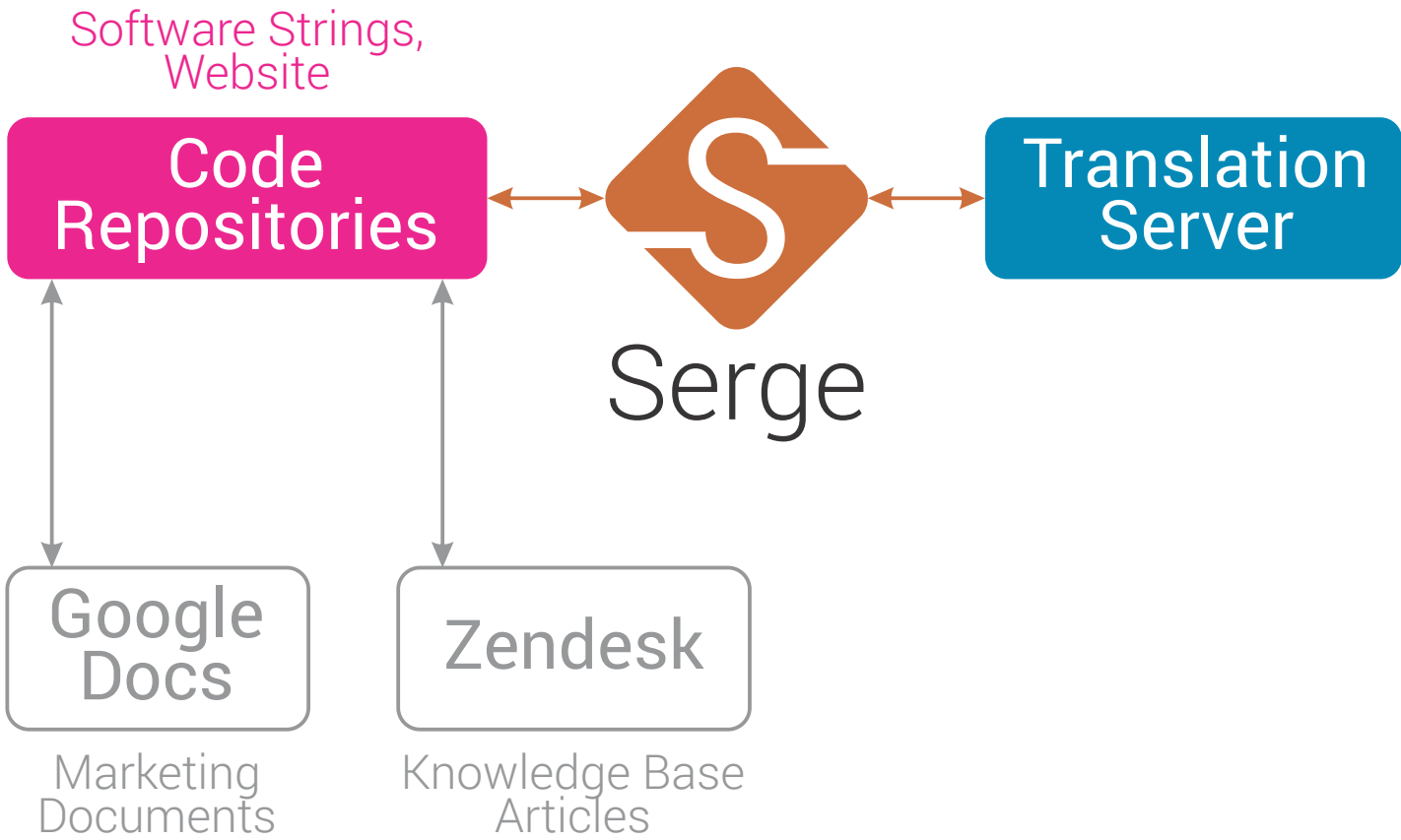

agile, scalable



Serge

String Extraction and Resource Generation Engine

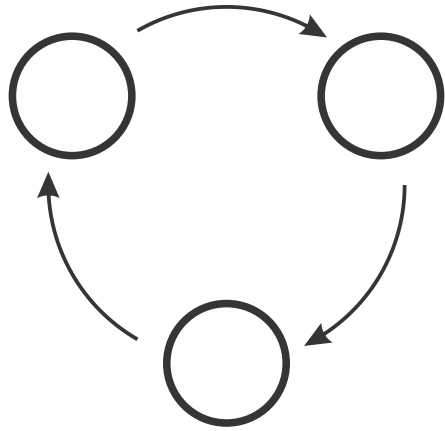
text.your
questions: 650 443 85 26



text.your questions: **650 443 85 26**

```
$ serge sync myproject.serge
```

text.your
questions: 650 443 85 26



You write configuration files (once)
and put them on your localization server

Serge runs on your server
(in a continuous loop)

