**Документация, содержащая описание функциональных характеристик программного обеспечения «PROMT Translation Server Developer Edition (for Linux)» и информацию, необходимую для установки и эксплуатации программного обеспечения**

# Руководство администратора программного обеспечения «PROMT Translation Server Developer Edition (for Linux)»

Россия, 199155,

Санкт-Петербург, Уральская ул., д. 17, лит. Е, кор. 3, пом. 15Н.

E-mail: common@promt.ru

support@promt.ru

Internet: https://www.promt.ru

https://www.translate.ru

Телефон: +7 812 655-0350

Факс: +7 812 655-0021

# Оглавление

# Введение

## Об этом документе

Данное руководство предназначено для «PROMT Translation Server Developer Edition (for Linux)» (далее – PTS), программного обеспечения для машинного перевода для ОС Linux (перечень совместимых ОС указан в разделе «Требования к программному обеспечению»).

Данный документ предназначен для администратора PTS и содержит описание основных функциональных характеристик PTS, а также информацию, необходимую для установки и эксплуатации PTS. В описание включены также архитектура PTS, описание интерфейса командной строки приложений PTS и обзор конфигурационных файлов.

Web сервис PTS описан в отдельном документе «Web service SDK» предоставляемом отдельно разработчикам.

## Термины и сокращения

| Термин | Описание |
|---|---|
| Языковая пара | Определяет, с какого языка и на какой язык будет переведен текст. В некоторых случаях (имена параметров и т.д.) может использоваться термин "направление перевода" или просто "направление" |
| Профиль перевода | Набор лингвистических настроек, которые модуль перевода использует для повышения качества перевода в конкретной тематической области. В некоторых случаях может использоваться термин "шаблон", или "тематика", или "шаблон тематики" |
| Генеральный словарь | Основной словарь для каждой языковой пары. Этот словарь содержит общую лексику и не может быть отредактирован пользователем. Модуль перевода всегда использует генеральный словарь независимо от настроек профиля перевода. |
| Специализированный словарь | Словарь, который расширяет генеральный словарь в определенной области. Пользователь не может редактировать специализированные словари. |
| Пользовательский словарь | Словарь, который пользователь может создавать и редактировать. Пользовательский словарь состоит из новых слов или фраз, а также измененных статей от генерального, специализированного или других пользовательских словарей. |
| Память переводов (TM) | Структура данных, которая хранит сегменты, состоящие из исходного сегмента (обычно, предложение) и сегмента перевода |
| RBMT модули | Модули перевода машинного перевода, основанного на правилах перевода (Rule Based Machine Translation) |
| Статистическая модель (SMT модель) | Набор лингвистических данных, необходимых для выполнения статистического машинного перевода (SMT). Модель строится (тренируется) на корпусе параллельных текстов |

| | |
|---|---|
| Нейронная модель (NMT модель) | Набор лингвистических данных, необходимых для выполнения нейронного машинного перевода (NMT). Модель строится (тренируется) на корпусе параллельных текстов |

# Обзор основных компонентов

PTS - набор программных модулей, позволяющих получить машинный перевод на Unix-подобных операционных системах в среде интранет. PTS предоставляет HTTP(S) веб-сервис с простым REST API, а также пользовательский веб-интерфейс. Кроме удаленного доступа по HTTP протоколу, клиенты могут использовать локальный API для создания утилит командной строки (примером такой утилиты, входящей в поставку PTS, служит **promttrans**). PTS поддерживает горизонтальное масштабирование за счет возможности распределения нагрузки на другие сервера перевода.

С точки зрения администратора, PTS содержит следующие основные компоненты:

- Лингвистические данные
- Ядро перевода
- Балансировщик нагрузки
- Службы (демоны)
- Веб-интерфейс/веб-сервис
- Приложения командной строки

### Лингвистические данные

Лингвистические данные - это набор словарей, баз ТМ, слов, которые не требуется переводить, и других настроек, влияющих на результат перевода. Словари хранятся в виде файлов, в формате, совместимом с версией PTS для Windows. Базы ТМ используют формат открытой системы управления базой данных - SQLite. Остальные настройки хранятся в текстовом виде в конфигурационных файлах. Все эти данные должны располагаться на одной машине с ядром перевода.

### Ядро перевода

Главный компонент в архитектуре PTS - это ядро перевода. Оно предоставляет API для перевода и управления лингвистическими данными. Все модули ядра являются совместно используемыми библиотеками.

При загрузке, ядро перевода читает конфигурационный файл и определяет доступные лингвистические модули и данные. Само ядро работает как диспетчер для передачи запросов между модулями. Вся обработка делается модулями.

После получения запроса, ядро создает и инициализирует объект, который содержит информацию о запросе:

- тип запроса
- направление обработки запроса
- параметры запроса
- результат запроса

Ядро загружает лингвистические модули (и, таким образом, создает конвейер для обработки запросов) в порядке, определенном конфигурацией. Запрос проходит через конвейер в обоих направлениях, сначала вперед (от первого модуля до последнего), и затем назад. Каждый модуль получает объект запроса и анализирует его атрибуты.

### Балансировщик нагрузки

Балансировщик нагрузки представляет собой промежуточный модуль между веб-сервером и ядром перевода. Он служит для распределения нагрузки между серверами перевода PTS, разрешая таким образом проблему горизонтальной масштабируемости на уровне отдельных запросов веб-сервиса. Список серверов перевода задается в конфигурационном файле и считывается в момент запуска балансировщика. Балансировщик нагрузки используется как единая точка входа для веб-сервиса PTS, в то время как серверов перевода может быть более одного.

### Службы (демоны)

В состав PTS входят следующие процессы, запускаемые в режиме демонов:

- **nginx (служба promt-nginx)** – автономная версия Nginx, которая устанавливается вместе с PTS и не имеет системных зависимостей. Служба может конфликтовать с установленным системным веб-

сервером (из-за использования одного TCP порта), поэтому рекомендуется отключать или удалять системную службу перед установкой PTS.

