119270, Москва, Лужнецкая наб., д. 6,
стр.1, офис 214, ООО «ЭР СИ О»
Тел./факс: (495) 287-98-87
E-mail: info@rco.ru
http://www.rco.ru

Руководство разработчика

RCO Personal Information SDK – программа для выявления и сокрытия персональных данных

Версия 1.1

Москва, 2024

В содержание данного документа могут быть внесены изменения без предварительного уведомления. Названия организаций, имена и даты, используемые в качестве примеров, являются вымышленными, если не оговорено обратное.

Содержание

Введение

RCO Personal Information SDK позволяет получать деперсонифицированный текст и информацию о найденных в тексте персональных данных и произведенных заменах на соответствующие кодовые обозначения. Поставляется в сочетании с пакетом **RCO Fact Extractor SDK**, вместе со специализированными лингвистическими ресурсами, оптимизированными под задачу, и конфигурацией со списком типов персональных данных и кодовых обозначений при их замене.

RCO Personal Information SDK анализирует поступающий документ и извлекает из него упоминания персональных данных, в том числе:

- ФИО;
- Данные идентифицирующих документов (паспорт, СНИЛС, ИНН и прочие);
- Биографические данные: дата и место рождения;
- Адресные данные, данные для связи (место жительства, телефон, e-mail и прочие);
- Информация о собственности (марка, модель, номер транспортного средства, кадастровый номер недвижимости);
- Сведения о финансах и медицинские идентифицирующие данные (номер банковского счета, номер медицинской карты, ОМС и прочие).

RCO Personal Information SDK устанавливает связи между упомянутыми персонами и персональными данными, осуществляет замену персональных данных на их обозначения в зависимости от категории в соответствии с заданными настройками.

При выделении персональных данных **RCO Personal Information SDK** использует результаты синтактико-семантического анализа текста. Обеспечена совместимость на уровне программных интерфейсов с **RCO Personal Information SDK**, выполняющим синтактико-семантического анализ.

RCO Personal Information SDK имеет возможность расширять список категорий извлекаемых персональных данных, поддерживает русский язык и функционирует в следующих операционных системах:

- Astra Linux;
- AltLinux;
- Windows.

Внимание! Использование **RCO Personal Information SDK** возможно лишь при наличии работающей версии **RCO Fact Extractor SDK**.

Порядок работы с библиотекой

Взаимодействие с библиотекой осуществляется через описываемый в данном документе программный интерфейс по следующему алгоритму:

1. Загрузка библиотеки **RCO Fact Extractor**;
2. Создание контекста **RCO Fact Extractor**;
3. Загрузка библиотеки **RCO Personal Information**;
4. Создание дескриптор обработчика;
5. Инициализация дескриптор обработчика;
6. Разбор текст с помощью **RCO Fact Extractor**;
7. Обработка текста, разобранного **RCO Fact Extractor**;
8. Запуск выявления персональных данных в тексте;
9. Получение деперсонифицированного текста и информации о произведенных заменах;
10. Удаление дескриптора результатов деперсонификации;
11. Удаление дескриптора результатов разбора с помощью **RCO Fact Extractor**;
12. Удаление дескриптора обработчика;
13. Удаление контекста **RCO Fact Extractor**.

Шаги 6-11 могут повторяться в цикле.

Интерфейс библиотеки

Функции работы с персональными данными

DPCreateHandle

Функция создает дескриптор обработчика.

```
DP_ERROR DPCreateHandle( DP_HANDLE* phDP, DPFunctionTypeLog
   pLogCallback, void* pLogContext ););
```

Параметры

phDP
[вых] Указатель на созданный дескриптор.

pLogCallback
[вх] Указатель на функцию обработки сообщений о работе библиотеки (сигнатура функции описана в Приложении 1). Функция по указателю будет вызываться в процессе инициализации дескриптора и обработки текста.

pLogContext
[вх] Контекст вызова функции pLogCallback.

Возвращаемые значения

При успешном завершении – DP_SUCCESS (0). При передаче в функцию некорректных аргументов – DP_ERR_INVALID_ARGUMENT (-2).

DPDestroyHandle

Функция удаляет дескриптор обработчика, освобождая выделенную под него память.

```
DP_ERROR DPDestroyHandle( DP_HANDLE hDP );
```

Параметры

hDP
[вх] Удаляемый дескриптор. Если hDP равен 0, то удаление не производится.