Serge generates/updates
translation interchange files and
pushes them to your translation server

Serge generates and pushes localized
resource files into VCS; this triggers CI builds

Serge vs LSP Automation APIs

Serge doesn't send your *original* resource files for translation. It uses *translation interchange files*

You have full control of parsers. You have your own TM

Serge is fully extensible, has many built-in plugins, can be used not only for translation itself



Serge requires minimal “last-mile” integration

Serge Configuration Files

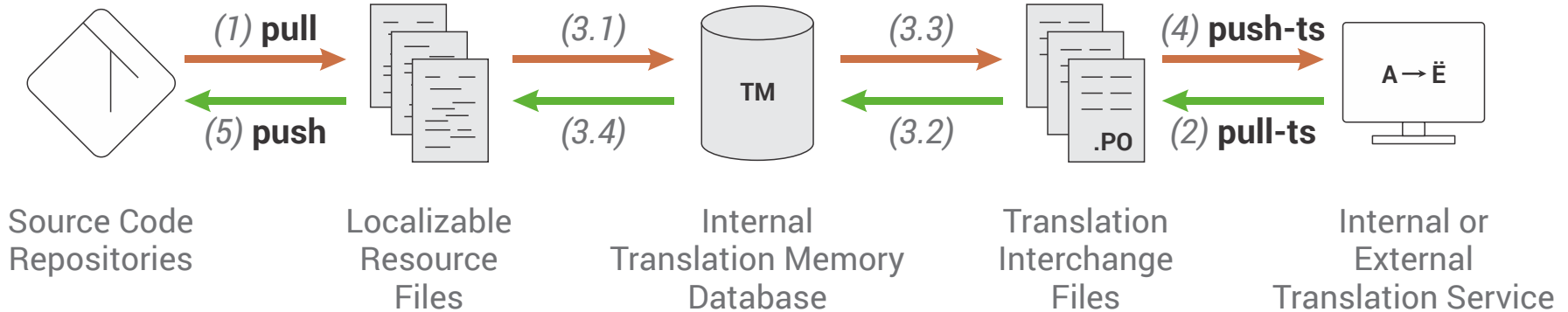
serge.io/docs

```
sync
{
  ts
  {
    # Translation Service parameters
    # ...
  }
  vcs
  {
    # Version Control System parameters
    # ...
  }
}
jobs
{
  {
    # Job 1 parameters
    # ...
  }
  {
    # Job 2 parameters
    # ...
  }
}
```

```
$ serge sync myproject.serge
```