- **transfcgid.run (служба promt-balancer, режим балансировки)** – серверный процесс, который взаимодействует с веб-сервером (nginx) по протоколу HTTP в качестве прокси сервера и перенаправляет запросы на другие сервера перевода. Балансировщик решает следующие задачи:
  - Балансирует нагрузку между серверами перевода
  - Реализует простую схему масштабируемости с возможностью наращивания мощностей перевода
  - Используется в качестве единой точки входа веб-сервиса (сервис доступен для вызова извне для клиентов PTS)
- **transfcgid.run (служба promt-translator , режим перевода)** – серверный процесс, который взаимодействует с веб-сервером (nginx) по протоколу FastCGI. Сервер перевода решает следующие задачи:
  - Реализует методы веб-сервиса (клиенты PTS не могут вызывать его напрямую, это может делать только балансировщик нагрузки)
  - Запускает дочерние процессы, которые выполняют перевод
  - Балансирует нагрузку между дочерними процессами
  - Реализует простую схему отказоустойчивости, когда дочерний процесс завершается с ошибкой или зависает

  Мастер-процесс запускается пользователем, дочерние процессы управляются мастер-процессом
- **dcs.run (служба promt-dcs)** - служба управления данными, обеспечивает доступ к Translation Memory. **dcs** использует базу данных SQLite для хранения сегментов Translation Memory. Запускается пользователем.
- **Promt.Host.exe (служба promt-managed) -** серверный мастер-процесс для организации SMT/NMT перевода и решения некоторых вспомогательных задач. Процесс запускается с помощью mono. **Promt.Host.exe** управляет дочерними процессами - **Promt.Unit.exe**, таким образом реализуется отказоустойчивость. Мастер-процесс запускается пользователем, дочерние процессы управляются мастер-процессом

### Web-интерфейс

Веб-интерфейс PTS состоит из нескольких HTML страниц, которые демонстрируют работу основных методов веб-сервиса PTS: перевод текста, перевод документов и перевод Web страниц.

### Приложения

PTS включает консольное приложение (**promttrans.run**), которое предназначено для демонстрации функций ядра перевода.

### Обработка запроса

Типичная схема обработки запроса на перевод:

1. Клиент отправляет запрос на перевод текста, используя для передачи HTTP-протокол.
2. Web сервер (Nginx) получает запрос и определяет, что запрос требуется передать для обработки процессу `transfcgid` по HTTP протоколу для дальнейшей балансировки между серверами перевода.
3. Процесс `transfcgid` находит наименее загруженный сервер перевода и пересылает запрос ему (этим сервером может быть в том числе и тот, на котором запущен балансировщик).
4. Web сервер (Nginx) получает запрос и определяет, что запрос требуется передать для обработки процессу `transfcgid` по FastCGI-протоколу для дальнейшего обработки ядром перевода.
5. Процесс `transfcgid` находит наименее загруженный дочерний процесс TransFcgid и пересылает запрос ему.
6. Дочерний процесс `transfcgid` осуществляет перевод текста. В процессе перевода, в зависимости от настроек, может возникнуть необходимость использовать Translation Memory. В этом случае дочерний процесс `transfcgid` обращается к процессу `dcs`.
7. Если перевод использует SMT или NMT движок, то дочерний процесс `transfcgid` обращается за переводом к процессу `Promt.Host.exe`.
8. `Promt.Host.exe` определяет дочерний процесс, в который загружена требуемая модель, и вызывает процесс `Promt.Unit.exe`.

# Системные требования

## Требования к аппаратным средствам

Минимальные системные требования к набору **без SMT**:

1. Dual-core CPU
2. 4 GB RAM
3. 500 Мбайт свободного пространства (лингвистические данные пользователя могут занять дополнительное место)

Минимальные системные требования к набору **с SMT**:

4. Quad-core CPU
5. RAM на каждое SMT/NMT направление перевода, GB (диапазон значений связан с тем, что память может увеличиваться в процессе перевода):
    - 3-10 GB для SMT моделей
    - 1-5GB для NMT моделей
6. Место на диске на каждое SMT направление перевода, GB:
    a) 10-50 GB для SMT моделей
    b) 1-3GB для NMT моделей

## Требования к программному обеспечению

PTS устанавливается с помощью собственного инсталлятора и был протестирован со следующими ОС:

- CentOS 8
- Ubuntu 18.04.4
- Ubuntu 19.10
- Ubuntu 20.04
- Debian 10
- AstraLinux CE 2.12
- AstraLinux SE 1.6

Какой-либо жесткой привязки PTS к определенному дистрибутиву ОС не существует, вместо этого используются маркеры совместимости. На текущий момент это:

- Наличие системной библиотеки GLIBC версии 2.17 или выше
- Наличие системных библиотек libgcc_s.so.1 и libstdc++.so.6
- Наличие менеджера системных служб systemctl

Программное обеспечение Nginx и mono, которое используется при работе PTS входит в состав дистрибутива и требует дополнительной установки пакетов.

# Установка PTS

В общем случае для получения работающего сервера PTS требуется только запустить инсталлятор и следовать инструкциям. Для работы PTS требуется сгенерированный для данного компьютера файл лицензии. Для получения файла лицензии выполните следующее:

1. Получите уникальный идентификатор компьютера `HardwareId`.
2. Передайте полученный `HardwareId` в службу поддержки компании ПРОМТ для получения файла лицензии.
3. Установите набор, указав путь к полученному файлу лицензии.