Возвращаемые значения

Всегда успешное завершение – DP_SUCCESS (0).

DPInitializeHandle

Функция инициализирует дескриптор обработчика дескриптор обработчика.

```
DP_ERROR DPInitializeHandle( DP_HANDLE hDP, const wchar_t* pwsConfig,
   const wchar_t* pwsLangParameters, void* pResManagerArray, void*
   reserved );
```

Параметры

hDP
[вх] Инициализируемый дескриптор.

pwsConfig
[вх] Указатель на JSON с описанием конфигурации обработчика (описание формата конфигурации приведено в Приложении 2).

pwsLangParameters
[вх] Указатель на строку с описанием параметров языка обработки (строка должна содержать путь к библиотеке анализа текста RCO Fact Extractor и код языка; пример строки для разбора текстов на русском языке в ОС Windows: "DLL:RCOFXRu.dll,LANG:RUS/Russian").

pResManagerArray
[вх] Указатель на массив с указателями на функции работы с ресурсами и их контекст (сигнатуры функций работы с ресурсами приведены в Приложении 1).

Возвращаемые значения

При успешном завершении – DP_SUCCESS (0). При передаче в функцию некорректных параметров – DP_ERR_INVALID_ARGUMENT (-2), при наличии прочих ошибок – DP_ERR_UNDEFINED (-1).

DPProcessText

Функция производит разбор текста с использованием указанного дескриптора, выявляет в тексте персональные данные, формируя деперсонифицированный текст, в котором персональные данные заменены на кодовые обозначения.

```
DP_ERROR DPProcessText( DP_HANDLE hDP, const wchar_t* pwsText, const
   wchar_t* pwsTextID, void* hFXResult, DP_HANDLE* phResult );
```

Параметры

hDP
[вх] Используемый дескриптор.

pwsText
[вх] Указатель на строку текста для обработки.

pwsTextID
[вх] Указатель на идентификатор текста (идентификатор используется при формировании сообщений о работе библиотеки).

hFXResult
[вых] Дескриптор результатов разбора текста, полученный в функции FXProcessText библиотеки RCO Fact Extractor.

phResult
[вых] Указатель на дескриптор результатов разбора текста.

Возвращаемые значения

При успешном завершении – DP_SUCCESS (0). При передаче в функцию некорректных параметров – DP_ERR_INVALID_ARGUMENT (-2), при наличии прочих ошибок – DP_ERR_UNDEFINED (-1).

DPDestroyResult

Функция удаляет дескриптор результата обработки текста, освобождая выделенную под него память.

```
DP_ERROR DPDestroyResult( DP_HANDLE hResult )
```

Параметры

hResult
[вх] Удаляемый дескриптор результата разбора текста. Если hResult равен 0, то удаление не производится.

Возвращаемые значения

Всегда успешное завершение – DP_SUCCESS (0).

DPGetDepersonText

Функция выдает деперсонифицированный текст (переданный в DPProcessText), в котором персональные данные заменены на кодовые обозначения.

```
DP_ERROR DPGetDepersonText( DP_HANDLE hResult, const wchar_t** ppText
   );
```

Параметры

hResult
[вх] Дескриптор результата разбора текста.

ppText
[вых] Указатель на строку с деперсонифицированным текстом.

Возвращаемые значения

При успешном завершении – DP_SUCCESS (0). При передаче в функцию некорректных параметров – DP_ERR_INVALID_ARGUMENT (-2).

DPGetDepersonInfo

Функция выдает информацию о найденных в тексте персональных данных и произведенных заменах на кодовые обозначения.

```
DP_ERROR DPGetDepersonInfo( DP_HANDLE hResult, const wchar_t*
   pFormat, const wchar_t** ppInfo );
```

Параметры

hResult
[вх] Дескриптор результата разбора текста.

pFormat
[вх] Указатель на строку с типом форматирования результата (на данный момент поддерживается только JSON).

ppInfo
[вых] Указатель на строку, содержащую в форматированном виде информацию о найденных в тексте персональных данных и произведенных заменах на кодовые обозначения (формат результатов замены приведен в Приложении 3).

Возвращаемые значения

При успешном завершении – DP_SUCCESS (0). При передаче в функцию некорректных параметров – DP_ERR_INVALID_ARGUMENT (-2); при указании неподдерживаемого типа форматирования результата – DP_ERR_UNSUPPORTED_FORMAT (-3).