sync

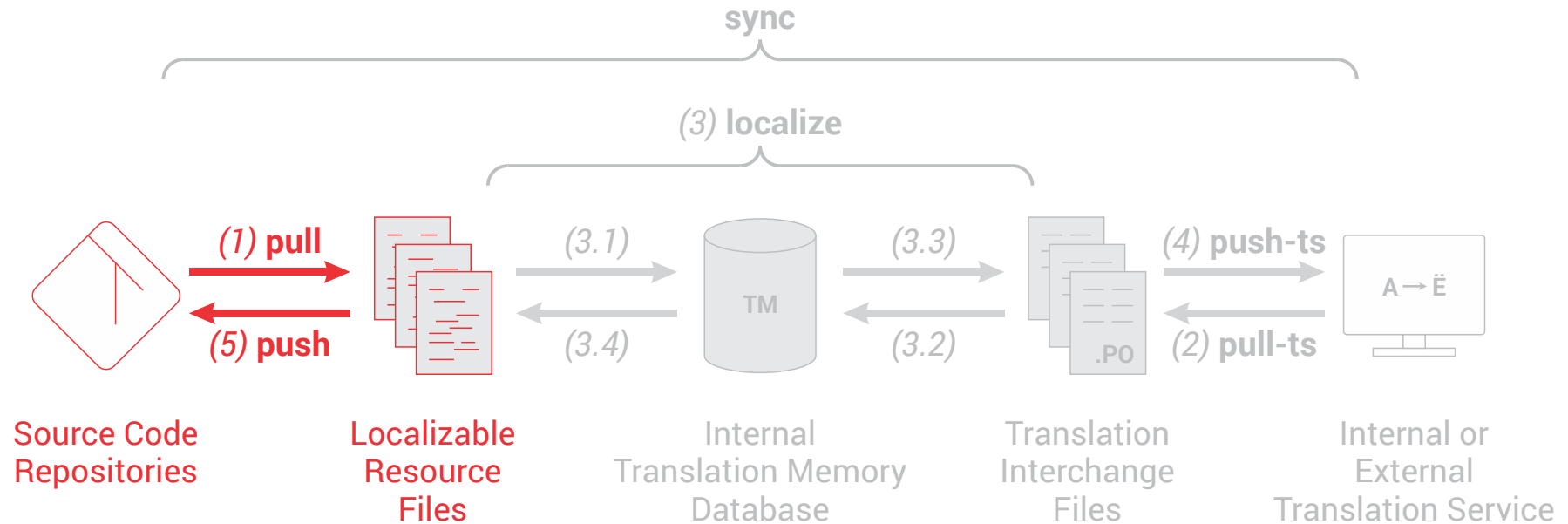
(3) localize



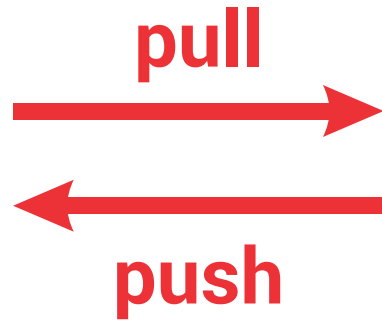
text.your questions: 650 443 85 26

```
$ serge pull myproject.serge
```

```
$ serge push myproject.serge
```



Integration with Version Control Systems



Plugins: Git, Gerrit, Subversion, Mercurial

Isolation from different VCS logic

Multiple repositories as one logical internationalization project

Automatic conflict resolution

```
sync
```

```
{  
  ts  
  {  
    # ...  
  }  
}
```

```
vcs
```

```
{  
  plugin git  
  data  
  {  
    add_unversioned YES  
    email l10nrobot@evernote.com  
    name L10N Robot  
  
    local_path /var/serge/data/android  
    remote_path  
    {  
      main ssh://l10n@git.evernote.com/android#develop  
      widget ssh://l10n@git.evernote.com/android-widget  
    }  
  }  
}
```

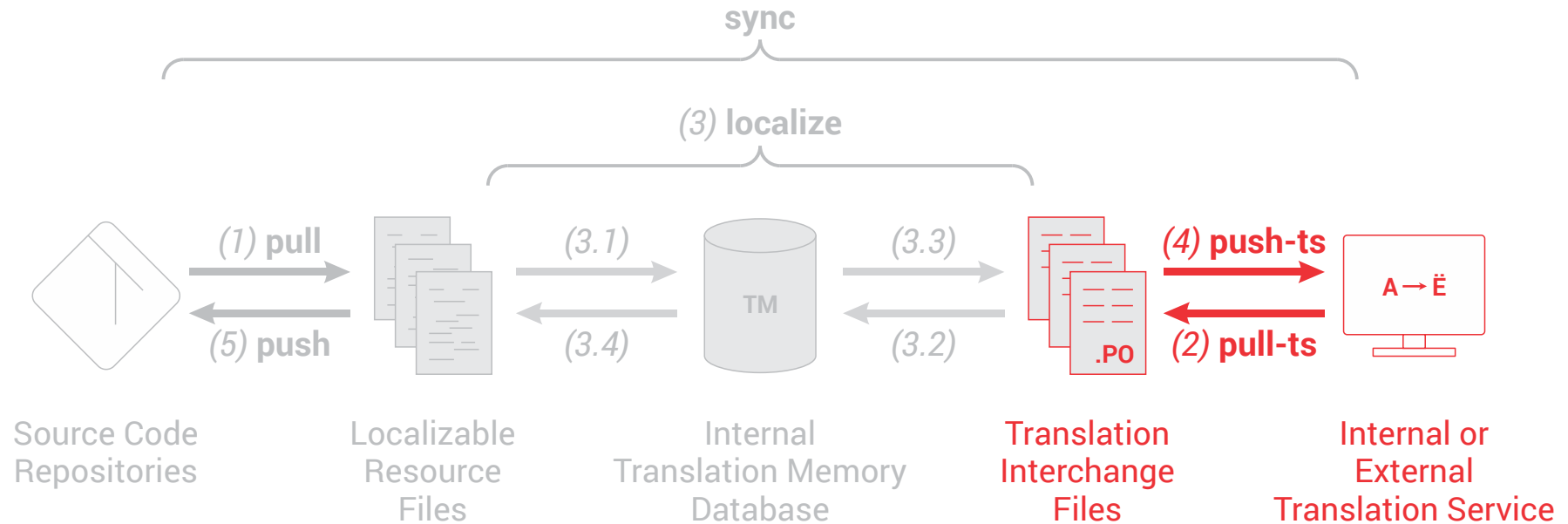
```
}
```

```
jobs
```

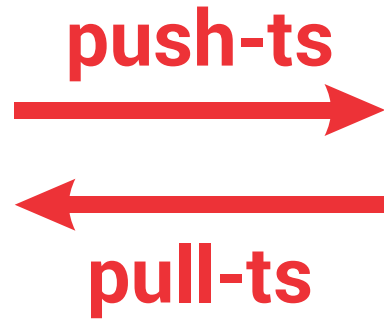
```
{  
  # ...  
}
```

```
$ serge push-ts myproject.serge
```

```
$ serge pull-ts myproject.serge
```