# Получение идентификатора компьютера

Для получения идентификатора компьютера до установки набора можно воспользоваться следующей командой (предполагается, что команда запускается из каталога с run-файлом):

```
chmod +x PTS3.9.run && ./PTS3.9.run -i
```

При этом в консоль будет выведено сообщение вида:

Current hardware id: Y27wF82sEHHo25v7DUMqUMttqRhmALqzXt99qke7bZk=

Текст после "Current hardware id:" (идентификатор, закодированный в base64-строку) необходимо скопировать и переслать службе поддержки компании ПРОМТ.

# Установка основного набора

PTS распространяется в виде файла с расширением .run, который представляет собой модифицированный 7z SFX архив с основным инсталлятором. Запуск run-файла осуществляется с помощью команды:

```
chmod +x PTS3.9.run && sudo ./PTS3.9.run
```

Инсталлятор автоматически найдет файл лицензии, если он находится в одной папке с инсталлятором. Файл лицензии можно также указать в процессе установки, либо передать с помощью ключа:

```
chmod +x PTS3.9.run && sudo ./PTS3.9.run -k file.lic
```

Работа PTS без файла лицензии невозможна. Для установки лицензий в уже установленный набор используется скрипт `update-license.sh`. Пример запуска:

```
sudo /usr/local/promt/bin64/update-license.sh file.lic
```

В процессе установки инсталлятор выполняет следующие действия:

1. Проверка параметров системы. Система будет проверена на совместимость с продуктом.
2. Распаковка данных. Программные модули и лингвистические данные будут распакованы в корневую папку продукта. Настройки будут модифицированы под текущее окружение ОС.
3. Установка файла лицензии. Файл будет скопирован из указанного места в корневую папку продукта.
4. Настройка фаервола. Инсталлятор попытается найти фаервол и разрешить входящие соединения на порт 80.
5. Создание и запуск системных служб PTS.

Во время установки инсталлятор может задать следующие вопросы:

- Подтверждение принятия лицензии (y/n)
- Подтверждение пути установки (ввод значения или "enter" для значения по умолчанию)
- Подтверждение пути файла лицензии (ввод значения или "enter" для принятия найденного файла лицензии)
- Подтверждение создания нового пользователя (ввод значения или "enter" для значения по умолчанию)

На системах с активной защитой SELinux (например, семейство RHEL) необходимо убедиться, что указанная корневая папка продукта принадлежит директории /usr/. Рекомендуется использовать путь по умолчанию.

# Ключи командной строки

Для установки в неинтерактивном режиме существует способ запуска инсталлятора с ключом "-y", в этом случае будут использованы значения по умолчанию (или заданные с помощью ключей). Список других ключей инсталлятора:

- -h, --help: вывод справки
- -i, --id: вывод идентификатора оборудования (hardware id), который используется для генерации файла лицензии.
- -e, --extract [PATH]: распаковка содержимого архива в указанную директорию без запуска скрипта установки и без последующего удаления. Создает директорию, если ее не существует.
- -y, --yes: ответ на все запросы в автоматическом режиме. Для диалогов «y/n» - будет выбрано «y», для диалогов с вводом – значение по умолчанию, если это возможно. Передается в основной инсталлятор.
- -k, --key [PATH]: путь файла лицензии PTS. Передается в основной инсталлятор.
- -p, --path [PATH]: путь установки PTS. Передается в основной инсталлятор.
- -m, --model [PATH]: путь до архива SMT/NMT модели. Используется для установки SMT/NMT аддонов, тип модели определяется автоматически. Передается в основной инсталлятор.

# Тестирование работоспособности

Для проверки работы веб-интерфейса, запустите браузер и введите следующий адрес:

```
http://localhost/pts
```

Для проверки работы веб-сервиса, вы можете выполнить следующую команду:

```
curl 'http://localhost/pts/service/TranslateText?from=en&to=es&text=book' && echo
```

**Примечание**: По-умолчанию PTS устанавливается с доступом только по HTTP протоколу. Доступ по HTTPS протоколу требует дополнительной настройки сервера Nginx и не описан в данном документе.

# Включение логирования

Логирование по умолчанию выключено. Для включения логирования необходимо изменить конфигурационный файл *promtkernel.conf* (расположение по умолчанию - */usr/local/promt/promtkernel.conf*). Найдите раздел *General,* параметр *Modules* и вставьте строку "log," в начало списка модулей. Файлы с логами будут создаваться в каталоге */usr/local/promt/log*. Для получения дополнительной информации, см. раздел "Конфигурационный файл".

**Примечание**: Обратите внимание, что логирование требует много места на диске и может повлиять на производительность системы.

В логах сохраняется содержимое файлов (в том числе HTML при переводе сайтов), тип операции перевода (text, file, url) и URL страниц (в случае перевода сайтов). Пример содержимого файла логов:

```
<request>
<status>
success
</status>
<starttime>	Thu Sep  7 10:06:03 2017	</starttime>
<processingtime>	1.01342</processingtime>
<endtime>	Thu Sep  7 10:06:04 2017	</endtime>
<TRANS_TYPE>
```

*url*
*</TRANS_TYPE>*
*<TRANS_URL>*
*statmt.org*
*</TRANS_URL>*
*<DIRECTION>*
*er*
*</DIRECTION>*
*<TEMPLATE>*
*Universal*
*</TEMPLATE>*
*<SOURCE>*
*PGh0bWw+PGhlYWQ…*
*</SOURCE>*
*<RESULT>*
*PGh0bWw+PGhlYWQ…*
*</RESULT>*
*<UNKNOWNWORDS>*
*ELRA*
*</UNKNOWNWORDS>*
*</request>*

**Примечание**: Содержимое файлов сохраняется в виде Base64 строки.

## Поддержка логирования для сервиса promt-managed

В `Promt.Host.exe` и `Promt.Unit.exe` добавлена поддержка логирования. Для включения логирования необходимо изменить файл сервиса *(/etc/systemd/system/promt-managed.service*) и добавить ключ «-v» в строку

```
ExecStart=/usr/local/promt/mono/mono usr/local/promt/bin64/Promt.Host.exe -v
```

Выполните команды:

```
sudo systemctl daemon-reload
sudo systemctl restart promt-managed
```

Лог файлы будут доступны в каталоге: */usr/local/promt/log*

Для `Promt.Host.exe` - *managed-host.log*

Для создаваемых им юнитов (`Promt.Unit.exe`) – *managed-%model%_unit.log*, где *%model%* имя загруженной в юнит модели.

