



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

Руководство разработчика RCO Fact Extractor SDK – библиотека лингвистического анализа текста

(Версия для Linux, Microsoft Windows)

Москва, 2021

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.

© ООО «ЭР СИ О», 2021. Все права защищены.

ЭР СИ О, Russian Context Optimizer, RCO являются охраняемыми товарными знаками.

ООО «ЭР СИ О» может являться правообладателем патентов и заявок, поданных на получение патента, товарных знаков и объектов авторского права, которые имеют отношение к содержанию данного документа.

Предоставление вам данного документа не означает передачи какой-либо лицензии на использование данных патентов, товарных знаков и объектов авторского права, за исключением использования, явно оговоренного в лицензионном соглашении ООО «ЭР СИ О».

Все другие названия юридических лиц и изделий являются охраняемыми товарными знаками или товарными знаками, принадлежащими их владельцам.

Содержание

Обзор	5
Введение.....	7
Выделение сущностей	8
Семантическая сеть	10
Фреймы и поиск ситуаций	12
Настройка лингвистического обеспечения	14
Порядок работы с библиотекой	15
Инициализация	15
Типовой сценарий работы.....	16
Интерфейс библиотеки.....	17
Интерфейс для доступа к лингвистическим ресурсам.....	17
Open	17
Close.....	17
Read	18
Интерфейс для добавления и переопределения внешних функций.....	18
FirstFunction.....	19
NextFunction	19
GetAddress	19
GetContext.....	19
GetName.....	19
GetType	19
Функции для работы с анализатором текста.....	21
FXCreateContext	21
FXDestroyContext	22
FXProcessText.....	23
FXDestroyResult	24
Функции для работы с результатом анализа текста	25
FXGetFirst	25
FXGetNext	27
FXGetAttributeValueCount	28
FXGetAttributeValue.....	29
FXSetEntityNameGenerationMode	31
FXGetEntityNameGenerationMode	31
Работа с сущностями в предложении: функции FXGetFirst, FXGetNext.....	32
Получение первого предложения текста	32
Получение очередного предложения текста.....	32
Получение первой сущности в предложении	33
Получение очередной сущности в предложении	33
Получение предложения, содержащего сущность	34
Работа со связями сущности: функции FXGetFirst, FXGetNext.....	35
Получение первой связи сущности	35
Получение очередной связи сущности	35
Получение сущности, в которую входит связь.....	36
Получение коннектора, при помощи которого установлена связь	36
Получение сущности, кореферентной заданной.....	37

Работа с фреймами: функции FXGetFirst, FXGetNext	38
Получение первого фрейма в тексте	38
Получение очередного фрейма в тексте	38
Получение первого слота фрейма	39
Получение очередного слота фрейма.....	39
Получение сущности, соответствующей слоту фрейма	40
Работа с актуальным контекстом: функции FXGetFirst, FXGetNext	41
Получение первого актуального контекста текста	41
Получение очередного актуального контекста текста	41
Получение сущности, соответствующей актуальному контексту	42
Получение информации о текущей настройке библиотеки: функция FXGetMetaData .	43
Описание атрибутов	44
Атрибуты предложения.....	44
Атрибуты сущности	46
Атрибуты связи.....	52
Атрибуты фрейма	55
Атрибуты контекста	56
Атрибуты текста	56
Приложение 1. Формат конфигурационного файла	58
Приложение 2. Возможные значения кодов ошибки	65
Приложение 3. Формат файла журнала.....	66
Приложение 4. Типы выделяемых сущностей.....	67
Приложение 5. Управление генерацией имен сущностей.....	69
Приложение 6. Описание место-временного контекста.....	72
Показатели места: система описания	73
Семантические атрибуты.....	73
Показатели времени: система описания	73
Семантические атрибуты.....	73
Структурные атрибуты	75
Приложение 7. Классификация частей речи	76
Приложение 8. Описание парадигмы словоизменения	78
Приложение 9 Классификация падежей, родов и указаний на число	83
Приложение 10 Словари, поставляемые в стандартной конфигурации RCO FX Ru.....	84
Приложение 11 Фильтры, поставляемые в стандартной конфигурации RCO FX Ru.....	89
Приложение 12 Правила, поставляемые в стандартной конфигурации RCO FX Ru.....	92

Обзор

В настоящем документе описывается ядро пакета **RCO Fact Extractor SDK** – библиотека **RCO FX Ru**.

Библиотека **RCO FX Ru** осуществляет полный синтактико-семантический разбор текста в форматах HTML и TXT. Библиотека выделяет различные классы сущностей, упомянутых в тексте (персоны, организации, география, предметы, действия, атрибуты и др.), и строит сеть отношений, связывающих эти сущности, а также предоставляет всю грамматическую информацию о составляющих текста. Средства библиотеки также обеспечивают семантическую интерпретацию результатов разбора текста – поиск описаний ситуаций, удовлетворяющих заданным семантическим шаблонам.

Библиотека работает на платформах **Linux** и **Windows**. Выпускается в редакциях:

- **Standard** – выявление различных типов сущностей, синтез тематического портрета, учет референтных упоминаний персон и организаций;
- **Standard Plus** – добавлена возможность поиска описаний фактов;
- **Professional** – помимо прочего, есть доступ к семантической сети текста;
- **Exclusive** – включение собственных модулей в процесс обработки, поставка словарей и правил в открытом виде, настройки и доработки под клиента.

В состав лингвистического обеспечения пакета, помимо общих словарей и правил русского языка, входят правила выделения специальных объектов (дат, адресов, документов, телефонов, денежных сумм, марок автомобилей и пр.), шаблоны для распознавания различных классов событий и фактов (сделок, биографические факты, конфликты, экономических показателей и пр.), характеристик объекта (позитива, негатива и др.), высказываний прямой и косвенной речи.

Для эффективного решения ряда прикладных задач на основе библиотеки **RCO FX Ru** дополнительно предлагаются исходные коды соответствующих приложений на C++, в том числе для:

- построения смыслового портрета документа – множества слов и словосочетаний, ранжированных по значимости;
- построения реферата текста, в том числе рефератов по каждой сущности;
- построения ассоциативной семантической сети текста или коллекции текстов;
- построения глоссария по коллекции текстов;
- трансляции запроса на естественном языке в пакет запросов поисковой машины.

Обрабатываемая кодировка текста – Win1251.

Формат текста – HTML, либо простой текст.

В состав библиотеки входят следующие файлы:

rcofxru.dll	Собственно библиотека (только в Windows-версии).
librcofxru.so	Собственно библиотека (только в Linux- версии).
include*.h	Заголовочные файлы с описанием констант и прототипов функций.
rcofxru.lib	Lib-файл для компоновки в среде MS Visual C (только в Windows-версии).
fx.ini	Ini-файл, автоматически загружаемый при начале работы библиотеки.
config.xml	Конфигурационный файл, содержащий описание ресурсов и настроек, используемых при работе библиотеки.
dic\MorphDct.dat, dic\MorphDctNames.dat	Бинарные словари морфологического анализа.
dic\LingDct.dat	Бинарный словарь моделей управления.
dic*.rlt	Файлы с CAPE -правилами для токенизатора.
dic*.rul	Набор файлов, содержащих правила выделения в тексте специальных конструкций. Некоторые из этих файлов могут быть закодированы и недоступны для просмотра.
dic*.dct, dic*.flt	Набор файлов-словарей, содержащих ключевые слова для выделения в тексте специальных конструкций, именованных объектов и проч. Некоторые из этих файлов могут быть закодированы и недоступны для просмотра.
dic\shortcuts.txt, dic\shortcuts_endsent.txt	Словари сокращений, используемых при разбивке текста на предложения и слова. Могут быть закодированы и недоступны для просмотра.
ld*.dot	Семантические шаблоны – описания фреймов, создаваемые при помощи приложения RCO Fact Tuner .
ld*.flt	Словари, используемые при создании семантических фильтров на базе семантических шаблонов. Могут быть закодированы и недоступны для просмотра.
objects*.xml	Несколько описаний именованных объектов (для примера).

Дополнительно в комплект поставки входит файл **test.cpp**, содержащий исходный код тестового примера, иллюстрирующего использование библиотеки.

Введение

В данной главе приводится общая информация о возможностях анализа текста, предоставляемых библиотекой **RCO FX Ru**. В зависимости от приобретаемой вами конфигурации библиотеки, те или иные возможности могут быть недоступны.

Подробную информацию о работе алгоритмов библиотеки можно получить в статьях, доступных на сайте разработчика (<http://www.rco.ru>):

1. Ермаков А.Е., Плешко В.В. Семантическая интерпретация в системах компьютерного анализа текста // Информационные технологии. – 2009. – N 6 (http://www.rco.ru/attach.asp?a_no=216).
2. Ермаков А.Е., Плешко В.В., Митюнин В.А. RCO Pattern Extractor: компонент выделения особых объектов в тексте // Информатизация и информационная безопасность правоохранительных органов: XI Международная научная конференция. Сборник трудов – Москва, 2003 (http://www.rco.ru/article.asp?ob_no=237).

Выделение сущностей

Базовые единицы содержания текста в принятой здесь терминологии называются сущностями.

Сущность (entity) – нечто (предмет, существо, событие и т.п.), обозначенное в тексте словом (*вкладывать, эмиссия, Лукойл, металлический, скоро*), словосочетанием (*компьютерный анализ текста, Тюменская нефтяная компания, Александр Иванов, пятью неделями раньше*), специальной численно-символьной конструкцией (*т. 930-8500, \$3000, 98%, ИНН 9378532*).

Кроме того, к сущностям формально будут относиться все строевые элементы, присутствующие в тексте и не имеющие сами по себе никакого значения – слова служебных частей речи (союзы, предлоги), знаки препинания. Это удобно для поддержания единообразного интерфейса, позволяющего получать всю информацию как о содержании, так и о формальной организации текста.

Значимые сущности, упоминаемые в тексте и выделяемые средствами библиотеки, можно разделить на три класса:

1. Именованные сущности: персоны, организации, географические объекты, артефакты (продукты человеческой деятельности) и прочие, обозначаемые в тексте с использованием имен собственных. Для определения имен собственных и приведения их к нормальной форме используются алгоритмы словарного и бессловарного морфологического анализа, информация о возможных способах написания названий в русском языке, алгоритмы синтаксического анализа и снятия омонимии, алгоритмы установления кореферентности обозначений. Программа сможет отождествить различные обозначения одного и того же предприятия в тексте, например *ООО «Ковдорский горно-обогатительный комбинат», Ковдорский горно-обогатительный комбинат, ГОК, горно-обогатительный комбинат, предприятие, которое*. Аналогично, программа может отождествить такие обозначения одной персоны, как *Владимир Иванович Банк, Владимир, Банк, он*, отличив по контексту город *Владимир* от человека *Владимира* и человека с фамилией *Банк* от финансового учреждения. Для высокоточной обработки обозначений заранее известных персон и организаций в библиотеке предусмотрена возможность подключения их детализированных описаний (см. документацию «*Руководство администратора RCO. Настройка описаний объектов*»).

2. Неименованные сущности, обозначенные в тексте словами основных частей речи: нарицательными именами существительными, прилагательными, нестроевыми глаголами – одушевленными и неодушевленными предметами (*новое оружие массового поражения, тариф на энергоноситель*), признаки (*нищий, нищета*), события (*голодовка шахтеров, требовать понизить тариф на энергоноситель*). Имена таких сущностей определяются на основании синтаксического анализа текста и правил русского языка. Так, от каждого упоминания имени существительного в тексте программой строятся нормальные формы всех именных групп, главным словом которых выступает данное существительное. Например, если в тексте упоминается *новое химическое и бактериологическое оружие*, то библиотека позволит получить имена всех неодушевленных предметов, о которых шла речь: *новое химическое оружие, новое бактериологическое оружие, химическое оружие, бактериологическое оружие, оружие*. Для каждого имени подобной сущности можно получить информацию о том, является данное имя полным или производным – употреблялось ли данное имя в тексте самостоятельно или только в составе другого имени. Множество всех производных имен в сочетании с полными именами представляет собой семантический портрет текста, который содержит все упомянутые элементы смысла и может быть эффективно использован для автоматической классификации и поиска документов. Полные имена можно использовать для представления человеку краткого резюме документа – списка ключевых тем.

3. Специальные сущности – сущности, обозначаемые в тексте конструкциями особого вида, обычно смешанными из цифр и символов: даты и обстоятельства времени, денежные суммы, идентификационные данные персон и организаций и т.п. Такие конструкции распознаются благодаря формальным правилам, написанным на особом языке. Пользователь библиотеки имеет возможность добавлять свои правила для распознавания новых типов специальных сущностей, например, номеров автомобилей (см. документацию «*RCO Pattern Extractor – компонент выделения конструкций в тексте*»).