Integration with Translation Services



Plugins: Pootle (pootle.translatehouse.org)
Our server: translate.evernote.com

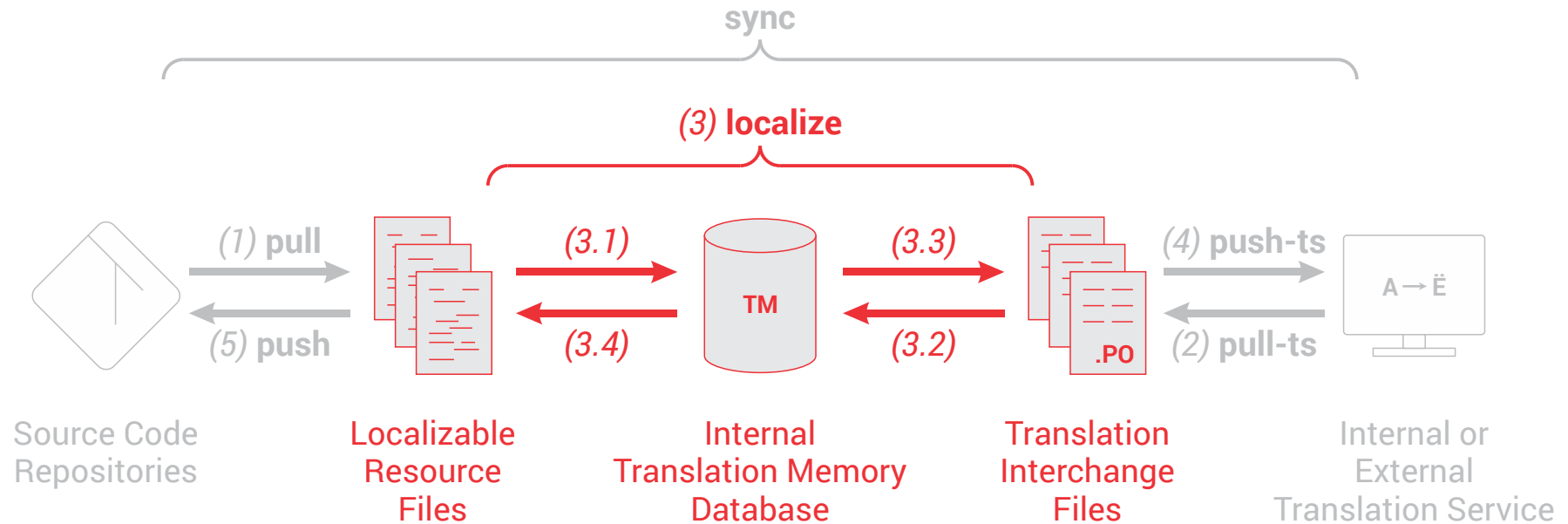
Alternative: publish .po files for
offline translation,
integrate with vendor's API


```
sync
{
  ts
  {
    plugin                pootle
    data
    {
      manage_py_path      /var/pootle/pootle/manage.py
      project_id           android_evernote
    }
  }
}

vcs
{
  # ...
}

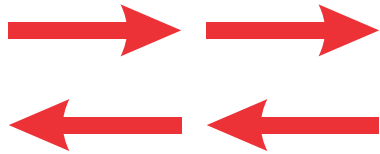
jobs
{
  # ...
}
```

```
$ serge localize myproject.serge
```



Core Functionality

localize

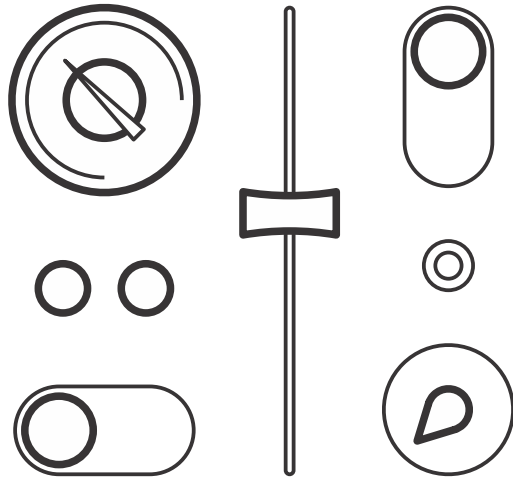


Parsing and generating resource files

Generating and parsing translation interchange files (.PO)