## Улучшение логирования для promt-translator/promt-balancer

В `transfcgid` увеличено количество логируемых мест и изменен режим записи в файл лога – теперь новые записи добавляются к существующему файлу, а не перезаписывают его.

Для активации режима логирования на уровне сервисов, необходимо добавить ключ "–v" в аргументы командной строки при запуске `transfcgid` в файлах сервисов `promt-translator` и `promt-balancer` (см. параграф про логирование в *promt-managed*).

В каталоге */usr/local/promt/log* будут созданы:

*translator.trace* – для *promt-translator*
*balancer.trace* – для *promt-balancer*

## Включение сбора статистики

Сбор статистики переводов по умолчанию выключен. Для включения статистики, необходимо изменить конфигурационный файл *promtkernel.conf* (расположение по умолчанию - */usr/local/promt/promtkernel.conf*). Найдите раздел *General,* параметр *StatLevel* и измените его значение на цифру *1*. Файл статистики называется *"stat.db"* и представляет собой базу данных формата SQLite3. Файл расположен по пути *[DataPath]/stat.db*, где *[DataPath]* – это папка данных продукта (значение можно посмотреть в конфигурационном файле, обычно это */home/promt*). Для получения дополнительной информации, см. раздел "Конфигурационный файл".

## Сбор отзывов пользователей

В веб-сервисе существует метод, позволяющий получать и накапливать отзывы (feedback) от пользователей. Пример запроса из браузера:

```
http://localhost/pts/service/addfeedback?title=nice&text=good
```

Содержимое отзывов записывается в файл "[DataPath]/feedback.xml", где *[DataPath]* – это папка данных продукта (значение можно посмотреть в конфигурационном файле, обычно это /home/promt). Пример содержимого файла:

```
<feedback>
<time>
Fri Sep  8 11:21:36 2017
</time>
<title>
nice
</title>
<text>
good
</text>
</feedback>
```

**Примечание:** В случае конфигурации с несколькими серверами перевода данные отзывов будут находиться на различных серверах.

# Установка аддонов SMT/NMT перевода

Перевода на базе движков SMT/NMT требует значительное место на диске и использует большой объем оперативной памяти, поэтому распространяются в виде отдельных инсталляционных пакетов. Пакет представляет собой исполняемый файл run-файл (по аналогии с основным набором) и отдельный файл архива модели. Пример запуска установки модели:

```
chmod +x PTS_snmt.run && sudo ./PTS_snmt.run –m model.zip –k file.lic
```

В зависимости от того, какой именно аддон устанавливался, после завершения установки произойдет следующее:

- При установке SMT аддона появится новое направление, соответствующее архиву модели.
- При установке NMT аддона универсальный профиль перевода будет модифицирован таким образом, чтобы перевод осуществлялся с использованием технологии NMT.

# Удаление PTS

Удаление продукта осуществляется посредством запуска скрипта "uninstall.sh" из папки "bin64" в корневой папке продукта. При запуске скрипт проверит, запущен ли он с правами суперпользователя, а также спросит явное разрешение на удаление продукта у пользователя. Скрипт полностью удаляет PTS из системы, в том числе: системные сервисы и корневую папку продукта.

# Веб-интерфейс

Веб-интерфейс состоит из трех страниц: **Перевод текста, перевод документа и перевод веб-страницы**. Все страницы вызывают соответствующие методы веб-сервиса, и не требуют никакого серверного кода (такого как PHP). Вы можете использовать меню **Язык интерфейса**, чтобы изменить язык пользовательского интерфейса.

## Перевод текста

Страница предназначена для перевода простого текста. Выберите входной и выходной языки. Если Вы не знаете точно, на каком языке исходный текст, выберите **Определить язык**. Точность автоматического определения языка зависит от количества текста. Введите исходный текст в поле **Исходный текст** - перевод сразу появится в поле **Перевод**. Вы можете указать профиль перевода для настройки перевода в конкретной области или тематике.

## Перевод документов

Страница предназначена для перевода файлов в различных форматах. Укажите формат своего документа, выберите файл с диска и нажмите **Перевести**. Файл с результатом перевода будет автоматически загружен на ваш компьютер.

## Перевод веб-страницы

Чтобы перевести веб-страницу, укажите ее адрес (например, www.someaddress.com) и нажмите **Перевести**. Переведенная страница будет открыта в новой вкладке. Обратите внимание, что переходы по ссылкам на переведенной странице также ведут на переведенные страницы.

# Командная строка

## Управление службами

По умолчанию при установке служба PTS автоматически запускается и настраивается на запуск при загрузке системы, однако в ряде случаев может возникнуть необходимость выполнения ручных операций (например, перезапуска службы). Служба PTS реализована в виде нескольких сервисов (имена соответствуют именам файлов описания, расположенных в `/etc/systemd/system`):

- *promt-nginx.service* — запускает nginx, который входит в состав PTS.
- *promt-balancer.service* — запускает transfcgid в режиме балансировщика.
- *promt-translator.service* — запускает transfcgid в режиме переводчика
- *promt-dcs.service* — запускает процесс DCS
- *promt-managed.service* — запускает процесс SMT/NMT

Для выполнения операций со службой и ее компонентами используется команда вида:

```
sudo systemctl <операция> <имя_сервиса>
```

где <*имя_сервиса*> — имя из списка выше (суффикс `.service` можно опускать), <*операция*> — название операции (start, stop, restart, status).