Приложение 1. Сигнатуры функций логирования и работы с ресурсами

В данном приложении приведены типы функций работы с ресурсами (аналогичны функциям работы с ресурсами в SDK Fact Extractor) и тип функции работы с логами.

Открытие ресурса

```
void* (*DPFunctionTypeResManagerOpen)(void* context, const wchar_t*
   szResourceId);
```

Параметры:

Context
[вх] Контекст вызова функции.

szResourceId
[вх] Идентификатор открываемого ресурса.

Возвращаемое значение:

Дескриптор открытого ресурса. Возвращает 0, если ресурс не удалось открыть.

Закрытие ресурса

```
void (*DPFunctionTypeResManagerClose)(void* context, void*
   hResource);
```

Параметры:

Context
[вх] Контекст вызова функции.

hResource
[вх] Дескриптор подлежащего закрытию ресурса.

Чтение ресурса

```
bool (*DPFunctionTypeResManagerRead)(void* context, void* hResource,
   void* pBuffer, unsigned long ulNumberOfBytesToRead, unsigned long*
   pulNumberOfBytesRead);
```

Параметры:

Context
[вх] Контекст вызова функции.

hResource
[вх] Дескриптор читаемого ресурса.

pBuffer
[вх] Указатель на область памяти, в которую будут записаны считанные из ресурса данные. Допустимо значение *NULL* – в этом случае функция запишет по адресу *pulNumberOfBytesRead* минимальный размер требуемого буфера в байтах.

ulNumberOfBytesToRead
[вх] Количество байтов, запрашиваемых на чтение.

pulNumberOfBytesRead
[вых] Указатель на переменную, в которую будет записано количество реально считанных байтов или размер требуемого буфера (если передано значение *pBuffer*, равное *NULL*).

Возвращаемое значение:

Флаг успешности чтения ресурса.

Работа с логами

```
void(*DPFunctionTypeLog)(DPLogLevel eLL, const wchar_t* pwsText,
   void* pContext);
```

Параметры:

eLL
[вх] Уровень сообщения. Возможные значения – eDPLL_Error (0), eDPLL_Warn (1), eDPLL_Info (2), eDPLL_Trace(3).

pwsText
[вх] Текст сообщения.

pContext
[вх] Контекст вызова функции.

Приложение 2. Формат конфигурации обработчика

В данном приложении приведено описание конфигурации обработчика.

```
{
  "deperson_replacements": [
    {
      "semantic_type": "Person:Name",
      "name": "Персона",
      "case_names": ["Персоны", "Персоне", "Персону", "Персоной",
   "Персоне"],
      "counted": true
    },
    {
      "semantic_type": "Special:PlaceOfBirth",
      "name": "[МестоРождения]",
      "counted": false
      }
  ]
}
```

semantic_type – семантический тип сущности в результатах разбора текста RCO Fact Extractor, которая содержит персональные данные, текст которой должен быть заменен на кодовое обозначение;

name – кодовое обозначение, на которое должен быть заменен текст с персональными данными;

case_names – кодовое обозначение в различных падежах (опционально): родительном, дательном, винительном, творительном, предложном;

counted – флаг: если указано значение true, то при замене к кодовому обозначению будет добавлен номер объекта, начиная с 1 (используется для персон).

Приложение 3. Формат результата замены персональных данных

В данном приложении приведено описание формата результатов замены персональных данных на кодовые обозначения.

```
"replacements": [
   {
     "replace_index": 0,
     "text": "Петя",
     "name": "Персона_1",
     "original_text_offset": 0,
     "original_text_length": 4,
     "replaced_text_offset": 0,
     "replaced_text_length": 13,
     "parent_replace_index": -1
   },
   {
     "replace_index": 1,
     "text": "Контактный телефон: 8902567666",
     "name": "[Телефон]",
     "original_text_offset": 54,
     "original_text_length": 30,
     "replaced_text_offset": 58,
     "replaced_text_length": 9,
     "parent_replace_index": 0
   }
]
}
```

replace_index – номер замены в тексте, начиная с 0;

text – строка в исходном тексте;

name – кодовое название, на которое заменяются персональные данные;

original_text_offset и **original_text_length** – позиция персональных данных в исходном тексте (смещение от начала текста и длина в символах соответственно);

replaced_text_offset и **replaced_text_length** – позиция замены в деперсонифицированном тексте (смещение от начала текста и длина в символах соответственно);

parent_replace_index – -1 или номер замены имени персоны, если удалось привязать данные персональные данные к какой-то персоне.