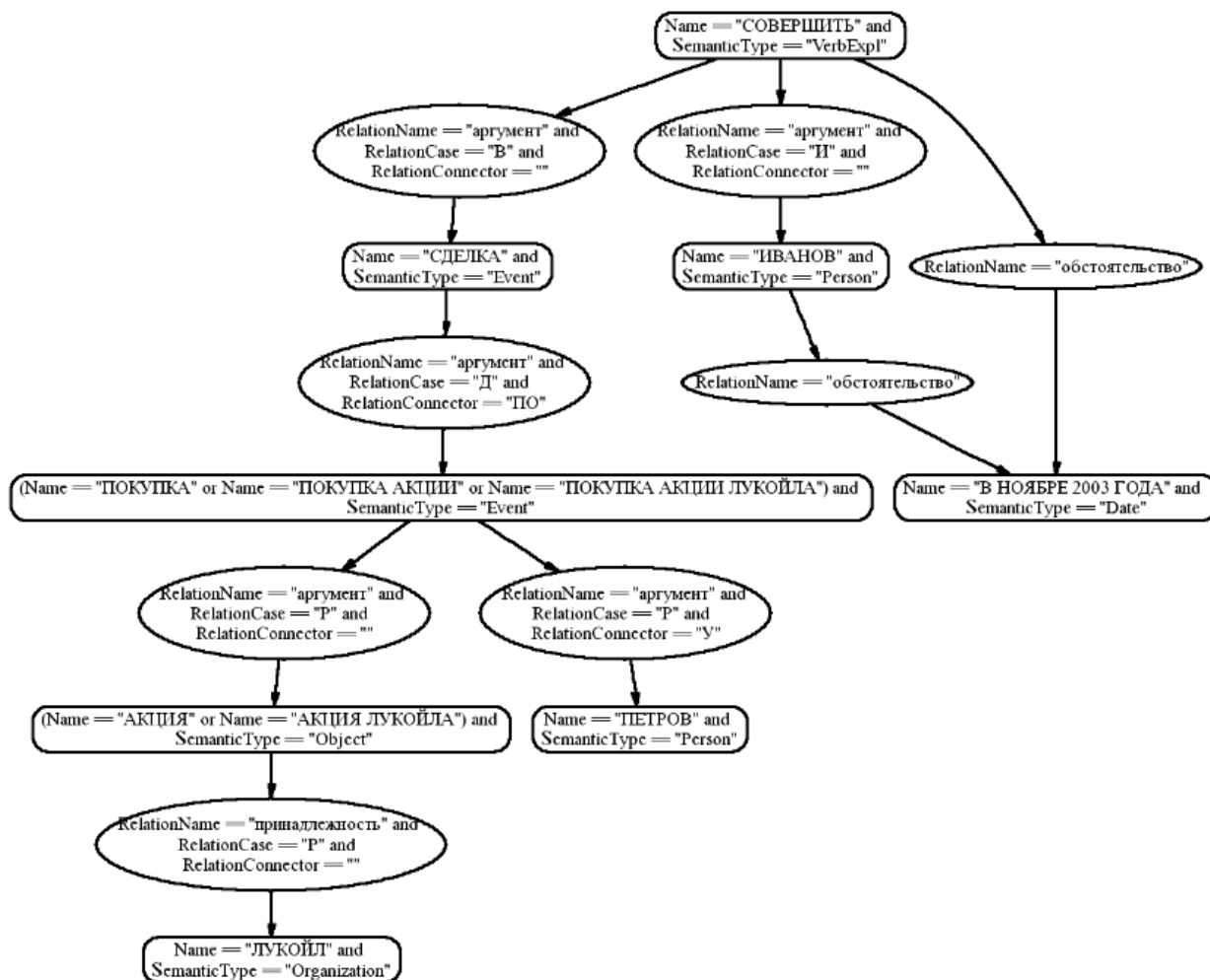
Основная полезная информация, которую можно получить по каждой сущности, следующая:

- Имя сущности – текстовая строка в нормализованной форме или идентификатор известного заданного объекта (организации, персоны). Имя сущности может применяться для ее идентификации;
- Категория, к которой относится сущность. Перечень стандартных категорий приведен в разделе «[Приложение 4. Типы выделяемых сущностей](#)». Дополнительно существует возможность определения собственных категорий.
- Грамматические характеристики (для каждого упоминания сущности в тексте) – часть речи, род, число, падеж, лицо и проч.;
- Семантические атрибуты – например, одушевленность, предметность, событийность и пр.;
- Структурные атрибуты – например, для персоны можно получить ее фамилию, имя и отчество, для даты – день, месяц, год и т.п.;
- Набор ссылок на все предложения текста, содержащие упоминания данной сущности, и ссылок на упоминания сущности в самом тексте. Информация о границах предложений, в которых упоминались сущности, позволяет отобрать из текста тематические рефераты по требуемым сущностям. Кроме того, информация о совместной встречаемости сущностей в предложениях может служить критерием наличия между ними некой смысловой или ассоциативной связи. Информация об упоминаниях необходима для подсветки сущностей в тексте документа;
- Коммуникативные характеристики (для каждого упоминания сущности в тексте) – набор признаков, характеризующих коммуникативную позицию сущности в предложении – член предложения (подлежащее, сказуемое, дополнение и пр.) и тип клаузы (придаточное предложение, причастный оборот, в скобках и т.п.). Такие признаки позволяют отразить способ представления автором ситуации, описываемой в предложении: какие из ее элементов находятся в фокусе его внимания, а какие излагаются как фоновые, как преподносится ситуация: как известный факт, как произошедшее событие или как нечто предполагаемое. Так, данный показатель позволяет отобрать ключевые сущности текста и сформировать его семантический портрет для автоматической классификации или автоматического реферирования.

Семантическая сеть

Семантическая сеть содержит все сущности, упоминавшиеся в тексте: наименования предметов и лиц, действий и признаков, связанные различными типами синтактико-семантических связей. Элементарная сеть представляет результат синтаксического анализа и дополнительных семантических трансформаций дерева синтаксических зависимостей между словами в отдельном предложении.

Программный интерфейс библиотеки позволяет получить для каждой сущности набор направленных связей, исходящих от нее к другим сущностям. Направление связи обычно соответствует направлению синтаксического подчинения (для равноправных однородных членов предложения пара одинаковых направленных связей идет в обе стороны).



Приведен пример структуры семантической сети, полученной в результате анализа предложения: *В ноябре 2003 года Ивановым была совершена сделка по покупке акций Лукойла у Петрова.* Графическое изображение семантической сети получено из программы **RCO Fact Tuner**.

Связи в сети имеют набор следующих атрибутов:

- **FX_ATTR_RELATIONTYPE** – тип синтактико-семантической связи между узлами, например, *аргумент*, *признак*, *принадлежность*.
- **FX_ATTR_RELATIONROLE** – семантическая роль, определенная для связей предиката с аргументом, получаемая обычно из словаря моделей управления, например *субъект*, *объект*, *инструмент*.
- **FX_ATTR_RELATIONCASE**, **FX_ATTR_RELATIONCONNECTOR** – семантический падеж и коннектор (предлог, союз), при помощи которых устанавливается связь (обычно предиката с его аргументом). Представляют альтернативу семантической роли, так как

роль не всегда может быть установлена однозначно. Один и тот же семантический падеж соответствует различным грамматическими падежами в зависимости от построения фразы. Например, семантический именительный субъекта действия и винительный объекта соответствуют одноименным грамматическим падежам в активном залоге (*программист написал программу*), а в пассивном выражаются грамматическим творительным и именительным соответственно (*программа написана программистом*).

- Представление содержания текста в форме семантической сети позволяет абстрагироваться от многих особенностей его коммуникативной организации. Такая сеть инвариантна к синтаксической структуре предложений и порядку слов с точностью до структуры пропозиции, выбранной автором для описания ситуации. Например, конструкциям *Иванов купил акции* и *акциях, купленных Ивановым* будут соответствовать одинаковые сети. В то же время пропозициям вида *Иванов становится покупателем акций Лукойла* и *покупка акций Лукойла – дело рук Иванова* будут соответствовать иные сети. Вследствие этого семантическая сеть является промежуточным уровнем представления между собственно семантической схемой ситуации и ее языковым описанием.

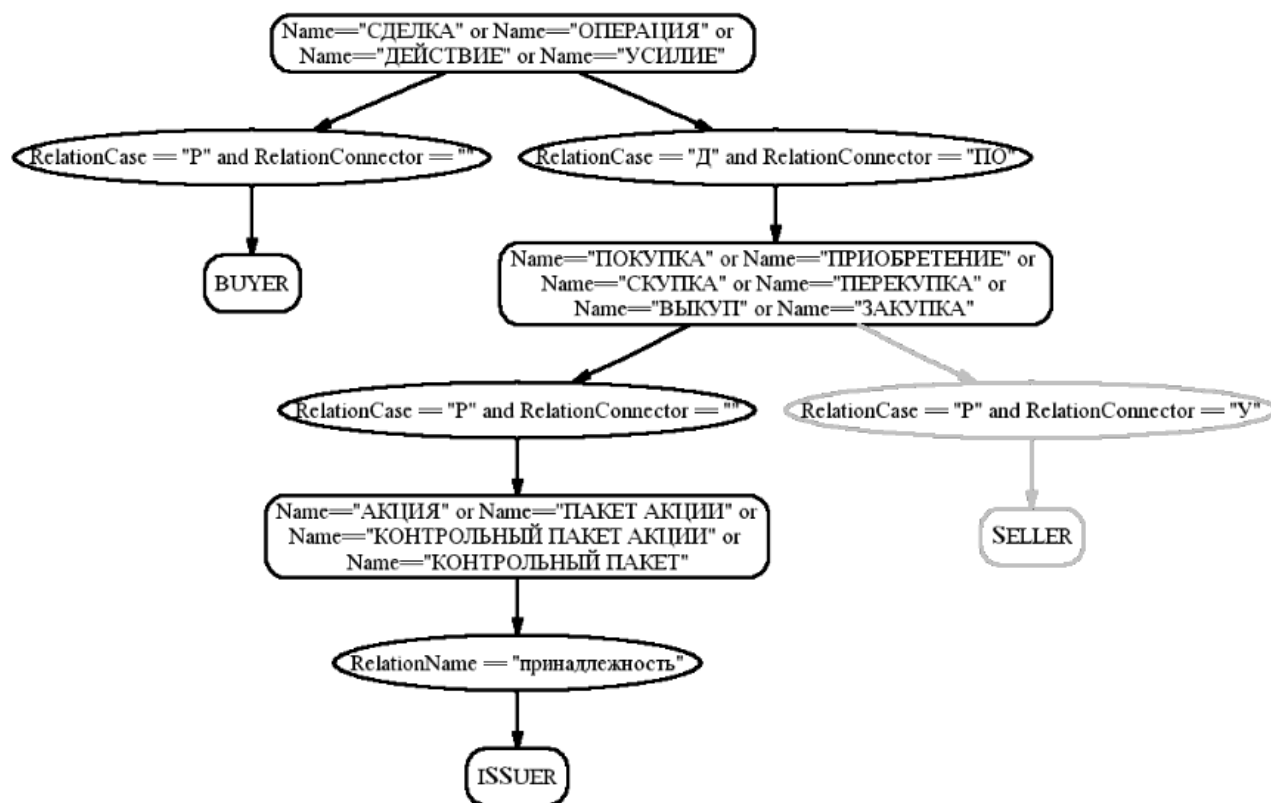
Фреймы и поиск ситуаций

Средства поиска фактов позволяют найти в тексте описание ситуаций в соответствии с заданными схемами, например, *встреча* или *покупка акций*, и извлечь требуемую информацию, связанную с ситуациями – имена задействованных участников-фигурантов. Поиск ситуаций производится по семантической сети, описанной в разделе выше.

Логическая схема ситуации в традиционной терминологии искусственного интеллекта называется фреймом. Фрейм имеет имя, которое идентифицирует класс описываемых им ситуаций, а также содержит слоты, которые имеют свои имена, идентифицирующие роли участников ситуации. Для конкретной ситуации, описанной в тексте, часть слотов может быть заполнена именами ее конкретных участников, упомянутых в тексте (*покупатель=Иванов, продавец=?, эмитент акций=Лукойл, количество акций=10%, сумма сделки=?, дата=2010*).

Модель фрейма задается множеством семантических шаблонов, каждый из которых описывает множество изоморфных семантических сетей, соответствующих некоторому типовому способу описания ситуации в тексте.

На рисунке ниже приведен пример шаблона, покрывающего множество описаний ситуации в форме пропозиции вида *Покупатель совершает действие по приобретению у продавца акций предприятия*.



В шаблоне на этом рисунке три узла, обозначенные метками *BUYER*, *ISSUER* и *SELLER*, представляют возможных фигурантов ситуации *покупка акций* в ролях *покупатель*, *эмитент* и *продавец* соответственно. Узел *SELLER* вместе с идущей к нему связью помечен необязательным, так как продавец может не указываться в тексте, а именно пара *покупатель-эмитент* представляет основной интерес для ситуаций класса покупки акций.

Основными элементами шаблона являются:

- целевые фигуранты – узлы сети, которые соответствуют интересующим сущностям, выделенным в тексте. Например, в ситуации покупки акций потенциально присутствуют фигуранты с ролями *продавец*, *покупатель* и *эмитент акций*, которые могут быть представлены в тексте самыми разными сущностями. Целевые фигуранты фактически

представляют собой переменные шаблона – незаполненные слоты фрейма, значения которых можно будет установить после приложения шаблона к тексту.

- вспомогательные узлы – узлы сети с заданными ограничениями на атрибуты, которые позволяют распознать присутствие описания ситуации в тексте. Обычно они соответствуют наименованию ситуации (*покупка, покупать, приобретать*) или именам обязательных фигурантов, идентифицирующим ее (тип товара: *акция, контрольный пакет*).
- схема ситуации – набор связей между фигурантами и вспомогательными элементами с заданными ограничениями на атрибуты связей. Схема ситуации обычно соответствует набору связей в семантической сети простого неосложненного предложения, свободного от дополнительных участников, обстоятельств, определений и прочего.

Поиск ситуации есть поиск в семантической сети текста такой подсети, которая изоморфна одному из шаблонов. Если подсеть найдена, ситуация считается распознанной, после чего производится извлечение сущностей и их маркировка ролями, заданными в соответствующих узлах шаблона. Таким образом, результатом поиска является установленный класс ситуации и набор указателей на сущности семантической сети с указанием соответствующих им ролей из шаблона – фрейм известного класса с полностью или частично заполненными слотами.