Many built-in configuration options (what and how to process, how to use TM)

Advanced control plugins



Configuration Options

Configuration: General Options

```
# job definition
{
  id                evernote.windows.rc
  source_language   en
  destination_languages de es es-419 ja ru zh-cn
  #...
}
```

```
$ serge localize --job=evernote.windows.rc windows.serge
$ serge localize --lang=ru,zh-cn windows.serge
```

Configuration: DB Options

```
# job definition
{
  #...

  db_source           DBI:SQLite:dbname=/var/serge/db/translate.db3
  db_username         serge # not for SQLite
  db_password         %ENV:SERGE_DB_PASSWORD% # not for SQLite

  db_namespace       windows_evernote

  #...
}
```

Supported drivers: SQLite, MySQL, Postgres

Multiple databases = Multiple TMs

Configuration: Source Options

```
# job definition
{
  #...

  source_dir          /var/serge/data/evernote_windows/app/res
  source_process_subdirs YES
  source_match        \.rc$
  source_exclude      ^~

  #...
}
```

Extensive use of regular expressions

Configuration: Parsing Options

```
# job definition
{
  #...

  parser
  {
    plugin                parse_rc
    data
    {
      # plugin-specific data
    }
  }

  normalize_strings      NO
  #...
}
```


Configuration: Output Options

```
# job definition
{
    #...

    ts_file_path                /var/serge/po/evernote_windows/%LOCALE%/%FILE%.po
    output_encoding              UCS-2LE
    output_bom                   YES
    output_file_path             /var/serge/data/evernote_windows/app/res/%FILE%.%LANG%

    output_lang_rewrite
    {
        no                       nb
        zh-cn                     zh-Hans
    }

    output_only_mode             NO
    #...
}
```

Configuration: Example

```
source_dir           /var/serge/data/evernote_windows/app/res
source_process_subdirs YES
source_match        \.rc$

ts_file_path        /var/serge/po/evernote_windows/%LOCALE%/%FILE%.po
output_file_path    /var/serge/data/evernote_windows/app/res/%FILE%.%LANG%
```

```
source path        → /var/serge/data/evernote_windows/app/res/path/to/file.rc
```

```
relative path     → path/to/file.rc → %FILE%, %PATH%, %NAME%, %EXT%, ...
```

For each language (e.g. 'pt-br' → %LANG%, %LOCALE%, ...):

```
ts_file_path      → /var/serge/po/evernote_windows/pt_BR/app/path/to/file.rc.po
```

```
output_file_path  → /var/serge/data/evernote_windows/app/res/path/to/file.rc.pt-br
```

Configuration: TM Reuse Options

```
# job definition
{
  #...

  reuse_translations      YES
  reuse_orphaned         YES
  reuse_uncertain        NO
  reuse_as_fuzzy_default NO
  reuse_as_fuzzy         ja
  reuse_as_not_fuzzy     de zh-cn

  #...
}
```

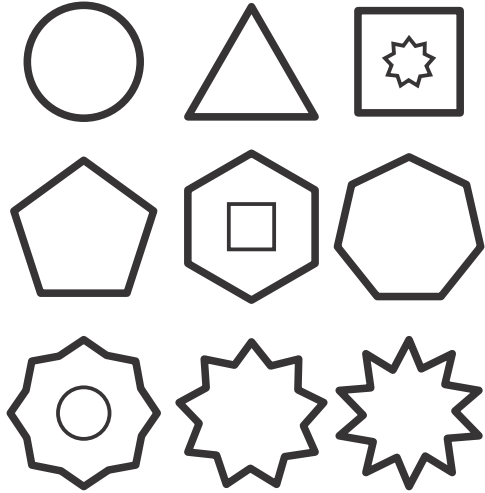
Configuration: TM Reuse Options

```
# job definition
{
  #...

  similar_languages
  {
    {
      source          es
      destination     es-419 # Spanish (Latin America)
      as_fuzzy        YES
    }
    {
      source          es-419
      destination     es
      as_fuzzy        YES
    }
  }
  #...
}
```

Configuration: Callback Plugins

```
# job definition
{
  #...
  callback_plugins
  {
    {
      plugin          transform
    }
    {
      plugin          completeness
      data
      {
        create_threshold 0.95
      }
    }
  }
  #...
}
```



File Parsers

20+ Parsers Out of the Box

Android .strings, Chrome Extension .json files, .DTD,
Perl/PHP/Ruby hashes, JavaScript (dictionary objects),
generic JSON files, INI-like key-value files, **PHP/XHTML**,
iOS .plist files, .pot files, Java .properties, Windows .rc files,
.Net framework .resx files, Blackberry .rrc files, MacOS/iOS
.strings files, QT Linguist .ts files, WiX installer .wxi files,
XLIFF 1.x/2.x files, **XML files** (generic, Android-specific,
InDesign-specific), Generic YAML files,
.master files (arbitrary plaintext markup),
generic regex-based parser

PHP/XHTML Parser

Static localization of (X)HTML pages with optional PHP snippets, including embedded localizable PHP and JavaScript strings inside wrapper `_(' . . . ')` functions.