# Локальный перевод с помощью Promttrans

**promttrans** - утилита командной строки, которая обеспечивает перевод, используя ядро перевода локально. Приложение позволяет протестировать все функции ядра перевода.

**Важно:** утилите `promttrans` в некоторых случаях для работы может потребоваться доступ на запись к некоторым рабочим каталогам PTS. Поскольку сервер PTS работает под специально созданным непривилегированным аккаунтом (по умолчанию — `PTS`), в таких случаях могут возникать ошибки. Избежать этого можно, если запускать `promttrans` в сеансе пользователя, под которым работает PTS:

```
sudo -u PTS ./promttrans.run
```

**Важно:** поскольку каталог, в котором располагается утилита, по умолчанию отсутствует в списке путей поиска (переменная окружения PATH), для ее запуска может потребоваться перейти в каталог `bin64` в корневой папке продукта):

```
cd /usr/local/promt/bin64
```

## Параметры

`-h, -?, -H, -help`
: Отображает справку для promttrans.

`-V, -version`
: Выводит версию программы.

`-f <fileName>, -configfile=<fileName>`
: Позволяет задать конфигурационный файл. Для получения дополнительной информации, см. параграф "Логика поиска конфигурационного файла в ядре".

`-t, --test`
: Проверяет конфигурационный файл. Выводит информацию, если произошла какая-то ошибка.

`-i <fileName>, --inputfile=<fileName>`
: Задает входной файл для перевода. Если файл не задан, то информация читается из стандартного потока ввода.

`-o <fileName>, --outputFile=<fileName>`
: Задает выходной файл, куда записывается результат перевода. Если файл не задан, то результат пишется в стандартный поток вывода.

`-a "<expression>", --attr="<expression>"`
: Передает значение атрибута в ядро перевода в формате `<attrName>=<attrValue>`, где attrName - это имя атрибута, а attrValue - значение.

Вся информация (тип запроса, направление перевода, профиль перевода, кодовые страницы и т.д.) передается через атрибуты. Для упрощения этого процесса, используются следующие опции:

`-G`
`--getservices`
: (`-a "TYPE=GETSERVICES"`) Используется, чтобы получить информацию о доступных языковых парах и профилях перевода. Результат возвращается в виде XML.

`-T`
`--translate`
: (`-a "TYPE=TRANSLATE"`) Используется для перевода текста. Для получения перевода, укажите также направление перевода

Замечание: Ключи `G` и `-T` нельзя использовать одновременно.

`-D <id>`
`--direction=<id>`
: (`-a "DIRECTION=<id>"`) Используется для указания языковой пары. В качестве `id` необходимо указать префикс направления перевода. Для получения списка доступных префиксов, используйте ключ `-G` (в XML ищете тег <id>). Используйте этот параметр вместе с параметром `-T`.

`-M <templateID>`

`--template=<templateID>`

(`-a "TEMPLATE=<templateID>"`) Используется для задания профиля перевода. В качестве `templateID` необходимо указать идентификатор профиля перевода. Для получения списка доступных идентификаторов используйте ключ `-G`.

`-S <codePage>`

`--sourcecp=<codePage>`

(`-a "SOURCECP=<codePage>`") Используется для задания кодировки входного текста. Перекодировка текста осуществляется с помощью библиотеки iconv, а codePage должно быть именем, известным iconv. Для получения списка доступных кодировок, воспользуйтесь командой `iconv --list`. Если кодировка не указана, то по умолчанию используется кодировка UTF8.

Используйте этот параметр вместе с параметром `-T`.

`-R <codePage>`

`--resultcp=<codePage>`

(`--a "RESULTCP=<codePage>"`) Используется для задания кодировки выходного текста. Обратите внимание, что, если в оригинале встретились незнакомые слова, то после перевода они останутся на языке оригинала. Самое простое решение в этой ситуации - использовать выходную кодировку на основе UNICODE, например, UTF8.

Если параметр не указан, то используется UTF8

### Примеры

Ниже даны некоторые примеры использования promttrans:

```
./promttrans.run -G
```

Возвращает список установленных языковых пар и профилей перевода.

```
./promttrans.run -T -D er
```

Выполняет перевод с английского на русский. Исходный текст читается из стандартного потока ввода, а перевод записывается в стандартный поток вывода.

```
./promttrans.run -T -D re -M internet -i source.txt -o result.txt -S utf8 -R utf8
```

Выполняет перевод с русского на английский с использование профиля "internet". Исходный текст читается из файла source.txt в кодировке UTF8, а перевод записывается в файл result.txt file в кодировке UTF8.

# Конфигурационный файл

## Описание

Конфигурационный файл *(promtkernel.conf)* содержит данные, необходимые для ядра PTS. Корректность файла *promtkernel.conf* можно проверить с помощью приложения **promttrans** с ключом *--test*. Ниже Вы найдете подробное описание формата этого файла и допустимые значений всех ключей. Местоположение файла по умолчанию - */usr/local/promt/*.

## Логика поиска конфигурационного файла в ядре

Сначала ядро проверяет значение переменной окружения `PROMT_CONF_NAME`. Если такая переменная доступна, то используется значение этой переменной. Иначе делается попытка открыть конфигурационный файл в каталоге верхнего уровня. Наконец, если эта попытка также потерпела неудачу, используется значение по умолчанию (`/usr/local/promt/promtkernel.conf`).