Дополнительно в шаблоне могут присутствовать необязательные узлы, которые соответствуют дополнительным целевым участникам. После нахождения изоморфизма производится поиск необязательных узлов, и соответствующие им сущности добавляются к числу найденных фигурантов.

Шаблоны задаются на формальном языке описания графов, который позволяет задать структуру сети и наложить ограничения на атрибуты узлов и связей в виде логических выражений. Для удобства настройки шаблонов используется приложение с графическим интерфейсом **RCO Fact Tuner**, позволяющее построить сеть на основе типовой фразы русского языка. После добавления требуемых ограничений на узлы сети, указания обязательных, необязательных, порожденных элементов и ролей участников ситуации шаблон сохраняется в нужном формализме, готовый для загрузки нашей библиотекой.

Отметим, что в ряде случаев синтаксический анализатор не может установить связь между участниками ситуации, опираясь на заложенные в него общие правила русской грамматики (*Соучредитель ООО «Ромашка» (20%) – ЗАО «Сирень»*). В этом случае синтаксико-семантические связи между некоторыми участниками семантической сети могут отсутствовать. Чтобы решить такие проблемы, в семантическую сеть могут добавляться связи особого типа (атрибут `FX_ATTR_RELATIONTYPE` имеет значение *next*), которые просто выстраивают в цепочку идущие друг за другом в предложении слова и знаки препинания, «перепрыгивая» через синтаксически подчиненные слова в именных группах, что позволяет писать шаблоны, инвариантные к длине словосочетаний. В итоге совокупность узлов сети всегда представляет собой полносвязный граф, что позволяет написать шаблон, извлекающий из приведенного примера и соучредителя, и его долю.

Помимо механизма выделения фреймов на основе настраиваемых шаблонов, библиотека содержит встроенные алгоритмы для распознавания некоторых особенно сложных ситуаций, полное и точное выявление которых средствами одних только лингвистических шаблонов затруднительно. К таковым, в частности, относятся прямая и косвенная речь и подобные им речемыслительные акты, которые выделяются как фреймы классов *DirectSpeech* и *IndirectSpeech*, имеющие слоты с ролями *Author* (*Вася, депутат*), *Action* (*заявить, подумать*), *Begin* (*–, “, что*), *End* (*.,!*).

Дополнительно настройки библиотеки позволяют фильтровать описания ситуаций, соответствующих реальным событиям или фактам. Из числа найденных фреймов и сущностей в их слотах могут быть исключены те фреймы, которые соответствуют нереальным ситуациям, и те сущности, которые реального участия в ситуации не принимали. Это производится на основании лингвистического контекста описания ситуации – наличия показателей отрицания и нереальности.

Настройка лингвистического обеспечения

Область применения приложений, использующих библиотеку **RCO FX Ru**, может потребовать расширения и настройки лингвистического обеспечения.

Информация по настройке различных составляющих лингвистического обеспечения представлена в следующих документах:

- **RCO Morph** – пакет настройки морфологического словаря
- **RCO Ling** – пакет настройки семантических словарей
- **RCO Pattern Extractor** – компонент выделения конструкций в тексте
- **RCO Fact Tuner** – программа настройки семантических шаблонов
- Руководство администратора **RCO** – настройка описаний объектов

Порядок работы с библиотекой

Инициализация

Инициализация библиотеки происходит автоматически при загрузке. В ходе инициализации библиотека пытается найти и разобрать xml-файл **fx.ini**. Он должен располагаться в одной папке с модулем библиотеки, загружаемым внешним приложением.

Файл **fx.ini** имеет следующий формат.

```
<?xml version="1.0" encoding="windows-1252"?>
<fx-configuration xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:noNamespaceSchemaLocation="le-config.xsd">
<logger>
    <file setpid="true">ИмяСоздаваемогоФайлаЖурнала</file>
    <level>УровеньПодробностиЗаписейВФайлеЖурнала</level>
</logger>
    <param id="print-parsing-report" value="off"/>
</fx-configuration>
```

Здесь *ИмяСоздаваемогоФайлаЖурнала* может быть, к примеру, **fx.log**, а *УровеньПодробностиЗаписейВФайлеЖурнала* принимать одно из следующих значений:

- *none* – никаких сообщений не выдается;
- *error* – выдаются сообщения об ошибках;
- *warning* – выдаются сообщения об ошибках и предупреждения;
- *info* – выдаются сообщения об ошибках, предупреждения и некоторая дополнительная информация;
- *verbose* – выдается наиболее полная информация обо всех событиях, происходящих при работе с библиотекой.

Параметру `setpid` при необходимости ведения лог-файла для каждого потока присваивается значение *true*, в противном случае – *false*. К имени файла (перед его расширением) добавляется символ '_' и идентификатор процесса (в формате *x16*), получаемый посредством вызова `GetCurrentProcessId()` – например, *fx_log_000004C4.txt*.

Параметр `print-parsing-report` может принимать значения *off* и *on*.

При установке значения *on* в ходе работы библиотеки в папке ее размещения создаются несколько служебных файлов, в которые записывается подробная информация о результатах разбора каждого предложения в тексте. Не рекомендуется устанавливать значение *on* без необходимости, так как при этом работа библиотеки замедляется.

В случае если библиотеке при загрузке не удалось обнаружить файл **fx.ini**, ее работа будет продолжена в нормальном режиме, однако без выдачи информации в файл журнала и без распечатки вспомогательных результатов анализа текста.

Типовой сценарий работы

Ниже описана стандартная последовательность ключевых действий, используемая при работе с библиотекой.

1. Создание анализатора текста с заданными параметрами и его инициализация лингвистическими ресурсами – вызов функции **FXCreateContext** и получение дескриптора созданного анализатора *hContext*. Предварительно необходимо считать содержимое конфигурационного файла и создать объект (менеджер ресурсов) для передачи в качестве параметров функции.
2. Анализ текста – вызов функции **FXProcessText** для дескриптора *hContext* и получение дескриптора результатов анализа *hResult*.
3. Считывание и обработка результатов анализа текста (например, сохранение требуемой информации в структурах БД). Для дескриптора *hResult* – вызов функций из группы, описанной в разделе «[Функции для работы с результатом анализа текста](#)».
4. Сброс результатов анализа текста и освобождение ресурсов – вызов функции **FXDestroyResult** для дескриптора *hResult*.
5. Переход к анализу следующего текста (шаг 2) или завершение работы с библиотекой (следующий шаг).
6. Уничтожение анализатора текста и освобождение ресурсов – вызов функции **FXDestroyContext** для дескриптора *hContext*.

Для получения подробной информации о реализации данного сценария обратитесь к исходному коду тестового приложения, входящего в комплект поставки библиотеки.

Интерфейс библиотеки

Интерфейс для доступа к лингвистическим ресурсам

Для обеспечения доступа к внешним ресурсам, необходимым для работы библиотеки, и прежде всего для загрузки лингвистических ресурсов (словарей, правил, описаний объектов, шаблонов) библиотека использует специальный внешний объект – менеджер ресурсов, который поддерживает программный интерфейс, описываемый типом **FX_RESOURCE_MANAGER**.

Программист-пользователь библиотеки может обеспечить различную реализацию объектов типа **FX_RESOURCE_MANAGER** для загрузки ресурсов из файловой системы или из базы данных.

Тип **FX_RESOURCE_MANAGER** определен следующим образом:

```
struct FX_RESOURCE_MANAGER
{
    virtual FX_RESOURCE_HANDLE Open( const char* szResourceId ) = 0;
    virtual void Close( FX_RESOURCE_HANDLE hResource ) = 0;
    virtual bool Read( FX_RESOURCE_HANDLE hResource, void*
        pBuffer, unsigned long ulNumberOfBytesToRead, unsigned
        long* pulNumberOfBytesRead ) = 0;
};
```

Ниже описаны функции из структуры **FX_RESOURCE_MANAGER**, которые программисту необходимо реализовать для создания менеджера ресурсов, используемого библиотекой.

В приложении к библиотеке – тестовом приложении test.cpp – уже включена простая реализация менеджера, считывающего ресурсы из одноименных файлов.

Open

Функция открывает заданный ресурс на чтение.

```
FX_RESOURCE_HANDLE Open(
    const char* szResourceId // дескриптор ресурса
);
```

Параметры

szResourceId

[вх] Дескриптор ресурса (путь к файлу или идентификатор в БД).

Возвращаемое значение

При успешном открытии ресурса функция должна возвращать дескриптор, не равный *NULL* и используемый в дальнейшем для чтения ресурса посредством вызова [Read](#).

После завершения работы с ресурсом его следует закрыть при помощи вызова [Close](#).

Close

Функция закрывает заданный ресурс.

```
void Close(  
FX_RESOURCE_HANDLE hResource // дескриптор ресурса  
);
```

Параметры

hResource

[вх] Дескриптор подлежащего закрытию ресурса, получаемый вызовом **Open**.

Read

Функция считывает порцию данных из заданного ресурса и передвигает внутренний указатель на следующие доступные данные.

```
bool Read(  
FX_RESOURCE_HANDLE hResource // дескриптор ресурса  
void* pBuffer // буфер  
unsigned long ulNumberOfBytesToRead // число требуемых байтов  
unsigned long* pulNumberOfBytesRead // число полученных байтов  
);
```

Параметры

hResource

[вх] Дескриптор подлежащего закрытию ресурса, получаемый вызовом **Open**.

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

В случае успешного чтения ресурса функция должна возвращать *true*, иначе – *false*.

Интерфейс для добавления и переопределения внешних функций

Функции, определенные в пользовательском приложении, последовательно, начиная с первой, исполняются в библиотеке **RCO FX Ru** путем передачи указателя на них посредством менеджера внешних функций, интерфейс которого описывается типом **FX_EXTERNAL_FUNCTION_MANAGER**. Внешние функции могут быть применены в правилах CAPE. CAPE – мощный язык описания конструкций текста, позволяющий оперировать как формальными особенностями написания текста, используя, в частности, язык регулярных выражений, так и всеми грамматическими атрибутами слов – частью речи, родом, числом, падежом и т.д. (см. «**RCO Pattern Extractor – компонент выделения конструкций в тексте. Руководство администратора**»).

Тип **FX_EXTERNAL_FUNCTION_MANAGER** определен следующим образом

```
struct FX_EXTERNAL_FUNCTION_MANAGER
{
virtual FX_EXTERNAL_FUNCTION_HANDLE FirstFunction() = 0;
virtual FX_EXTERNAL_FUNCTION_HANDLE
    NextFunction(FX_EXTERNAL_FUNCTION_HANDLE) = 0;
virtual void* GetAddress(FX_EXTERNAL_FUNCTION_HANDLE) = 0;
virtual void* GetContext(FX_EXTERNAL_FUNCTION_HANDLE) = 0;
virtual const char* GetName(FX_EXTERNAL_FUNCTION_HANDLE) = 0;
virtual long GetType(FX_EXTERNAL_FUNCTION_HANDLE) = 0;
}
```

FirstFunction

Вызывает первую внешнюю функцию.

```
FX_EXTERNAL_FUNCTION_HANDLE FirstFunction()
```

Возвращаемое значение

При успешном срабатывании функция возвращает объект для работы с остальными внешними функциями.

NextFunction

Вызывает следующую внешнюю функцию.

```
FX_EXTERNAL_FUNCTION_HANDLE NextFunction(FX_EXTERNAL_FUNCTION_HANDLE)
```

GetAddress

Запрашивает адрес внешней функции в памяти.

```
void* GetAddress(FX_EXTERNAL_FUNCTION_HANDLE) // адрес функции в
    памяти
```

GetAddress

[вых] Указатель на область памяти, в которой хранится внешняя функция.

GetContext

Запрашивает контекст вызова функции.

```
void* GetContext(FX_EXTERNAL_FUNCTION_HANDLE) // контекст вызова
    функции
```

GetContext

[вых] Указатель на объект, используемый функцией и передаваемый ей при инициализации.

GetName

Запрашивает имя внешней функции.

```
const char* GetName(FX_EXTERNAL_FUNCTION_HANDLE) // имя функции
```

GetName

[вых] Указатель на имя внешней функции.

GetType

Запрашивает целочисленное значение, определяющее тип функции.

`long GetType(FX_EXTERNAL_FUNCTION_HANDLE) // тип функции`

GetType

[вых] Указатель на тип внешней функции.

Функции для работы с анализатором текста

FXCreateContext

В соответствии с заданной конфигурацией функция загружает лингвистические ресурсы и создает новый объект – анализатор текста, через обращение к которому впоследствии будет осуществляться анализ текстовых документов.

```
FX_ERROR FXCreateContext(  
    const wchar_t* szConfig // текст с конфигурацией анализатора в  
        формате XML  
    FX_RESOURCE_MANAGER* pResourceManager // менеджер ресурсов  
    FX_HANDLE* phContext // анализатор текста  
    FX_EXTERNAL_FUNCTION_MANAGER* phExtFunctionManager // менеджер  
        внешних функций  
);
```

Параметры

szConfig

[вх] Строка в кодировке Unicode, содержащая описание создаваемой конфигурации анализатора текста в XML-формате (подробнее – в разделе «[Приложение 1. Формат конфигурационного файла](#)»).

pResourceManager

[вх] Указатель на объект – менеджер ресурсов, который обеспечивает программный интерфейс для загрузки лингвистических ресурсов, используемых библиотекой. Функция **FXCreateContext** будет вызывать функции данного интерфейса, подробно описанные в разделе «[Интерфейс для доступа к лингвистическим ресурсам](#)». При передаче нулевого указателя используется менеджер ресурсов по умолчанию, интерпретирующий идентификатор как путь к файлу на локальном диске.

phContext

[вых] Указатель на переменную, которая получает дескриптор созданного объекта – анализатора текста.

phExtFunctionManager

[вх] Указатель на объект менеджер внешних функций.