Full control of segmentation. By default, only the content of certain tags is extracted, but one can explicitly include or exclude certain nodes.

Validating parser: you can get email alerts each time the file fails to parse.

PHP/XHTML Parser

```
<p data-l10n-context="context" data-l10n-hint="hint">string</p>
```

```
<h1>string</h1>
```

```
<p>string</p>
```

```
<p lang="">string</p>
```

```

```

```
<p>
```

```
  outer string
```

```
  <span lang="">inner string</span>
```

```
</p>
```

```
<p>
```

```
  <span lang="en">outer string</span>
```

```
  <span lang="">inner string</span>
```

```
</p>
```

```
<p>
```

```
  Click here: <a href="http://sample.com">http://sample.com</a>
```

```
</p>
```

PHP/XHTML Parser

```
<p>  
  <span Lang="en">Click here:</span>  
  <a href="http://sample.com">http://sample.com</a>  
</p>
```

```
<input type="search" placeholder="string">  
<input type="text" value="string">
```

```
<div>string</div>  
<div Lang="en">string</div>
```

```
<?php  
  echo "string";  
  echo _("string");  
  echo __('string');  
  echo ___('string');  
?>
```

```
<script type="text/javascript">  
  alert("string");  
  alert(___('string'));  
</script>
```

PHP/XHTML Parser

```
# job definition
{
  # ...
  parser
  {
    plugin          parse_php_xhtml
    data
    {
      expand_entities    NO
      validate_output    YES
      email_from         110n-robot@acme.org
      email_to           website-engineer@acme.org
                       website-project-manager@acme.org
    }
  }
}
```

XML Parser

Localize any arbitrary nodes whose paths match provided regular expressions.

Specified nodes can be treated as containing HTML, and are processed by a nested PHP+XHTML parser.

Support Android-specific and InDesign-specific data handling rules

XML Parser

```
<products>
  <description>Product list</description>
  <items>
    <item sku="P001">
      <price>1.23</price>
      <title>First Product</title>
      <description><![CDATA[
        <p>First Product Description</p>
        <p>Another paragraph</p>
      ]]></description>
    </item>
    <item sku="P002">
      <price>2.34</price>
      <title>Second Product</title>
      <description><![CDATA[
        <p>Second Product Description</p>
      ]]></description>
    </item>
  </items>
</products>
```

XML Parser

```
# job definition
{
  # ...
  parser
  {
    plugin                parse_xml
    data
    {
      node_match          \/(title|description)$
      node_exclude        ^products\/description$
      node_html           \/description$

      email_from          110n-robot@acme.org

      email_to             web-engineer@acme.org
                          web-project-manager@acme.org
    }
  }
}
```

.master Parser

You can mark up (segment) any arbitrary documents: plain-text emails, glossaries, other documents that can't be otherwise easily parsed, and generate both source and localized files from the "master" file.

Ability to provide context keys, comments and special flags (e.g. pad the resulting value with spaces).

.master Parser

plain_email.en.txt

```
To: ${name} <${email}>  
From: ${from}  
Subject: ACME account activation
```

This email confirms your recent ACME account activation request.

Follow the link below, and we will activate your account with username \${username}:

```
http://${url}/activate?username=${username}&secret=${secret}
```

- The ACME team

.master Parser

plain_email.en.txt.master

To: \${name} <\${email}>

From: \${from}

Subject: <%ACME account activation%%Email Subject%>

<%This email confirms your recent ACME account activation request.%>

<%Follow the link below, and we will activate your account with username \${username}:%>

[http://\\${url}/activate?username=\\${username}&secret=\\${secret}](http://${url}/activate?username=${username}&secret=${secret})

- <%The ACME team%>

.master Parser

```
# job definition
{
  # ...

  parser
  {
    plugin          parse_master
  }
}
```

Regex-based Metaparser

Support your custom localization file formats by providing a set of regular expression-based extraction rules.

Extract keys, values, key-value pairs, multi-line hints (translation comments).

Escaping/unescaping rules.