# Формат файла

Файл содержит секции и параметры. Секция начинается с имени в квадратных скобках. Секция содержит параметры в следующем формате:

```
name=value
```

## Секция General

Первая секция в файле - *[General]*. Обязательным параметром этой секции является ключ *Modules*; значение этого ключа - это список доступных модулей. Вы можете удалить любой модуль из этого списка; в этом случае модуль никогда не будет загружен. Например, Вы можете удалить модуль *log* и прекратить логирование запросов на перевод

**Предупреждение:** не все модули могут быть отключены, в некоторых случаях PTS может стать неработоспособен.

## Секции для модулей

Далее идут секции с описанием модулей, их имена соответствуют названиям модулей в списке модулей параметра *Modules*. Значения некоторых параметров не зависят от модуля:

`Priority` (обязательный параметр)
: Определяет порядок загрузки модулей. Различные модули могут иметь одинаковое значение Priority. Это означает, что только один из таких модулей будет загружен при обработке запроса. В качестве примера можно рассмотреть модули xxMain, где xx определяет направление перевода. Очевидно, что несколько таких модулей не может быть загружено для обработки одного запроса, потому что направление перевода должно быть четко определено.

`Filename` (обязательный параметр)
: Имя библиотеки, которая соответствует данному модулю. Это имя может представлять собой как абсолютный, так и относительный путь. Абсолютный путь будет вычислен с использованием корневого каталога PTS.

`Conditions` (обязательный параметр)
: Это поле содержит список параметров модуля, которые будут использованы для проверки необходимости загрузки модуля. Если у текущего запроса нет таких параметров, модуль не будет загружен. Например, модуль *log* имеет следующие параметры:

```
Conditions=Type
Type=TRANSLATE
```

: Это означает, что модуль *log* будет загружен только для обработки запросов с `Type=TRANSLATION`, т.е. запросов на перевод

`Type` (обязательный параметр)
: Тип запросов, которые обрабатывает модуль. В основном это запросы на перевод, т.е. `Type=TRANSLATION`

# Модули

## log

Этот модуль используется для логирование запросов на перевод. В настройках модуля задаются атрибуты запроса, которые требуется записать в лог. По умолчанию модуль отключен.

**Замечание**: логирование позволяет просмотреть ошибки, которые могут возникать в процессе перевода.

Параметры модуля:

`SuccessAttr`
: Задает список атрибутов, которые должны логироваться, если запрос завершился успешно. Наиболее важные атрибуты описаны ниже.

`ErrorAttr`
: Задает список атрибутов, которые должны логироваться, если запрос завершился неуспешно. Наиболее важные атрибуты описаны ниже

`SuccessFileName`
`ErrorFileName`
: Имя файла для записи успешных или неуспешных запросов соответственно. Имена SuccessFileName и ErrorFileName могут содержать подстановочные символы, которые определяют логику создания новых файлов. Например, имя success%m%y.prmlog.xml задает генерацию нового файла каждый месяц и каждый год, т.е. в мае 2001 года имя будет success0501.prmlog.xml. Для получения дополнительной информации, см. strftime.

## config

Этот модуль отвечает за загрузку словарей и других лингвистических данных, которые необходимы для перевода с заданным профилем перевода. Исключение этого модуля вызовет ошибку.

Модуль загружается при обработке запросов типа `GETSERVICES` и `TRANSLATE`

## ermain, eimain, remain...

Эти модули отвечают для перевод. Каждый модуль соответствует одной языковой паре. Первые символы в имени модуля - это префикс языковой пары (например, **er** для англо-русского). Если исключить какой-то из этих модулей, то соответствующая языковая пара будет недоступна

## plaintrans

Модуль используется для перевода простого текста. Должен иметь более высокий приоритет, чем **xxmain**. Если этот модуль исключить, то простой текст переводиться не будет, а результатом перевода будет пустая строка.

## formattrans

Модуль используется для перевода текста во внутреннем формате. Внутренний формат (if) - это специальная форма представления элементов исходного формата. Внутренний формат генерируется с помощью конвертера из исходного формата.

## html2if, sgml2if, txt2if...

Эти модули конвертируют исходный формат во внутренний формат (if). Первые символы в имени модуля - это название формата.

# Описание атрибутов запроса

Запрос, который обрабатывает ядро PTS, содержит атрибуты и их значения. Ниже приведены наиболее важные атрибуты.

`ERROR`
: Этот атрибут будет содержать значение ошибки. Он должен быть добавлен в параметр ErrorAttr секции log, если вы хотите логировать ошибки PTS.

WARNING
: Этот атрибут будет содержать предупреждение. Предупреждение - это не критическая ошибка, например, ошибка конвертации символов.

TYPE
: Тип запроса, например, GETSERVICES, TRANSLATE

DIRECTION
: Префикс языковой пары, например, `er` значит Англо-Русский перевод

SOURCE
: Исходный текст

SOURCECP
: Кодировка исходного текста

RESULT
: Результат перевода

RESULTCP
: Кодировка результата перевода

TEMPLATE
: Профиль перевода

UNKNOWNWORDS
: Список незнакомых слов

REMOTE_ADDR
: IP адрес запроса

# Конфигурация NMT перевода

## Общая информация

По-умолчанию конфигурация NMT моделей выполняется автоматически с учетом оборудования, на котором запускается модель. Рекомендуется не изменять эти настройки без необходимости.

В PTS настройки NMT модели делятся на глобальные (влияют на все модели) и локальные (позволяют конфигурировать каждую модель в отдельности). Локальные настройки имеют приоритет над глобальными, и поэтому возможны конфигурации, когда каждая модель будет работать с собственными настройками. В зависимости от сценариев использования PTS этот подход позволяет:

1. Использовать одинаковые настройки для всех моделей
2. Использовать одинаковые настройки для части моделей, а другую часть конфигурировать для каждой модели в отдельности.
3. Использовать разные настройки для всех моделей.
4. Задавать для каждой модели один или несколько вычислительных юнитов (CPU или GPU). К примеру, возможны конфигурации, где 2 модели работают на GPU, а остальные на CPU.
5. Использовать более одной видеокарты для одной модели.