Возвращаемое значение

В случае успешного создания анализатора текста функция возвращает `FX_SUCCESS`, иначе – один из кодов ошибки `FX_ERROR`. В случае неудачи смотрите описание ошибки в [файле журнала](#).

Замечание. При завершении работы с библиотекой необходимо вызвать **FXDestroyContext** для освобождения ресурсов.

FXDestroyContext

Функция удаляет объект – анализатор текста и освобождает используемые им лингвистические ресурсы.

```
FX_ERROR FXDestroyContext(  
FX_HANDLE hContext // анализатор текста  
);
```

Параметры

hContext

[вх] Дескриптор объекта, подлежащего удалению.

Возвращаемое значение

В случае успешного удаления объекта функция возвращает FX_SUCCESS, иначе – один из кодов ошибки FX_ERROR.

FXProcessText

Функция анализирует текст и создает объект, через который впоследствии можно получить доступ ко всем результатам анализа.

```
FX_ERROR FXProcessText (  
    FX_HANDLE hContext // анализатор текста  
    const wchar_t* szText // текст для анализа  
    const wchar_t* szTextID // идентификатор текста  
    FX_TEXT_FORMAT eFormat // формат текста  
    FX_HANDLE* phResult // результат анализа  
);
```

Параметры

hContext

[вх] Дескриптор анализатора, посредством которого выполняется анализ текста.

szText

[вх] Текст для анализа в кодировке Unicode.

szTextID

[вх] Идентификатор текста в кодировке Unicode. Отображается при выдаче сообщений в ходе анализа текста.

eFormat

[вх] Формат текста. Допустимы следующие значения:

- `FX_FORMAT_AUTO` – формат текста (html или просто текст) будет определен автоматически по началу документа.
- `FX_FORMAT_HTML` – текст будет анализироваться с очисткой от html-тегов и использованием их при выделении блоков, параграфов и предложений.
- `FX_FORMAT_PLAIN` – текст будет анализироваться без очистки от html-тегов с использованием правил выделения параграфов и предложений, включенных в конфигурационном файле.

phResult

[вых] Указатель на переменную, которая получает дескриптор результата анализа текста.

Возвращаемое значение

В случае успешного завершения функция возвращает `FX_SUCCESS`, иначе – один из кодов ошибки `FX_ERROR`.

Замечание. Начало работы с результатом анализа текста через полученный дескриптор описано в разделе «[Функции для работы с результатом анализа текста](#)».

FXDestroyResult

Функция освобождает ресурсы, связанные с результатом анализа текста.

```
FX_ERROR FXDestroyResult(  
FX_HANDLE hResult // результат анализа текста  
);
```

Параметры

hResult

[вх] Дескриптор результатов анализа текста, подлежащих удалению.

Возвращаемое значение

В случае успешного завершения функция возвращает FX_SUCCESS, иначе – один из кодов ошибки FX_ERROR.

Функции для работы с результатом анализа текста

Интерфейс библиотеки приспособлен для обеспечения единообразного доступа к различным типам информации о результатах анализа текста. С этой целью все виды получаемой информации разделены на два класса:

- Объекты – сложные логические информационные единицы, которые могут обладать структурой и/или атрибутами. Основные типы объектов, используемых для описания результатов анализа текста: предложения, сущности в предложении, связи между сущностями, фреймы, слоты фрейма. Под наличием у объекта структуры понимается тот факт, что некоторые объекты могут иметь другие объекты, входящие в их состав, а также связи с другими объектами. Например, в объекте-предложении содержатся упомянутые в нем объекты-сущности, в объекте-фрейме – его объекты-слоты; каждый объект-сущность связан с объектом-связью (синтактико-семантической), а каждый объект-связь в свою очередь опять связан с объектом-сущностью. Каждый конкретный объект описывается своим уникальным идентификатором. Обход объектов осуществляется при помощи вызовов функций **FXGetFirst** и **FXGetNext**, которые получают идентификаторы одних объектов и возвращают идентификаторы других объектов, входящих в их состав или связанных с ними.
- Атрибуты объектов – простые информационные единицы, описывающие характеристики объектов и имеющие один из простых типов: **FX_ATTRTYPE_BOOL**, **FX_ATTRTYPE_LONG**, **FX_ATTRTYPE_STRING**, **FX_ATTRTYPE_IDARRAY**. Каждый атрибут имеет имя и значение. Объекты различных типов имеют атрибуты с различными именами, например, для объекта-предложения можно получить значения атрибутов с именами **FX_ATTR_SENTENCE_OFFSET** и **FX_ATTR_SENTENCE_LENGTH** – начало и длину предложения в тексте; для объектов-сущностей можно получить значения атрибутов **FX_ATTR_MORPH_SPEECHPART** и **FX_ATTR_ENTITY_PARTOFSSENTENCE** – части речи и члена предложения и т.п. Атрибуты объекта можно получить по его идентификатору, запросив количество и значение атрибутов с требуемым именем при помощи вызовов функций **FXGetAttributeValueCount** и **FXGetAttributeValue**.

FXGetFirst

Функция позволяет получить идентификатор первого объекта требуемого типа, выделенного в тексте. В зависимости от типа объекта его идентификатор может быть получен либо через идентификаторы объектов других типов, либо непосредственно через дескриптор результата анализа текста.

```
FX_ERROR FXGetFirst(
FX_HANDLE hResult, // дескриптор результата анализа
const FX_ID* pidSource, // указатель на идентификатор объекта
unsigned long InfoType, // тип запрашиваемой информации
FX_ID* pidTarget // указатель на результат
);
```

Параметры

hResult

[вх] Дескриптор, полученный в результате вызова **FXProcessText**.

pidSource

[вх] Указатель на переменную, содержащую идентификатор объекта, от которого осуществляется доступ к запрашиваемому объекту, или **NULL**.

InfoType

[вх] Тип запрашиваемой информации. Может принимать одно из следующих значений:

Первый объект в последовательности (последующие могут быть получены при помощи вызова FXGetNext)	
<i>FX_INFO_SENTENCE</i>	Предложение текста
<i>FX_INFO_ENTITY</i>	Сущность в заданном предложении текста
<i>FX_INFO_RELATIONSHIP</i>	Связь, исходящая из заданной сущности
<i>FX_INFO_FRAME</i>	Фрейм, выделенный в тексте
<i>FX_INFO_FRAME_SLOT</i>	Слот в заданном фрейме
<i>FX_INFO_CONTEXT</i>	Актуальный контекст текста
Единичный объект	
<i>FX_INFO_TARGET_ENTITY_FOR_RELATIONSHIP</i>	Сущность, в которую входит заданная связь
<i>FX_INFO_CONNECTOR_ENTYTY_FOR_RELATIONSHIP</i>	Сущность-коннектор, посредством которой установлена заданная связь
<i>FX_INFO_ENTITY_FOR_FRAME_SLOT</i>	Сущность, соответствующая заданному слоту фрейма
<i>FX_INFO_SENTENCE_FOR_ENTITY</i>	Предложение, в которое входит заданная сущность
<i>FX_INFO_ENTITY_FOR_CONTEXT</i>	Сущность, представляющая заданный контекст
<i>FX_INFO_ENTITY_COREFERENT</i>	Сущность, кореферентная заданной сущности

pidTarget

[вых] Указатель на переменную, которая возвращает идентификатор запрашиваемого объекта.

Возвращаемое значение

В случае наличия хотя бы одного запрошенного объекта функция возвращает *FX_SUCCESS*, иначе – *FX_WARN_NO_MORE_RESULTS*.

Замечание. Для некоторых типов запрошенной информации идентификаторы последующих объектов в последовательности можно получить при помощи вызова функции **FXGetNext**.

FXGetNext

Функция предназначена для обхода последовательности упорядоченных объектов и позволяет получить идентификатор очередного объекта требуемого типа, выделенного в тексте. В зависимости от типа объекта его идентификатор может быть получен либо через идентификаторы объектов других типов, либо непосредственно через дескриптор результата анализа текста. Идентификатор первого объекта в последовательности можно получить при помощи вызова функции [FXGetFirst](#).

```
FX_ERROR FXGetNext(
    FX_HANDLE hResult, // дескриптор результата анализа
    const FX_ID* pidSource, // указатель на идентификатор объекта
    unsigned_long InfoType, // тип запрашиваемой информации
    FX_ID* pidTarget // указатель на результат
);
```

Параметры

hResult

[вх] Дескриптор, полученный в результате вызова [FXProcessText](#).

pidSource

[вх] Указатель на переменную, содержащую идентификатор объекта, от которого осуществляется доступ к последовательности запрашиваемых объектов, или *NULL*.

InfoType

[вх] Тип запрашиваемой информации. Может принимать одно из следующих значений:

<i>FX_INFO_SENTENCE</i>	Предложение текста
<i>FX_INFO_ENTITY</i>	Сущность в заданном предложении текста
<i>FX_INFO_RELATIONSHIP</i>	Связь, исходящая из заданной сущности
<i>FX_INFO_FRAME</i>	Фрейм, выделенный в тексте
<i>FX_INFO_FRAME_SLOT</i>	Слот в заданном фрейме
<i>FX_INFO_CONTEXT</i>	Актуальный контекст текста

pidTarget

[вх/вых] Указатель на переменную, которая при вызове функции содержит идентификатор текущего объекта последовательности, полученный в результате вызова [FXGetFirst](#) или предыдущего вызова [FXGetNext](#). После вызова функции данная переменная будет содержать идентификатор объекта, следующего в последовательности за заданным.

Возвращаемое значение

В случае наличия хотя бы одного запрошенного объекта функция возвращает *FX_SUCCESS*, иначе – *FX_WARN_NO_MORE_RESULTS*.

Замечание. Функция должна вызываться в цикле для получения информации обо всех объектах в последовательности. Доступ к первому объекту можно получить при помощи вызова функции [FXGetFirst](#).

FXGetAttributeValueCount

Функция позволяет получить количество значений заданного атрибута объекта (предложения, сущности, связи и т.п.) для последующего считывания этих значений при помощи вызова [FXGetAttributeValue](#). Описание доступных атрибутов приведено в разделе «[Описание атрибутов](#)».

```
FX_ERROR FXGetAttributeValueCount(  
FX_HANDLE hResult, // дескриптор результата анализа  
const FX_ID* pidSource // идентификатор объекта  
unsigned long AttrID, // идентификатор атрибута  
unsigned short *puValueCount // количество значений атрибута  
);
```

Параметры

hResult

[вх] Дескриптор, полученный в результате вызова [FXProcessText](#).

pidSource

[вх] Указатель на переменную, содержащую идентификатор объекта, для которого запрашивается количество значений атрибута.

AttrID

[вх] Идентификатор атрибута, для которого запрашивается количество значений. Возможные значения указаны в разделе «[Описание атрибутов](#)».

puValueCount

[вых] Указатель на переменную, которая будет содержать количество значений атрибута.

Возвращаемое значение

При наличии у сущности запрашиваемого атрибута и хотя бы одного его значения функция возвращает `FX_SUCCESS`. В отсутствие атрибута – `FX_ERR_NO_SUCH_ATTR`. В случае ошибки функция возвращает один из кодов ошибки `FX_ERROR`.

FXGetAttributeValue

Функция возвращает значение заданного атрибута объекта (предложения, сущности, связи и т.п.). Описание доступных атрибутов приведено в разделе «[Описание атрибутов](#)».

```
FX_ERROR FXGetAttributeValue(  
FX_HANDLE hResult, // дескриптор результата анализа  
const FX_ID* pidSource // идентификатор объекта  
unsigned long AttrID, // идентификатор атрибута  
unsigned short uValueIndex // индекс значения атрибута  
unsigned long* pulBufferSize // размер буфера  
void* pBuffer // буфер для получения значения  
);
```

Параметры

hResult

[вх] Дескриптор, полученный в результате вызова **FXProcessText**.

pidSource

[вх] Указатель на переменную, содержащую идентификатор объекта, для которого запрашивается значение атрибута.

AttrID

[вх] Идентификатор атрибута, для которого запрашивается значение. Возможные значения указаны в разделе «[Описание атрибутов](#)».

uValueIndex

[вх] Индекс запрашиваемого значения атрибута. Начинается с нуля. Количество доступных значений можно получить путем вызова **FXGetAttributeValueCount**.

pulBufferSize

[вх/вых] Указатель на переменную, которая при вызове функции содержит размер в байтах передаваемого буфера *pBuffer*. После вызова функции значение переменной будет равно реальному количеству байт, необходимых для размещения значения атрибута, включая завершающий ноль для строковых значений.

pBuffer

[вых] Указатель на буфер, в который будет записано значение атрибута.

Может быть равен *NULL*, тогда функция вычисляет требуемый размер буфера и возвращает его в переменной, на которую указывает *pulBufferSize*. Если указатель не равен *NULL*, функция записывает значение атрибута в буфер, однако не более чем переданное ей в *pulBufferSize* количество байт.

Возвращаемое значение

Функция возвращает *FX_SUCCESS*, если переданное значение *pBuffer* было равно *NULL* и размер требуемого буфера был успешно вычислен.

Функция возвращает *FX_SUCCESS*, если значение было успешно скопировано в переданный *pBuffer*.