Regex-based Metaparser

sample.ini

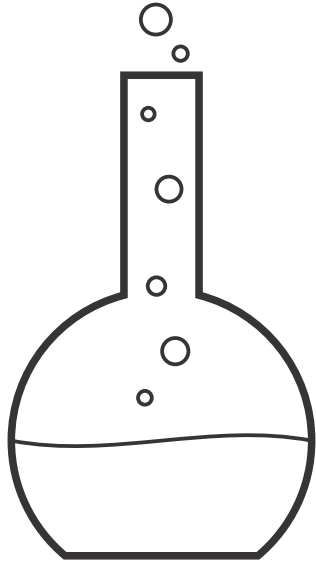
```
; global comment
# another global comment

[section]
; localization note
foo=Foo
bar = Bar
; localization note 2 line 1
; localization note 2 line 2
baz= Baz

[subsection]
# comment
etc =Etc\nEtc2\nEtc3
```

Regex-based Metaparser

```
# job definition
{
  # ...
  parser
  {
    plugin          metaparser
    data
    {
      /*
      Parse .ini files
      */
      hint          ^\s*[\;#]\s*(.*)\s*$           # ; foo or # foo
      keyvalue      ^(\S+)\s*=\s*(.*)\s*$         # foo = bar
      localize      ^(\S+\s*=\s*)(.*)(\s*)$
      reset         ^\s*$                          # blank line
    }
  }
}
```



Advanced Control Plugins

text.your
questions: 650 443 85 26

process_if Plugin

Do not process certain source files
(**can_process_source_file** callback phase)

Do not extract certain strings for translation
(**can_extract** callback phase)

Do not generate/save certain localized files
(**can_generate_localized_file** callback phase)

Criteria: file path, file content, string content, string comment,
target language, previously set flag

process_if recipe: translate only files that have special “LOCALIZABLE” flag

```
# job definition
{
  # ...

  callback_plugins
  {
    {
      plugin
      phase
      data
      {
        if
        {
          content_matches
          then
          {
            return
            }
          }
        }
      }
    }
  }
}
```

process_if
can_process_source_file

\bLOCALIZABLE\b

YES

process_if recipe: do not translate a string if it starts with “!!”

```
# job definition
```

```
{
```

```
  # ...
```

```
  callback_plugins
```

```
  {
```

```
    {
```

```
      plugin  
      phase
```

```
      process_if  
      can_extract
```

```
      data
```

```
      {
```

```
        if
```

```
        {
```

```
          content_doesnt_match
```

```
          ^!!
```

```
          then
```

```
          {
```

```
            return
```

```
            YES
```

```
          }
```

```
        }
```

```
      }
```

```
    }
```

```
  }
```

```
}
```

replace_strings Plugin

Rewrite source file content before parsing
(**after_load_file** callback phase)

Rewrite localized file content before saving
(**before_save_localized_file** callback phase)

Rewrite translation on the fly
(**rewrite_translation** callback phase)

Rewrite input/output file paths
(**rewrite_path** callback phase)

replace_strings recipe: statically rewrite a variable

strings.js → var lang = "en";
strings.de.js → var lang = "de";
strings.pt-br.js → var lang = "pt-br";

```
# job definition
{
  # ...
  callback_plugins
  {
    {
      plugin          process_if
      phase           before_save_localized_file

      data
      {
        #             replace what      replace with
        replace       `lang = "en"`     `lang = "%LANG%"`
      }
    }
  }
}
```

limit_languages Plugin

Gives an ability to specify the list of target languages on a per-file basis.

By default, each file is localized into all languages specified in the job's **destination_languages** parameter.

Setting the list of languages (default rule):

```
L10N_LIMIT_DESTINATION_LANGUAGES=es,de,pt-br
```

Excluding languages (default rule):

```
L10N_EXCLUDE_DESTINATION_LANGUAGES=ja,zh-cn
```