## Глобальные настройки NMT

В конфигурационном файле PTS (обычно находится по пути /usr/local/promt/promtkernel.conf) за глобальную настройку NMT моделей отвечают два параметра:

```
[Managed]
NmtBatchSize=0
NmtOptions=
```

Параметр NmtBatchSize – это размер батча (количество строк), которые передаются в Marian на перевод текста. Значение данного параметра – это баланс между скоростью перевода (для высоких значений) и скоростью отклика перевода (для низких значений). Как правило подбирается экспериментально в зависимости от требований производительности.

Параметр NmtOptions – это строка инициализации NMT движка Marian, которая может содержать несколько десятков параметров в зависимости от конкретной конфигурации модели и оборудования. По-умолчанию эти настройки пустые и генерируются автоматически в зависимости от конфигурации оборудования, на котором запущен PTS. Логика определения настроек:

1. Определяется количество видеокарт, поддерживающих CUDA. Если такие видеокарты найдены – все они используются для перевода NMT. Если видеокарт не найдено – используется процессор. Параметр NmtBatchSize задается из расчета 64 для каждой видеокарты.
2. Определяется количество физических ядер процессора (не путать с количеством логических ядер) и наличие инструкций AVX2. Все физические ядра используются для перевода NMT. Оптимизации перевода включаются при поддержке процессором инструкций AVX2. Параметр NmtBatchSize задается из расчета 2 для каждого ядра процессора.

К примеру, для конфигурации с одной видеокартой будут определены такие параметры:

```
NmtBatchSize=64
NmtOptions=--log-level off --beam-size 6 --normalize 0.6  --workspace 1280 --workspace-limit true --mini-batch 8 --maxi-batch 100 --max-length 100 --max-length-crop true --devices 0
```

Для конфигурации с восьмиядерным процессором будут определены такие параметры:

```
NmtBatchSize=16
NmtOptions=--log-level off --beam-size 6 --normalize 0.6 --workspace 128 --workspace-limit false --mini-batch 1 --maxi-batch 1 --max-length 100 --max-length-crop true --cpu-threads 8 --optimize --gemm-type fp16packed
```

# Локальные настройки NMT

Локальные настройки NMT перевода находятся в конфигурационном файле model.xml соответствующей модели. К примеру для ER направления этот файл находится по пути: /usr/local/promt/data/snmt/er_release_04.2019/model.xml. По аналогии с глобальными настройками, в этом файле могут быть заданы настройки NMT, которые по-умолчанию скрыты. Пример конфигурации с одной видеокартой:

```
<?xml version="1.0" encoding="utf-8"?>
<model>
  <nmt_batch_size>64</nmt_batch_size>
  <nmt_options>--log-level off --beam-size 6 --normalize 0.6  --workspace 1280 --workspace-limit true --mini-batch 8 --maxi-batch 100 --max-length 100 --max-length-crop true --devices 0</nmt_options>
</model>
```

При задании этих настроек модель будет игнорировать глобальные настройки.

# Строка инициализации Marian

Большинство параметров в строке инициализации Marian являются константами, которые не требуется изменять ни при каких конфигурациях. Однако некоторые из них можно менять в зависимости от сценария использования PTS и оборудования сервера, вот некоторые из них:

--log-level off: отключение вывода в консоль информации в процессе работы Marian.
--workspace N: задание начального размера рабочего пространства перевода в мегабайтах. Напрямую влияет на потребление памяти, используется в основном при переводе на GPU.
--workspace-limit true: ограничение размера рабочего пространства. При попытке увеличения памяти процесса перевод текущей порции строк будет прекращен, и работа процесса продолжится дальше. Используется в основном при переводе на GPU.
--mini-batch N: размер порции (количество строк), которое используется в Marian для перевода. Является балансом между скоростью перевода (для больших значений) и скоростью отклика (для маленьких значений).

--maxi-batch N: количество порций, отсортированных по длине, на которые делится входящий текст внутри Marian. Является настройкой оптимизации для повышения производительности перевода GPU.
--max-length N: максимальная длина входного текста в токенах. Используется в связке с параметром max-length-crop для обрезки длинных строк, напрямую влияет на потребление памяти при переводе.
--max-length-crop true: включение обрезки длинных строк значение, которое задано параметром max-length.
--devices N1 N2: задание порядковых номеров видеокарт, которые будут использованы при переводе с текущей моделью. Как правило нумерация начинается с нуля, но номера можно задавать в любом порядке и в любом количестве.
--cpu-threads N: задание количества потоков CPU, которые будут использованы при переводе с текущей моделью. Рекомендуется задавать значения от 1 до N, где N – количество физических ядер CPU.

# Поддержка нескольких GPU

По-умолчанию PTS будет использовать все имеющиеся видеокарты для всех NMT моделей. Эту логику можно менять с помощью локальных параметров. Пример конфигурации с использованием 1й видеокарты для отдельно взятой модели:

```
<nmt_batch_size>64</nmt_batch_size>
<nmt_options>--log-level off --beam-size 6 --normalize 0.6  --workspace 1280 --workspace-limit true --mini-batch 8 --maxi-batch 100 --max-length 100 --max-length-crop true --devices 0</nmt_options>
```

Пример конфигурации с использованием 4х видеокарт для отдельно взятой модели:

```
<nmt_batch_size>128</nmt_batch_size>
<nmt_options>--log-level off --beam-size 6 --normalize 0.6  --workspace 1280 --workspace-limit true --mini-batch 8 --maxi-batch 100 --max-length 100 --max-length-crop true --devices 0 1 2 3</nmt_options>
```

Пример конфигурации с использованием 2х видеокарт для двух отдельных моделей:

```
<nmt_batch_size>128</nmt_batch_size>
<nmt_options>--log-level off --beam-size 6 --normalize 0.6  --workspace 1280 --workspace-limit true --mini-batch 8 --maxi-batch 100 --max-length 100 --max-length-crop true --devices 0 1</nmt_options>

<nmt_batch_size>128</nmt_batch_size>
<nmt_options>--log-level off --beam-size 6 --normalize 0.6  --workspace 1280 --workspace-limit true --mini-batch 8 --maxi-batch 100 --max-length 100 --max-length-crop true --devices 2 3</nmt_options>
```