Функция возвращает *FX_WARN_DATA_TRUNCATED* при запросе строкового значения атрибута, если размер переданного буфера *pBuffer* оказался недостаточен для копирования всего значения. В этом случае в *pBuffer* будет скопирована часть строкового значения длины *pulBufferSize*, а возвращенное значение в *pulBufferSize* будет равно размеру буфера, необходимому для получения полной строки.

Функция возвращает *FX_ERR_BUFFER_TOO_SMALL* при запросе нестрокового значения атрибута, если размер *pBuffer* оказался меньше размера запрашиваемого значения.

Функция возвращает `FX_ERR_NO_SUCH_ATTR` в случае отсутствия запрашиваемого атрибута у сущности.

Функция возвращает `FX_ERR_NO_VALUE`, если у атрибута нет значения с таким индексом.

В случае прочих ошибок функция возвращает один из кодов ошибки `FX_ERROR`.

Замечание. Для нестроковых атрибутов функция ожидает получить буфер следующего размера:

для типа атрибута `FX_ATTRTYPE_BOOL` – не меньше `sizeof(bool)` байт;

для типа атрибута `FX_ATTRTYPE_LONG` – не меньше `sizeof(long)` байт;

для типа атрибута `FX_ATTRTYPE_FLOAT` – не меньше `sizeof(float)` байт.

Для атрибутов типа `FX_ATTRTYPE_STRING` функция копирует строковое значение в переданный буфер `pBuffer`, но не более чем $(*pulBufferSize/sizeof(wchar_t) - 1)$ символов, всегда завершая строку нулевым символом.

FXSetEntityNameGenerationMode

Функция позволяет установить требуемые параметры для генерации имен сущностей, вступающих в действие при всех последующих вызовах [FXGetAttributeValue](#) с параметром [FX_ATTR_ENTITY_NAME](#). Список возможных параметров приведен в разделе «[Приложение 5. Управление генерацией имен сущностей](#)».

```
FX_ERROR FXSetEntityNameGenerationMode(  
FX_HANDLE hResult, // дескриптор  
unsigned short idParam // идентификатор параметра  
long nValue, // значение параметра  
);
```

Параметры

hResult

[вх] Дескриптор, полученный в результате вызова [FXProcessText](#).

idParam

[вх] Идентификатор параметра, для которого устанавливается значение.

nValue

[вх] Значение параметра.

Возвращаемое значение

В случае ошибки функция возвращает один из кодов ошибки [FX_ERROR](#).

Замечание. Допускается многократное изменение значений параметров для получения различных вариантов имени одной и той же сущности. Любое изменение параметров может повлиять на количество генерируемых имен сущности, вследствие чего после каждого изменения параметров перед вызовом [FXGetEntityAttributeValue](#) необходимо повторно вызывать [FXGetAttributeValueCount](#).

FXGetEntityNameGenerationMode

Функция позволяет получить текущие значения параметров, используемые для генерации имен сущностей, установленные вызовом [FXSetEntityNameGenerationMode](#) или используемые изначально.

```
FX_ERROR FXSetEntityNameGenerationMode(  
FX_HANDLE hResult, // дескриптор  
unsigned short idParam // идентификатор параметра  
long* pnValue, // значение параметра  
);
```

Параметры

hResult

[вх] Дескриптор, полученный в результате вызова [FXProcessText](#).

idParam

[вх] Идентификатор параметра, для которого требуется получить значение.

pnValue

[вых] Указатель на переменную, в которую будет записано значение параметра.

Возвращаемое значение

В случае ошибки функция возвращает один из кодов ошибки [FX_ERROR](#).

Работа с сущностями в предложении: функции FXGetFirst, FXGetNext

Вызовы данной группы позволяют получить всю информацию о сущностях, упомянутых в тексте, и о предложениях, в которых они упоминаются. Типы выделяемых сущностей описаны в разделе «[Введение: выделение сущностей](#)».

С точки зрения организации программного интерфейса каждое упоминание сущности в тексте – это отдельная сущность с собственными атрибутами. Отождествить между собой различные упоминания одной сущности можно на основании совпадения их имен (см. атрибут FX_ATTR_ENTITY_NAME).

Доступ к любым атрибутам сущности (см. раздел «[Атрибуты сущности](#)») осуществляется через уникальный идентификатор. Его можно получить, последовательно перебирая все сущности, выделенные в предложении, в котором сущность упоминалась.

Получение первого предложения текста

```
FX_ERROR FXGetFirst(  
    hResult, // дескриптор результата анализа  
    NULL, // не используется  
    FX_INFO_SENTENCE, // тип запрашиваемой информации  
    &idSentence // указатель на результат  
);
```

Параметры

idSentence

[вых] Переменная, в которую будет записан идентификатор первого предложения в тексте (в случае, если функция возвратит FX_SUCCESS).

Получение очередного предложения текста

```
FX_ERROR FXGetNext(  
    hResult, // дескриптор результата анализа  
    NULL, // не используется  
    FX_INFO_SENTENCE, // тип запрашиваемой информации  
    &idSentence // указатель на результат  
);
```

Параметры

idSentence

[вх/вых] Переменная, которая при вызове функции должна содержать идентификатор предложения, полученный в результате вызова [FXGetFirst](#) или [FXGetNext](#). После вызова функции данная переменная будет содержать идентификатор следующего предложения в тексте (в случае, если функция возвратит FX_SUCCESS).

Получение первой сущности в предложении

```
FX_ERROR FXGetFirst(  
    hResult, // дескриптор результата анализа  
    &idSentence, // указатель на идентификатор предложения  
    FX_INFO_ENTITY, // тип запрашиваемой информации  
    &identity // указатель на результат  
);
```

Параметры

idSentence

[вх] Переменная должна содержать идентификатор того предложения, для которого запрашивается первая сущность.

idEntity

[вых] Переменная, в которую будет записан идентификатор первой сущности в предложении (в случае, если функция возвратит FX_SUCCESS).

Получение очередной сущности в предложении

```
FX_ERROR FXGetNext(  
    hResult, // дескриптор результата анализа  
    &idSentence, // указатель на идентификатор предложения  
    FX_INFO_ENTITY, // тип запрашиваемой информации  
    &identity // указатель на результат  
);
```

Параметры

idSentence

[вх] Переменная должна содержать идентификатор того предложения, для которого запрашивается очередная сущность.

idEntity

[вх/вых] Переменная, которая при вызове функции должна содержать идентификатор сущности в предложении *idSentence*, полученный в результате вызова **FXGetFirst** или **FXGetNext**. После вызова функции данная переменная будет содержать идентификатор следующей сущности в предложении (в случае, если функция возвратит FX_SUCCESS).

Получение предложения, содержащего сущность

```
FX_ERROR FXGetFirst(  
    hResult, // дескриптор результата анализа  
    &idEntity, // указатель на идентификатор сущности  
    FX_INFO_SENTENCE_FOR_ENTITY, // тип запрашиваемой информации  
    &idSentence // указатель на результат  
);
```

Параметры

idEntity

[вх] Переменная должна содержать идентификатор той сущности, для которой запрашивается содержащее ее предложение.

idSentence

[вых] Переменная для записи идентификатора предложения, содержащего сущность.

Замечание. Получение предложения от сущности может оказаться необходимым в случае, если идентификатор сущности был получен не от предложения, а от другого объекта – например, от слота фрейма.

Работа со связями сущности: функции **FXGetFirst**, **FXGetNext**

Вызовы данной группы позволяют получить информацию о синтактико-семантических связях сущностей в предложении (см. раздел «[Введение: Семантическая сеть](#)»).

Получение первой связи сущности

```
FX_ERROR FXGetFirst(  
    hResult, // дескриптор результата анализа  
    &idEntity, // указатель на идентификатор сущности  
    FX_INFO_RELATIONSHIP, // тип запрашиваемой информации  
    &idRelationship // указатель на результат  
);
```

Параметры

idEntity

[вх] Переменная должна содержать идентификатор той сущности, для которой запрашивается первая исходящая из нее связь.

idRelationship

[вых] Переменная, в которую будет записан идентификатор первой связи, исходящей из сущности (в случае, если функция возвратит **FX_SUCCESS**).

Получение очередной связи сущности

```
FX_ERROR FXGetNext(  
    hResult, // дескриптор результата анализа  
    &idEntity, // указатель на идентификатор сущности  
    FX_INFO_RELATIONSHIP, // тип запрашиваемой информации  
    &idRelationship // указатель на результат  
);
```

Параметры

idEntity

[вх] Переменная должна содержать идентификатор той сущности, для которой запрашивается очередная исходящая из нее связь.

idRelationship

[вх/вых] Переменная, которая при вызове функции должна содержать идентификатор связи, исходящей из сущности *idEntity*, полученный в результате вызова **FXGetFirst** или **FXGetNext**. После вызова функции данная переменная будет содержать идентификатор следующей связи, исходящей из сущности (в случае, если функция возвратит **FX_SUCCESS**).

Получение сущности, в которую входит связь

```
FX_ERROR FXGetFirst(  
    hResult, // дескриптор результата анализа  
    &idRelationship, // указатель на идентификатор связи  
    FX_INFO_TARGET_ENTITY_FOR_RELATIONSHIP, // тип запрашиваемой  
        информации  
    &idEntity // указатель на результат  
);
```

Параметры

idRelationship

[вх] Переменная должна содержать идентификатор той связи, для которой запрашивается сущность, содержащая связь.

idEntity

[вых] Переменная для записи идентификатора той сущности, в которую входит связь.

Получение коннектора, при помощи которого установлена связь

```
FX_ERROR FXGetFirst(  
    hResult, // дескриптор результата анализа  
    &idRelationship, // указ. на идентификатор связи  
    FX_INFO_CONNECTOR_ENTYTY_FOR_RELATIONSHIP, // тип запрашиваемой  
        информации  
    &idEntity // указатель на результат  
);
```

Параметры

idRelationship

[вх] Переменная должна содержать идентификатор той связи, для которой запрашивается сущность-коннектор, с чьей помощью установлена эта связь.

idEntity

[вых] Переменная, в которую будет записан идентификатор сущности-коннектора (предлога, союза), при помощи которого установлена связь, если функция возвратит FX_SUCCESS. В случае, если связь установлена без коннектора, функция возвратит FX_WARN_NO_MORE_RESULTS.

Получение сущности, кореферентной заданной

```
FX_ERROR FXGetFirst(  
    hResult, // дескриптор результата анализа  
    &idEntity, // указатель на идентификатор сущности  
    FX_INFO_ENTITY_COREFERENT, // тип запрашиваемой информации  
    &idCoreferent // указатель на результат  
);
```

Параметры

idEntity

[вх] Переменная должна содержать идентификатор той сущности, для которой запрашивается кореферентная ей сущность.

idCoreferent

[вых] Переменная, в которую будет записан идентификатор сущности, кореферентной заданной. Ссылки на кореферентные сущности всегда идут от более позднего к более раннему упоминанию сущности в тексте (*новый налог на имущество* ← *налог* ← *этот налог*).

Работа с фреймами: функции **FXGetFirst**, **FXGetNext**

Вызовы данной группы позволяют получить всю информацию о найденных в тексте фреймах (см. раздел «[Введение: фреймы и поиск ситуаций](#)»).

Получение первого фрейма в тексте

```
FX_ERROR FXGetFirst(  
    hResult, // дескриптор результата анализа  
    NULL, // не используется  
    FX_INFO_FRAME, // тип запрашиваемой информации  
    &idFrame // указатель на результат  
);
```

Параметры

idFrame

[вых] Переменная, в которую будет записан идентификатор первого фрейма, найденного в тексте (в случае, если функция возвратит **FX_SUCCESS**).

Получение очередного фрейма в тексте

```
FX_ERROR FXGetNext(  
    hResult, // дескриптор результата анализа  
    NULL, // не используется  
    FX_INFO_FRAME, // тип запрашиваемой информации  
    &idFrame // указатель на результат  
);
```

Параметры

idFrame

[вх/вых] Переменная, которая при вызове функции должна содержать идентификатор фрейма, полученный в результате вызова **FXGetFirst** или **FXGetNext**. После вызова функции данная переменная будет содержать идентификатор следующего фрейма, найденного в тексте (в случае, если функция возвратит **FX_SUCCESS**).

Получение первого слота фрейма

```
FX_ERROR FXGetFirst(  
hResult, // дескриптор результата анализа  
&idFrame, // указатель на идентификатор фрейма  
FX_INFO_FRAME_SLOT, // тип запрашиваемой информации  
&idSlot // указатель на результат  
);
```

Параметры

idFrame

[вх] Переменная должна содержать идентификатор того фрейма, для которого запрашивается первый слот.

idSlot

[вых] Переменная, в которую будет записан идентификатор первого слота фрейма (в случае, если функция возвратит `FX_SUCCESS`).

Получение очередного слота фрейма

```
FX_ERROR FXGetNext(  
hResult, // дескриптор результата анализа  
&idFrame, // указатель на идентификатор фрейма  
FX_INFO_FRAME_SLOT, // тип запрашиваемой информации  
&idSlot // указатель на результат  
);
```

Параметры

idFrame

[вх] Переменная должна содержать идентификатор того фрейма, для которого запрашивается очередной слот.

idSlot

[вх/вых] Переменная, которая при вызове функции должна содержать идентификатор слота фрейма *idFrame*, полученный в результате вызова [FXGetFirst](#) или [FXGetNext](#). После вызова функции данная переменная будет содержать идентификатор следующего слота этого фрейма (в случае, если функция возвратит `FX_SUCCESS`).