limit_languages recipe: specify languages by regions

default → empty list of languages

LANG_EUROPE → add: da de es fi fr it nl
pl pt ru sv tr

LANG_APAC → add: id ja ko zh-cn zh-tw

LANG_ALL → add all languages

global.js → // LANG_ALL

feature_x.js → // LANG_EUROPEAN LANG_APAC

append_hint_message Plugin

Add arbitrary comments/link to translation units

Russian → Website → website2_evernote/careers/index.php.po → Unit #2445579

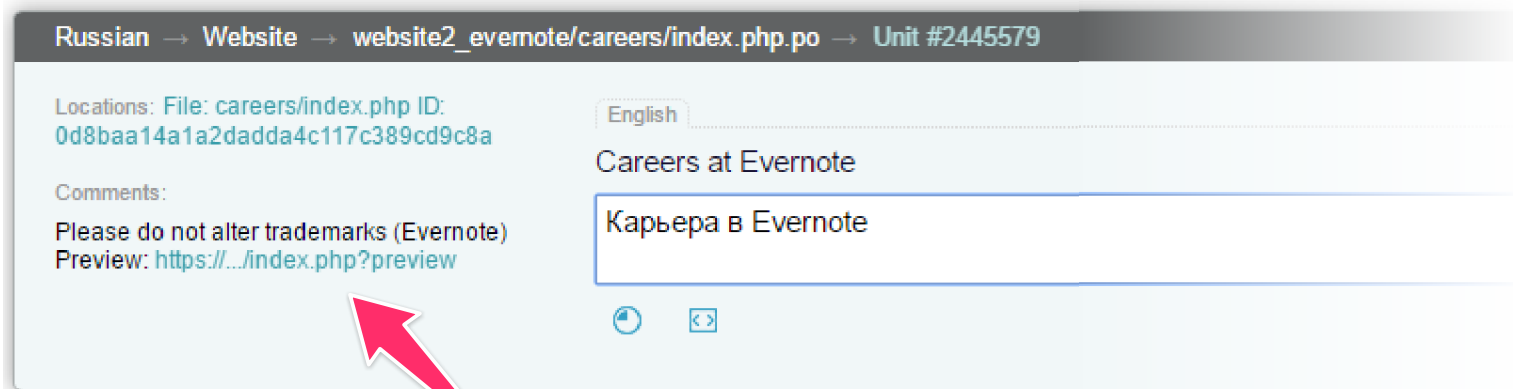
Locations: File: careers/index.php ID: 0d8baa14a1a2dadda4c117c389cd9c8a

Comments:
Please do not alter trademarks (Evernote)
Preview: <https://.../index.php?preview>

English

Careers at Evernote

Карьера в Evernote

The screenshot shows a web-based translation interface. At the top, a breadcrumb trail reads 'Russian → Website → website2_evernote/careers/index.php.po → Unit #2445579'. Below this, the 'Locations' section shows the file path 'careers/index.php' and a unique ID. The 'Comments' section contains the text 'Please do not alter trademarks (Evernote)' and a preview link. To the right, a language selector is set to 'English', and the source text 'Careers at Evernote' is shown above a text input field containing the Russian translation 'Карьера в Evernote'. At the bottom of the input field are icons for undo and redo. A large red arrow points from the bottom left towards the 'Comments' section.

limit_languages recipe: website preview links

```
# job definition
{
  # ...
  callback_plugins
  {
    {
      plugin          append_hint_message
      data
      {
        message       Preview (English):
                      https://preview-site/en/%FILE%
        message       Preview (localized):
                      https://preview-site/%LANG%/%FILE%
      }
    }
  }
}
```

feature_branch Plugin

Optimize localization of multiple branches by translating only deltas.



feature_branch example

```
sync
{
  vcs
  {
    plugin          git
    data
    {
      local_path    /var/data/serge/android
      remote_path
      {
        master      ssh://git.evernote.com/myapp
        feature/a    ssh://git.evernote.com/myapp#feature/a
        feature/b    ssh://git.evernote.com/myapp#feature/b
        # ...
      }
    }
  }
}
# ...
}
```

```

jobs
{
  :master
  {
    id                android.master
    # ...

    callback_plugins
    {
      {
        plugin        feature_branch
        data
        {
          master_job  android.master
        }
      }
    }
  }
}

:feature/a
{
  @inherit           .#jobs/:master
  id                android.feature.a
  # ...
}
}

```

transform Plugin

“Guess” translations by using existing ones and applying known transformations (case, whitespace, punctuation, wrapper tags).

Hello, world! → Привет, мир!

HELLO, WORLD → ???

Hello, world! → HELLO, WORLD

original string → upper case → trailing punctuation

Привет, мир! → ПРИВЕТ, МИР

test_language Plugin

Generate pseudolocalized files on the fly,
with optional string expansion and
character replacement.

This is an example

ᚦhĩš ĩš áη ēxámplē

[ᚦhĩš ĩš áη ēxámplē xxxxx xx]

completeness Plugin

Create, update or delete localized resource files only when they are translated above or below the certain threshold.

```
# job definition
{
  # ...

  callback_plugins
  {

```

```
    {
      plugin          completeness
      data
      {
        create_threshold 0.9
        update_threshold 0.3
        can_delete       YES
      }
    }

```

serge.io

Documentation, download and installation instructions

github.com/evernote/serge

t/ folder contains additional examples:
test configuration files for different parsers

text.your
questions: 650 443 85 26

Questions?

text your
questions: **650 443 85 26**

Anytime (IRC, GitHub):

serge.io/contact

linkedin.com/in/iafan