# Ограничение памяти GPU

Так как количество памяти GPU сильно ограничено по сравнению с оперативной памятью, то в PTS есть возможность задания лимита памяти. Количество памяти GPU для каждой модели складывается из размера модели (в среднем около 500Мб) и размера рабочего пространства модели, которую можно ограничить с помощью параметров инициализации Marian:

```
--workspace 1280 --workspace-limit true
```

Параметр workspace задает начальное значение рабочего пространства NMT в мегабайтах, а параметр --workspace-limit включает ограничение этой памяти. При попытке перевода текста, который вызывает увеличение рабочего пространства за пределы лимита, будет сгенерировано исключение, которое в дальнейшем корректно обработается и выполнится откат к RBMT переводу (в случае если это возможно).

Значения по-умолчанию подобраны таким образом, чтобы каждая модель потребляла не более 2Gb памяти GPU. Таким образом, на видеокарте уровня GTX1080 можно запустить одновременно до 4х NMT моделей. В случае необходимости лимит памяти можно уменьшить до 512Мб, тогда общее потребление модели составит 1Gb памяти и в GTX1080 поместится 8 моделей.

# Настройки перевода

## Варианты перевода

В PTS существует функциональность по отображению вариантов перевода отдельных слов. Данная функциональность включается в конфигурационном файле направления на уровне отдельной тематики:

```
[Templates]
Templates=Animals

[Animals]
Name=Animals
Variants=1
```

Строка "Variants" является необязательной, состояние по умолчанию - выключено.

## Учет регистра в словах без перевода

В PTS существует возможность задания свойства "MatchCase" в словах без перевода (WWT), которое влияет на игнорирование регистра при поиске таких слов. Данная функциональность реализована добавлением нового числа в свойствах слова (в конфигурационном файле направления на уровне отдельной тематики):

```
[Templates]
Templates=Animals

[Animals]
Name=Animals
WWT=cat:1:0:0,dog:1:0:1
```

В данном примере слово "cat" будет искаться без учета регистра, а слово "dog" – с учетом.

# Балансировка нагрузки

## Общее описание

По-умолчанию PTS устанавливается в конфигурации, где балансировщик нагрузки располагается на том же компьютере, что и сервер перевода. Однако PTS поддерживает горизонтальное масштабирование производительности перевода за счет возможности распределения нагрузки на другие сервера. Для этого требуются модификации следующих параметров конфигурационного файла PTS:

В секции General:

- **Threads** – количество потоков, выделенное для перевода на текущем сервере. Значение по-умолчанию 0 означает количество потоков равное количеству ядер в процессоре.

В секции Balancer:

- **Threads** – количество потоков, выделенное для балансировки на текущем сервере. Значение по-умолчанию 0 означает количество потоков равное количеству ядер в процессоре.
- Список серверов перевода в формате "**TSX=127.0.0.1;N**", где X – это порядковый номер сервера перевода (нумерация начинается с единицы, т.е. TS1 – это первый сервер), N – вес сервера (производительность в сравнении с остальными).

При задании этих параметров нужно исходить из общих параметров кластера серверов перевода. К примеру планируется создание кластера из 3 компьютеров, на одном из которых будет находиться балансировщик. Два из этих компьютеров имеет по 4 процессорных ядра, а третий – 16 ядер (для простоты будем считать, что производительность самих ядер и остальные параметры системы одинаковы). В таком случае рекомендуется следующая конфигурация балансировщика (некоторые параметры пропущены):

```
[General]
Threads=2

[Balancer]
Threads=50
TS1=127.0.0.1;2;
TS2=192.168.0.100;4;
TS3=192.168.0.101;16;
```

После изменения конфигурации требуется перезапустить сервисы PTS на компьютере, где запущен балансировщик. На всех серверах перевода должны быть развернуты одинаковые экземпляры PTS, лицензия должна быть установлена только на сервере с балансировщиком.

В такой конфигурации балансировщик будет работать на 50 одновременных входящих подключений (остальные будут ожидать в очереди обработки) и распределять нагрузку на 20 суммарных ядер перевода на 3 серверах (один из которых – он сам). На сервере с балансировщиком для перевода будут доступны только 2 потока из 4. Сами потоки балансировщика не вызывают значительной нагрузки на процессор, однако рекомендуется иметь хотя бы два свободных ядра, чтобы не возникло ситуации когда потоки перевода будут конкурировать с потоками балансировщика и тем самым снижать суммарную производительность кластера. При дальнейшем увеличении количества серверов перевода имеет смысл вообще исключить компьютер с балансировщиком из списка серверов перевода.

Количество потоков балансировщика рекомендуется задавать большим, чем суммарное значение ядер перевода из-за возможной неравномерности запросов на перевод - что может приводить к ситуации, когда один сервер будет нагружен на 100%, а остальные будут простаивать при неполной нагрузке. Веса серверов перевода следует выставлять пропорционально производительности каждого сервера, при прочих равных условиях таким числом можно считать количество ядер процессора (в реальности будут также иметь значение тип и частота оперативной памяти, производительность самого процессора, загруженность сервера другими задачами и т.п.)

# Балансировка отдельных направлений

В PTS существует возможность задания направлений перевода для каждого сервера перевода в отдельности. Это может быть полезно для более тонкой настройки распределения нагрузки между серверами. Настройка осуществляется путем задания списка направлений перевода в конфигурационном файле:

```
[Balancer]
Threads=50
TS1=127.0.0.1;er,re;
TS2=192.168.0.100;eg,ge;
```

В данном примере TS1 будет переводить только ER и RE направления, TS2 – только EG и GE. В случае отсутствия направлений считается, что сервер поддерживает все доступные направления.