Получение сущности, соответствующей слоту фрейма

```
FX_ERROR FXGetFirst(  
    hResult, // дескриптор результата анализа  
    &idSlot, // указатель на идентификатор слота  
    FX_INFO_ENTITY_FOR_FRAME_SLOT, // тип запрашиваемой информации  
    &idEntity // указатель на результат  
);
```

Параметры

idSlot

[вх] Переменная должна содержать идентификатор того слота, для которого запрашивается соответствующая ему сущность (участник ситуации).

idEntity

[вых] Переменная, в которую будет записан идентификатор сущности, соответствующей слоту (если функция возвратит `FX_SUCCESS`). На основании этого идентификатора можно далее получить всю информацию об участнике ситуации, соответствующей фрейму. В случае, если необязательный участник, указанный в шаблоне фрейма, не был найден в тексте, функция возвратит `FX_WARN_NO_MORE_RESULTS`.

Работа с актуальным контекстом: функции FXGetFirst, FXGetNext

Вызовы данной группы позволяют получить всю информацию об актуальном местовременном контексте, выделенном в тексте. Подробная информация приведена в разделе «[Приложение 6. Описание место-временного контекста](#)».

Актуальный контекст представляет собой объект, от которого можно получить как соответствующую ему сущность, выделенную в тексте, так и дополнительные атрибуты.

Получение первого актуального контекста текста

```
FX_ERROR FXGetFirst(  
    hResult, // дескриптор результата анализа  
    NULL, // не используется  
    FX_INFO_CONTEXT, // тип запрашиваемой информации  
    &idContext // указатель на результат  
);
```

Параметры

idContext

[вых] Переменная, в которую будет записан идентификатор первого актуального контекста текста – упомянутого в тексте места или времени его создания (в случае, если функция возвратит FX_SUCCESS).

Получение очередного актуального контекста текста

```
FX_ERROR FXGetNext(  
    hResult, // дескриптор результата анализа  
    NULL, // не используется  
    FX_INFO_CONTEXT, // тип запрашиваемой информации  
    &idContext // указатель на результат  
);
```

Параметры

idContext

[вх/вых] Переменная, которая при вызове функции должна содержать идентификатор актуального контекста, полученный в результате вызова [FXGetFirst](#) или [FXGetNext](#). После вызова функции данная переменная будет содержать идентификатор следующего актуального контекста, найденного в тексте (если функция возвратит FX_SUCCESS).

Получение сущности, соответствующей актуальному контексту

```
FX_ERROR FXGetFirst(  
hResult, // дескриптор результата анализа  
&idContext, // указатель на идентификатор контекста  
FX_INFO_ENTITY_FOR_CONTEXT, // тип запрашиваемой информации  
&idEntity // указатель на результат  
);
```

Параметры

idContext

[вх] Переменная должна содержать идентификатор того актуального контекста, для которого запрашивается соответствующая ему сущность (обозначение места или времени).

idEntity

[вых] Переменная, в которую будет записан идентификатор сущности, соответствующей контексту. На основании этого идентификатора можно далее получить всю информацию о сущности, обозначающей в тексте место или время его создания.

Получение информации о текущей настройке библиотеки: функция `FXGetMetaData`

Функция выдает в формате XML описание типов сущностей и фреймов, которые могут быть извлечены из текста при текущих загруженных библиотекой настройках – словарях, правилах, описаниях фреймов. Описание формата приведено в [Приложении 8](#).

```
FX_ERROR FXGetMetaData(  
    FX_HANDLE hContext, // анализатор текста  
    unsigned long* pulBufferSize, // размер буфера  
    wchar_t* pXmlInfo // указатель на результат  
);
```

Параметры

hContext

[вх] Дескриптор анализатора, посредством которого выполняется анализ текста.

pulBufferSize

[вх/вых] Указатель на переменную, которая при вызове функции содержит размер в байтах передаваемого буфера *pBuffer*. После вызова функции значение переменной будет равно реальному количеству байт, необходимых для размещения значения атрибута, включая завершающий ноль для строковых значений.

pXmlInfo

[вых] Указатель на текстовый буфер, оканчивающийся нулевым байтом и содержащий в формате XML описание контекста библиотеки.

Описание атрибутов

В этом разделе описываются атрибуты различных объектов, которые можно получить при помощи функций библиотеки `FXGetAttributeValueCount` и `FXGetAttributeValue`.

Атрибуты предложения

Значения описанных ниже атрибутов можно получить для каждого предложения, выделенного в тексте, при помощи функций библиотеки `FXGetAttributeValueCount` и `FXGetAttributeValue`.

Идентификатор атрибута	Тип атрибута	Описание
<code>FX_ATTR_SENTENCE_TYPE</code>	<code>FX_ATTRTYPE_LONG</code>	Тип предложения в тексте. Возможные значения: <code>FX_SENTENCETYPE_BLOCK</code> – с данного предложения начинается новый блок текста; <code>FX_SENTENCETYPE_PARAGRAPH</code> – с данного предложения начинается новый параграф текста. <code>FX_SENTENCETYPE_NORMAL</code> – обычное предложение. Показателями начала блока текста в html-формате считаются теги <code><TR></code> , <code><TD></code> , <code><H1></code> , <code><H2></code> , ..., и ряд других. Показателями начала параграфа текста в html-формате считаются теги <code><P></code> , <code></code> . Критерии выделения параграфов и предложений в тексте на основе особенностей его форматирования задаются в конфигурационном файле. Общие критерии выделения предложений основаны на правилах русской пунктуации.
<code>FX_ATTR_SENTENCE_OFFSET</code>	<code>FX_ATTRTYPE_LONG</code>	Количество символов от начала текста до начала предложения.
<code>FX_ATTR_SENTENCE_LENGTH</code>	<code>FX_ATTRTYPE_LONG</code>	Количество символов от начала предложения до его конца.
<code>FX_ATTR_SENTENCE_TEXT</code>	<code>FX_ATTRTYPE_STRING</code>	Текст предложения, очищенный от элементов форматирования текста (html-тегов, переносов строки и т.п.).
<code>FX_ATTR_SENTENCE_TEXT_CHECKSUM</code>	<code>FX_ATTRTYPE_LONG</code>	Контрольная сумма предложения.
<code>FX_ATTR_SENTENCE_TAXISINFO</code>	<code>FX_ATTRTYPE_LONG</code>	Число с информацией о таксисных характеристиках предложения. Число интерпретируется как набор битов, каждый из которых обозначает категорию. Значение бита 1 свидетельствует о

		принадлежности бита к соответствующей категории. Возможна комбинация (битовая маска) следующих значений: <i>SENTENCE_TAXIS_EXPRESSION</i> – эмоциональность; <i>SENTENCE_TAXIS_INTERROGATION</i> – вопрос; <i>SENTENCE_TAXIS_INTERVIEW</i> – интервью; <i>SENTENCE_TAXIS_PERSONALITY</i> – персонализация (1-2 лицо); <i>SENTENCE_TAXIS_REASONING</i> – рассуждение; <i>SENTENCE_TAXIS_FACTS</i> – отсылка к фактам; <i>SENTENCE_TAXIS_COLLOQUIAL</i> – просторечность, разговорность; <i>SENTENCE_TAXIS_PRESUMPTION</i> – гипотетичность; <i>SENTENCE_TAXIS_NOSUBJECT</i> – без субъекта в 3-ем лице.
--	--	--

Атрибуты сущности

Значения описанных ниже атрибутов можно получить для каждой сущности, выделенной в предложении, при помощи функций библиотеки **FXGetAttributeValueCount** и **FXGetAttributeValue**.

Идентификатор атрибута	Тип атрибута	Описание
Общие атрибуты сущности		
FX_ATTR_ENTITY_ID	FX_ATTRTYPE_STRING	Идентификатор референтности, который позволяет установить идентичность сущностей, имеющих одинаковые имена, как в пределах одного текстового документа, так и в пределах множества текстов. Так, если сущность соответствует объекту пользователя, заданному xml-описанием (например, <i>Александр Сергеевич Пушкин, Сбербанк</i>), то идентификатор референтности будет всегда иметь константное значение, соответствующее наименованию сущности; для именованной сущности, не заданной xml-описанием (<i>Иванов Василий Петрович, ООО 'Ромашка'</i>) идентификатор референтности будет уникален только в пределах одного документа; для неименованной сущности (<i>компьютерная технология, новый, разрабатывать программу</i>) данный идентификатор будет иметь уникальное значение при каждом упоминании в тексте одноименной сущности, за исключением тех случаев, когда лингвистическими алгоритмами установлено, что речь идет об одной и той же сущности в пределах документа (например: <i>компьютерная технология ... эта технология</i>). Идентификатор референтности рекомендуется использовать в качестве первичного ключа при сохранении сущностей в базе данных.
FX_ATTR_ENTITY_NAME	FX_ATTRTYPE_STRING	Имя сущности в канонической (нормальной) форме, построенное программой на основании правил русской грамматики. Может иметь несколько значений, например, <i>оружие, химическое оружие, бактериологическое оружие</i> для сущности, связанной со словом <i>оружие</i> во фразе <i>химического и бактериологического оружия</i> . Для всех различных обозначений одного и того же именованного объекта (например, <i>Александр Петрович Иванов, Иванов А., Иванов, он</i>), отождествленных программой в различных местах текста, будет выдаваться одно и то же, наиболее полное имя – в данном случае <i>Иванов Александр Петрович</i> . При построении канонической формы названий организаций выдаются только имена собственные, если

		таковые имеются (<i>ООО «Мобильные телесистемы» = «Мобильные телесистемы»</i>).
FX_ATTR_ENTITY_NAMETYPE	FX_ATTRTYPE_LONG	Тип имени сущности. Возможные значения: <i>FX_ENTITY_NAMETYPE_FULL</i> – полное имя сущности (в приведенном выше примере – <i>химическое оружие</i> и <i>бактериологическое оружие</i>); <i>FX_ENTITY_NAMETYPE_DERIVATED</i> – неполное имя сущности, входящее в состав более полного (в приведенном выше примере – <i>оружие</i>); В зависимости от решаемой задачи, целесообразно использовать либо только полные, либо все доступные имена сущности.
FX_ATTR_ENTITY_NAMETEMPLATE	FX_ATTRTYPE_STRING	Вариант имени сущности в виде, где каждое слово стоит в нормальной (канонической) форме, например, <i>новый машина российский президент</i> , что соответствует значению атрибута FX_ATTR_ENTITY_NAME, равному <i>новая машина российского президента</i> . Может использоваться для генерации поискового запроса (расширения каждого слова словосочетания всеми грамматическими формами) и для определения взаимной вложенности имен сущностей, в частности при построении иерархий имен.
FX_ATTR_ENTITY_NAMELENGTH	FX_ATTRTYPE_LONG	Длина имени сущности (число сущностей имени).
FX_ATTR_ENTITY_TYPE	FX_ATTRTYPE_STRING	Семантический тип сущности. Значения могут определяться в словарях и в правилах выделения специальных конструкций. Кроме того, несколько предопределенных типов выделяются алгоритмами библиотеки. Стандартные возможные значения приведены в разделе «Приложение 4. Типы выделяемых сущностей».
FX_ATTR_ENTITY_COMMONTYPE	FX_ATTRTYPE_LONG	Идентификатор общего типа сущности. Возможные значения: <i>CommonTypeDelim</i> – разделитель; <i>CommonTypeWord</i> – слово; <i>CommonTypeUnknownWord</i> – неизвестное слово; <i>CommonTypeSpec</i> – специальная конструкция; <i>CommonTypeLatinicWord</i> – слово, написанное латиницей.
FX_ATTR_ENTITY_PARTOFSENTENCE	FX_ATTRTYPE_LONG	Член предложения. Возможны значения: <i>PART_SENTENCE_UNDEFINED</i> – роль в предложении не определена; <i>PART_SENTENCE_SERVICE</i> – служебный элемент предложения; <i>PART_SENTENCE_ADVERBIAL_MODIFIER</i> – обстоятельство (обычно – наречие);

		<i>PART_SENTENCE_ATTRIBUTE</i> – определение (прилагательное); <i>PART_SENTENCE_ATTRIBUTE_GENITIVE</i> – определение (существительное-генитив); <i>PART_SENTENCE_ATTRIBUTE_VERB</i> – определение (причастие); <i>PART_SENTENCE_APPPOSITION</i> – приложение, уточнение; <i>PART_SENTENCE_ADVERBIAL_MODIFIER_MANNER</i> – деепричастие; <i>PART_SENTENCE_ADVERBIAL_PREPOSITIONAL_MODIFIER</i> – именная группа с предлогом (сирконстант); <i>PART_SENTENCE_ADVERBIAL_MODIFIER_TIME</i> – обстоятельство времени; <i>PART_SENTENCE_INDIRECT_OBJECT</i> – косвенное дополнение; <i>PART_SENTENCE_PREDICATE</i> – сказуемое; <i>PART_SENTENCE_DIRECT_OBJECT</i> – прямое дополнение; <i>PART_SENTENCE_PREDICATE_NOMINAL</i> – именное, субстантивное сказуемое (человек – чудо); <i>PART_SENTENCE_SUBJECT</i> – подлежащее.
FX_ATTR_ENTITY_CLAUSETYPE	FX_ATTRTYPE_LONG	Тип синтаксической клаузы – части предложения, в которой упоминается сущность. Возможно нулевое значение <i>CLAUSE_TYPE_UNDEFINED</i>, если тип клаузы установить не удалось, или комбинация (битовая маска) следующих значений: <i>CLAUSE_TYPE_SUBORDINATE</i> – сущность входит в придаточное предложение; <i>CLAUSE_TYPE_PARTICIPLE</i> – сущность входит в причастный оборот; <i>CLAUSE_TYPE_PARTICIPLE_ADVERB</i> – сущность входит в деепричастный оборот; <i>CLAUSE_TYPE_BRACKETS</i> – сущность входит в часть предложения, обрамленную скобками; <i>CLAUSE_TYPE_SPEECH</i> – сущность является частью прямой или косвенной речи.
FX_ATTR_ENTITY_TAXISINFO	FX_ATTRTYPE_LONG	Комплекс так называемых таксисных характеристик: видо-временные показатели глагола, модально-модусные смыслы. Возможна комбинация (битовая маска) следующих значений: <i>TAXIS_TIME_UNDEFINED</i> – глагольное время не определено; <i>TAXIS_TIME_PRESENT</i> – глагол употреблен в настоящем времени; <i>TAXIS_TIME_PAST</i> – глагол употреблен в прошедшем времени; <i>TAXIS_TIME_FUTURE</i> – глагол употреблен в будущем времени;

		<i>TAXIS_ASPECT_UNDEFINED</i> – глагольный вид не определен; <i>TAXIS_ASPECT_IMPERFECT</i> – глагол в несовершенном виде; <i>TAXIS_ASPECT_PERFECT</i> – глагол в совершенном виде; <i>TAXIS_MODUS_UNDEFINED</i> – модально-модусные характеристики не выявлены; <i>TAXIS_MODUS_NEGATION</i> – сущность употребляется с показателем отрицания (<i>не купил, не яблоко</i>); <i>TAXIS_MODUS_IRREALITY</i> – сущность употребляется с показателем ирреальности (<i>вроде бы яблоко, если купит</i>); <i>TAXIS_MODUS_REALITY</i> – сущность употребляется с показателем реальности (<i>именно яблоко, как стало известно, яблоки продаются</i>).
FX_ATTR_ENTITY_TEXT_CHECKSUM	FX_ATTRTYPE_LONG	Контрольная сумма имени сущности
FX_ATTR_ENTITY_OFFSET	FX_ATTRTYPE_LONG	Смещение упоминания сущности от начала текста – количество символов. Может использоваться для подсветки в тексте.
FX_ATTR_ENTITY_LENGTH	FX_ATTRTYPE_LONG	Длина упоминания сущности в тексте – количество символов. Может использоваться для подсветки в тексте.
FX_ATTR_ENTITY_TEXT	FX_ATTRTYPE_STRING	Исходный текст, соответствующий сущности в тексте и очищенный от элементов форматирования.
FX_ATTR_ENTITY_ATTRIBUTES	FX_ATTRTYPE_STRING	Строка, содержащая перечень дополнительных атрибутов сущности в виде пар <i>ТипАтрибута.ИмяАтрибута=ЗначениеАтрибута</i>; например: <pre>NamedObject.Key=ДИРЕКТОР; NamedObject.Lastname=ЛИСИЦИН; NamedObject.Middlename= ВЛАДИМИРОВИЧ; NamedObject.Name=ИВАН</pre> Такие атрибуты с любым именем, задаваемые пользователем, могут выделяться при помощи специально написанных правил для выделения специальных конструкций (все они обязаны иметь тип <i>Саре</i>). Кроме того, встроенные алгоритмы выделения именованных объектов могут создавать такие атрибуты с предопределенными именами, как в приведенном примере. Для преобразования строки в вектор пар { <i>ТипАтрибута.ИмяАтрибута, ЗначениеАтрибута</i> } можно использовать функцию <pre>_FXParseAttributesString(string sAttributes,</pre>

		std::vector<PAIR_STR_STR> &vAttributes)
FX_ATTR_ENTITY_NAMEENTITIES	FX_ATTRTYPE_IDARRAY	Массив идентификаторов объекта FX_ID, из которых составлено сгенерированное библиотекой имя сущности. Размер массива в байтах и число элементов FX_ID в нем следует получить «пустым запросом».
FX_ATTR_ENTITY_OFFSET_IN_SENTENCE	FX_ATTRTYPE_LONG	Смещение упоминания сущности от начала предложения в тексте, возвращаемом по запросу для атрибута <i>FX_ATTR_SENTENCE_TEXT</i> (см. раздел «Атрибуты предложения»), – количество символов. Может использоваться для подсветки в тексте.
FX_ATTR_ENTITY_LENGTH_IN_SENTENCE	FX_ATTRTYPE_LONG	Длина упоминания сущности в предложении в тексте, возвращаемом по запросу для атрибута <i>FX_ATTR_SENTENCE_TEXT</i> (см. раздел «Атрибуты предложения»), – количество символов. Может использоваться для подсветки в тексте.
FX_ATTR_ENTITY_ORTHOGRAPHIC_TYPE	FX_ATTRTYPE_LONG	Число, указывающее для известных (нормальных) слов на наличие заглавных букв и знаков препинания: ORTHOGRAPHIC_TYPE_UNDEFINED (0) – тип не определен; ORTHOGRAPHIC_TYPE_ALL_CAPITALS (1) – все буквы заглавные; ORTHOGRAPHIC_TYPE_FIRST_CAPITAL (2) – первая буква заглавная; ORTHOGRAPHIC_TYPE_FIRST_SMALL (3) – первая буква строчная; ORTHOGRAPHIC_TYPE_BREAK (4) – точка; ORTHOGRAPHIC_TYPE_QUOTE (5) – кавычка (любая); ORTHOGRAPHIC_TYPE_QUOTE_OPEN (6) – открывающаяся кавычка; ORTHOGRAPHIC_TYPE_QUOTE_CLOSE (7) – закрывающаяся кавычка; ORTHOGRAPHIC_TYPE_COLON (8) – двоеточие; ORTHOGRAPHIC_TYPE_SEMICOLON (9) – точка с запятой; ORTHOGRAPHIC_TYPE_DASH (10) – тире; ORTHOGRAPHIC_TYPE_BRACKET_OPEN (11) – открывающаяся скобка; ORTHOGRAPHIC_TYPE_BRACKET_CLOSE (12) – закрывающаяся скобка; ORTHOGRAPHIC_TYPE_HYPHEN (13) – дефис; ORTHOGRAPHIC_TYPE_COMMA (14) – запятая; ORTHOGRAPHIC_TYPE_HYPHEN_STRONG (15) – слово, пишущееся через дефис, написанное раздельно.

<i>Грамматические атрибуты сущности</i>		
FX_ATTR_MORPH_SPEECHPART	FX_ATTRTYPE_LONG	Часть речи. Коды возможных значений описаны в файле <i>_grammar_pos.h</i> .
FX_ATTR_MORPH_FORMINDEX	FX_ATTRTYPE_LONG	Номер словоформы в парадигме словоизменения. Описание приведено в Приложении 8.
FX_ATTR_MORPH_PARADIGMTYPE	FX_ATTRTYPE_LONG	Тип парадигмы словоизменения. Описание приведено в Приложении 8.
FX_ATTR_MORPH_CASE	FX_ATTRTYPE_LONG	Падеж. Описание приведено в Приложении 9.
FX_ATTR_MORPH_GENDER	FX_ATTRTYPE_LONG	Род. Описание приведено в Приложении 9.
FX_ATTR_MORPH_NUMBER	FX_ATTRTYPE_LONG	Число. Описание приведено в Приложении 9.
FX_ATTR_MORPH_NORMALFORM	FX_ATTRTYPE_STRING	Слово в нормальной форме.

Атрибуты связи

Значения описанных ниже атрибутов можно получить для каждой связи между парой сущностей, выделенных в предложении, при помощи функций библиотеки **FXGetAttributeValueCount** и **FXGetAttributeValue**.

Идентификатор атрибута	Тип атрибута	Описание
FX_ATTR_RELATIONTYPE	FX_ATTRTYPE_LONG	Тип связи. Коды возможных значений: RELATION_TYPE_UNDEFINED – тип связи не имеет названия (определяется связанной сущностью); RELATION_TYPE_HOMOGEN – однородные члены (<i>твердый</i> ⇔ <i>хрупкий</i>, <i>завод</i> ⇔ <i>фабрика</i>, <i>писать</i> ⇔ <i>читать</i>); RELATION_TYPE_QUANTIFIER – частица, местоимение, прилагательное (<i>этот</i> ← <i>человек</i>, <i>не</i> ← <i>поддержал</i>); RELATION_TYPE_NUMERIC – связь существительного с числительным (<i>пять</i> ← <i>человек</i>); RELATION_TYPE_ATTR_ADVERB – связь с наречием (<i>уволить</i> → <i>нехотя</i>); RELATION_TYPE_ATTR_ADJECTIVE – связь существительного с прилагательным (<i>президентский</i> ← <i>указ</i>); RELATION_TYPE_ATTR_NOUN – связь существительного с приложением, или связь в конструкции с предикатом тождества (<i>директор</i> → <i>Иванов</i>, <i>Иванов</i> → (<i>это</i>) <i>директор</i>); RELATION_TYPE_ATTR_GENITIVE – связь существительного с подчиненным генитивом, не описанная в словаре моделей управления (<i>указ</i> → <i>президента</i> → <i>России</i>); RELATION_TYPE_INFINITIVE – связь в глагольной группе (<i>попытка</i> → <i>научить</i> → <i>стрелять</i>); RELATION_TYPE_CIRCUMSTANCE – связь существительного или глагола с сирконстантом – необязательным участником ситуации (<i>написал</i> → (<i>в</i>) <i>Америке</i>, <i>цветок</i> → (<i>на</i>) <i>окне</i>); RELATION_TYPE_ARGUMENT – связь предиката с аргументом, устанавливаемая на основании словаря моделей управления, конкретный тип связи описывается указанными ниже атрибутами связи (<i>Иванов</i> ← <i>подписал</i>, <i>подписание</i> → <i>указа</i>, <i>оплата</i> → <i>услуги</i>); RELATION_TYPE_VERB_VERB – деепричастие при глаголе (<i>шел</i> → <i>качаясь</i>); RELATION_TYPE_ATTR_PARTICIPLE – причастие при существительном (<i>пишущее</i> ← <i>устройство</i>, <i>использованный</i> ← <i>контейнер</i>); RELATION_TYPE_NEXT – связь от предшествующей сущности к следующей за ней, которая позволяет использовать отношение упорядоченности составляющих текста при написании шаблонов для поиска фактов в тех случаях, когда синтаксический анализ не может установить связь на основе правил грамматики языка; CONTEXT_TYPE_CONTEXT_TIME – связь сущности-обозначения события (действия, процесса, состояния) с другой упомянутой в тексте сущностью – обозначением

		соответствующего момента времени (контекстом времени); <i>CONTEXT_TYPE_CONTEXT_PLACE</i> – связь сущности-обозначения события (действия, процесса, состояния) с другой упомянутой в тексте сущностью – обозначением соответствующего места (контекстом места); Для получения строкового названия связи по коду типа связи используйте функцию: <pre>const char* _FXGetRelationTypeString(char cCode)</pre>
FX_ATTR_RELATIONCASE	FX_ATTRTYPE_LONG	Семантический падеж аргумента предиката – имени существительного или местоимения, связанного подчинительной связью с глаголом, существительным, прилагательным. Соответствует грамматическому падежу в модели управления предиката, если пара <i>предикат</i> → <i>аргумент</i> употребляется в простом неосложненном предложении с активной конструкцией (без причастных оборотов и т.п.). Пример: <i>подписать</i> → <i>указ</i> – винительный грамматический и винительный семантический; <i>указ</i> ← <i>подписан</i> – именительный грамматический, но винительный семантический; <i>подписанным</i> ← <i>указом</i> – творительный грамматический, но винительный семантический. Коды возможных значений соответствуют латинским названиям падежей: <pre>RELATION_CASE_UNDEFINED RELATION_CASE_NOMINATIVE (Им) RELATION_CASE_GENITIVE (Род) RELATION_CASE_DATIVE (Дат) RELATION_CASE_ACCUSATIVE (Вин) RELATION_CASE_INSTRUMENTAL (Тв) RELATION_CASE_PREPOSITIONAL (Предл)</pre> Для получения строкового названия по коду используйте функцию: <pre>const char* _FXGetRelationCaseString(char cCode) .</pre>
FX_ATTR_RELATIONCONNECTOR	FX_ATTRTYPE_STRING	Коннектор отношения – предлог или союз, при помощи которого в тексте устанавливается синтаксическая связь, например: <i>при помощи, при, если, и</i>.
FX_ATTR_RELATIONROLE	FX_ATTRTYPE_STRING	Семантическая роль аргумента предиката – имени существительного или местоимения, связанного подчинительной связью с глаголом, существительным, прилагательным. Извлекается из словаря моделей управления. Основные полезные значения: <pre>RELATION_ROLE_SUBJECT – субъект; RELATION_ROLE_OBJECT – объект; RELATION_ROLE_CONTENT – содержание; RELATION_ROLE_BENEFACTIVE – бенефактив; RELATION_ROLE_WAY – способ; RELATION_ROLE_LOCATIVE – локатив;</pre>

		<i>RELATION_ROLE_PURPOSE</i> – цель; <i>RELATION_ROLE_CAUSE</i> – причина; <i>RELATION_ROLE_SITUATION</i> – ситуатив; <i>RELATION_ROLE_FEATURE</i> – признак.
--	--	--

Атрибуты фрейма

Значения описанных ниже атрибутов можно получить для каждого фрейма (ситуации), найденного в тексте, при помощи функций библиотеки **FXGetAttributeValueCount** и **FXGetAttributeValue**.

Идентификатор атрибута	Тип атрибута	Описание
FX_ATTR_FRAME_CLASS	FX_ATTRTYPE_STRING	Класс фрейма, указанный в конфигурационном файле config.xml в виде значения атрибута <i>class</i> тега <template> в секции <semantic expounder> .
FX_ATTR_FRAME_TEMPLATE	FX_ATTRTYPE_STRING	Строковый идентификатор шаблона, по которому был найден фрейм, заданный в конфигурационном файле config.xml в виде содержимого тега <template> в секции <semantic expounder> .
FX_ATTR_FRAME_TEXT_CHECKSUM	FX_ATTRTYPE_LONG	Контрольная сумма для части текста (обычно цепочки слов между сущностями, соответствующими слотам фрейма), из которой извлечен фрейм. Применяется для проверки: выделение одного и того же фрейма (одного типа, с одинаковыми участниками) хотя бы два раза с разными контрольными суммами, означает, что соответствующая информация была выражена различными способами и вероятность ошибки при лингвистическом разборе крайне мала.

Значения описанных ниже атрибутов можно получить для каждого слота фрейма, найденного в тексте.

Идентификатор атрибута	Тип атрибута	Описание
FX_ATTR_FRAME_SLOT_NAME	FX_ATTRTYPE_STRING	Метка (строковое значение), соответствующая слоту фрейма, указанная в шаблоне. Обычно определяет роль участника-фигуранта в ситуации, описываемой фреймом <i>buyer, seller, owner, property</i> и т.п. Метки участников, не описанных в шаблоне, но выделенных программой на основании общих алгоритмов, имеют префикс <i>~Other, ~Time, ~Place</i> и т.п. Метки <i>~Time</i> и <i>~Place</i> приобретают фигуранты-сущности, обозначающие место-временной контекст ситуации, упомянутые в тексте время и место действия. Метку <i>~Other</i> приобретают все участники, выраженные именами существительными, связанными с глаголом, фигурирующим в факте. Фигуранты, выделенные в слот <i>~Other</i>, могут использоваться для последующей классификации ситуаций, выделенных на основании фреймов общего вида, при помощи словарей-фильтров, содержащих ключевые слова и словосочетания.
FX_ATTR_FRAME_SLOT_RESTRICTION	FX_ATTRTYPE_STRING	Логическое выражение – ограничение, соответствующее слоту фрейма, указанное в шаблоне.

Атрибуты контекста

Значения описанных ниже атрибутов текста можно получить для каждого актуального место-временного контекста, выделенного в тексте, при помощи функций библиотеки [FXGetAttributeValueCount](#) и [FXGetAttributeValue](#). Подробная информация о контекстах приведена в разделе «[Приложение 6. Описание место-временного контекста](#)».

Идентификатор атрибута	Тип атрибута	Описание
FX_ATTR_CONTEXT_TYPE	FX_ATTRTYPE_LONG	Тип контекста. Возможные значения: <i>RELATION_TYPE_CONTEXT_TIME</i> – актуальное место текста; <i>RELATION_TYPE_CONTEXT_PLACE</i> – актуальное время текста.

Атрибуты текста

Значения описанных ниже атрибутов можно получить для каждого текста при помощи функции библиотеки [FXGetAttributeValue](#).

Идентификатор атрибута	Тип атрибута	Описание
FX_ATTR_TEXT_SYNT_PARSING_RECALL	FX_ATTRTYPE_LONG	Полнота синтаксического разбора текста в процентном выражении.
FX_ATTR_TEXT_LEX_NORM	FX_ATTRTYPE_LONG	Процентное соотношение известных слов и общего количества слов в тексте без учета знаков препинания.
FX_ATTR_TEXT_SENT_NORM	FX_ATTRTYPE_LONG	Процентное соотношение нормальных предложений и общего их количества. Нормальными считаются, в частности, предложения с глаголами, не слишком длинные и не слишком короткие и т.п.
FX_ATTR_TEXT_LEX_COUNT	FX_ATTRTYPE_LONG	Количество слов в тексте без учета знаков препинания, цифр и специальных конструкций .
FX_ATTR_TEXT_SENT_COUNT	FX_ATTRTYPE_LONG	Количество предложений в тексте, имеющих 1 слово, помимо знаков препинания, цифр и специальных конструкций .
FX_ATTR_TEXT_NAMED_COUNT	FX_ATTRTYPE_LONG	Количество именованных объектов в тексте.
FX_ATTR_TEXT_TAXIS_<КАТЕГОРИЯ>	FX_ATTRTYPE_LONG	Число с информацией о количестве предложений заданной категории. Возможны следующие значения <КАТЕГОРИИ>: <i>EXPRESSION</i> – эмоциональность;

		<i>INTERROGATION</i> – вопрос; <i>INTERVIEW</i> – интервью; <i>PERSONALITY</i> – персонализация (1-2 лицо); <i>REASONING</i> – рассуждение; <i>FACTS</i> – отсылка к фактам; <i>COLLOQUIAL</i> – просторечность, разговорность; <i>PRESUMPTION</i> – гипотетичность; <i>SENTENCE_TAXIS_NOSUBJECT</i> – без субъекта в 3-ем лице; <i>DATA</i> – данные, числовые величины; <i>SCIENTIFIC</i> – научность; <i>COMPLEXITY</i> – грамматическая сложность.
--	--	--

Приложение 1.

Формат конфигурационного файла

Ниже описан формат конфигурационного файла, необходимого библиотеке для инициализации модуля анализа текста (параметр функции [FXCreateContext](#)).

XML-тег	Описание
configuration	Корневой элемент. Содержит теги <code><filter-sentence></code> , <code><text-parser></code> , <code><text-analysis></code> , <code><semantic-exponder></code> .
Настройки фильтра сущностей	
./filter-sentence	Настройки фильтра сущностей. Фильтр сущностей позволяет повысить скорость анализа текста за счет игнорирования предложений текста, не содержащих упоминаний сущностей заданных типов. Если ни один из типов сущностей не задан, полному анализу подлежат все предложения текста. Имеет атрибут <code>strong-filtering</code> с возможными значениями: • <i>on</i> – информация о предложениях, не содержащих ни одной сущности заданного типа, не будет включена в результат анализа текста; • <i>off</i> – предложения, не содержащие ни одной сущности заданного типа, будут исключены только из вычислительно емких стадий синтаксического разбора и поиска фактов. Вся информация о сущностях, выделенных в этих предложениях, будет доступна, за исключением синтактико-семантических связей и фактов. Может содержать теги <entity>, в каждом из которых указывается свой тип сущности.
./filter-sentence/entity	Один из типов сущностей, выделяемых библиотекой.
Настройки парсера текста	
./text-parser	Настройки парсера текста, который фрагментирует HTML или текст на слова, предложения, параграфы. Имеет атрибут <code>words-in-sentence</code>, значение которого задает максимально возможное количество слов в предложении, включая знаки препинания). Более длинные предложения будут разбиты на фрагменты с длиной, не превышающей данное значение. Содержит теги: <plain-fragmentation> (правила фрагментации), <end-sentence-abbreviations> и <end-sentence-abbreviations> (списки сокращений).
./text-parser/plain-fragmentation	Имеет атрибут <code>level</code>, значение которого задает способ фрагментации простого текста. Возможные значения: • <i>paragraph</i> – считать каждый блок текста, выделенный по правилам фрагментации, новым параграфом; • <i>sentence</i> – считать каждый блок текста, выделенный по правилам фрагментации, новым предложением. Содержит описание правил фрагментации текста – набор тегов <rule>.

./text-parser/ plain-fragmentation/rule	Настройки правила выделения фрагментов (предложений или параграфов в зависимости от значения атрибута level тега <plain-fragmentation>) в тексте. Атрибут id идентифицирует конкретное правило фрагментации. Может принимать следующие значения: • <i>spaces</i> – правило выделяет начало фрагмента, если новая строка текста начинается с пробельных символов количеством не менее заданного в атрибуте param. Символ табуляции считается за два пробела. • <i>spaces-dash</i> – правило выделяет начало фрагмента, если новая строка текста начинается с дефиса, перед которым следуют пробельные символы в количестве не менее заданного в атрибуте param. Символ табуляции считается за два пробела. • <i>newline</i> – правило выделяет начало фрагмента, если перед началом строки текста следуют переводы строки в количестве не менее заданного в атрибуте param. Атрибут param может иметь любое целочисленное неотрицательное значение, которое используется в качестве ограничения на количество символов в соответствующем правиле фрагментации. Атрибут status с возможными значениями <i>on</i> или <i>off</i> определяет, включено или нет соответствующее правило фрагментации.
./text-parser/ cape-tokenizer	Блок содержит правила CAPE для работы токенизатора. Атрибут status с возможными значениями <i>on</i> или <i>off</i> определяет, включено или нет соответствующее правило токенизации. Семантика применения правил описана в разделе «Семантика применения правил» документа «<i>Руководство администратора RCO Pattern Extractor</i>».
./text-parser/ abbreviations	Идентификатор ресурса, передаваемый в менеджер ресурсов для загрузки списка сокращений, которые используются при разбиении текста на слова. Каждая строка списка сокращений содержит последовательность символов, завершающуюся точкой (<i>обл.</i>, <i>г.</i>, <i>н.э.</i>, <i>т.д.</i>), которая будет выделяться как единое слово (с точкой на конце).
./text-parser/ end-sentence-abbreviations	Идентификатор ресурса, передаваемый в менеджер ресурсов для загрузки списка сокращений, используемых при разбиении текста на предложения. Каждая строка списка сокращений содержит последовательность символов, завершающуюся точкой (<i>н.э.</i>, <i>т.д.</i>), которая будет обозначать конец предложения, если следующее за ней слово начинается с заглавной буквы.
Настройки лингвистического анализатора текста	
./text-analysis	Настройки лингвистического анализатора текста.
./text-analysis/ options	Содержит установки общих параметров лингвистического анализа текста – набор тегов <param>.
./text-analysis/ options/param	Установка параметра анализа текста. Атрибут id идентифицирует конкретный параметр, значение которого задается атрибутом value. Атрибут id может принимать следующие значения: • <i>parse-verb-phrases</i> – если атрибут value имеет значение <i>off</i>, будет выполняться более простой и ускоренный синтаксический анализ текста, выделяющий только связи в пределах именных групп; если

	атрибут <code>value</code> имеет значение <code>on</code>, будет выполняться более полный и более длительный синтаксический анализ текста, выделяющий все связи в предложении; • <i>set-next-relations</i> – если атрибут <code>value</code> имеет значение <code>on</code>, в семантическую сеть будут дополнительно включены связи, отражающие порядок следования сущностей (тип связи <code>RELATION_TYPE_NEXT</code>); • <i>anaphora-for-pronoun</i> – если атрибут <code>value</code> имеет значение <code>on</code>, при анализе текста будут включены алгоритмы замены местоимений их референтами; • <i>anaphora-distance</i> – положительное значение атрибута <code>value</code> определяет максимальное допустимое количество предложений между косвенным упоминанием объекта (<i>компания, завод</i>) и его именем (<i>Лукойл, Ижевский машиностроительный завод</i>). Нулевое значение атрибута <code>value</code> запрещает отождествление косвенных упоминаний с именами собственными; • <i>anaphora-for-common</i> – если атрибут <code>value</code> имеет значение <code>on</code>, при анализе текста будут включены алгоритмы установления связей между кореферентными именами нарицательными (<i>новый налог на имущество ← налог ← этот налог</i>), что позволит получить более полные имена сущностей и заполнить дополнительные слоты фреймов за пределами отдельных предложений.
<code>./text-analysis/morphology</code>	Значение атрибута <code>first-dictionary</code> задает идентификатор ресурса, передаваемый в менеджер ресурсов для загрузки основного бинарного морфологического словаря. Значение атрибута <code>second-dictionary</code> задает идентификатор ресурса, передаваемый в менеджер ресурсов для загрузки дополнительного бинарного морфологического словаря, который используется для определения склонений имен собственных.
<code>./text-analysis/linguistic</code>	Значение атрибута <code>dictionary</code> задает идентификатор ресурса, передаваемый в менеджер ресурсов для загрузки бинарного словаря моделей управления.
<code>./text-analysis/dictionaries</code>	Перечень текстовых словарей – классификаторов ключевой лексики, используемой различными лингвистическими правилами. Содержит набор тегов <dictionary>.
<code>./text-analysis/dictionaries/dictionary</code>	Идентификатор ресурса, передаваемый в менеджер ресурсов для загрузки текстового словаря – классификатора ключевой лексики.
<code>./text-analysis/cape-filters</code>	Перечень текстовых словарей – фильтров, используемых модулем RCO Pattern Extractor. Содержит набор тегов <filter>.
<code>./text-analysis/cape-filters/filter</code>	Идентификатор ресурса, передаваемый в менеджер ресурсов для загрузки текстового словаря – фильтра.
<code>./text-analysis/cape-rules</code>	Перечень правил выделения специальных конструкций (для модуля RCO Extractor). Содержит набор тегов <rule-phase>.
<code>./text-analysis/cape-rules/rule-phase</code>	Включает перечень правил, выполняемых в ходе одной фазы – набор тегов <rule>. Семантика применения правил позволяет задать общую схему принятия решения о выделении объектов в случае наличия неоднозначностей (описана в разделе «Семантика применения правил» документа «<i>Руководство администратора RCO Pattern Extractor</i>»).

<code>./text-analysis/cape-rules/rule-phase/rule</code>	Идентификатор ресурса, передаваемый в менеджер ресурсов для загрузки текста правил выделения специальных конструкций.
<code>./text-analysis/user-objects</code>	Перечень описаний именованных объектов (персон и организаций). Содержит набор тегов <resource> .
<code>./text-analysis/user-objects/resource</code>	Идентификатор ресурса, передаваемый в менеджер ресурсов для загрузки текста описания объекта.
Настройки выделения фреймов	
<code>./semantic-expounder</code>	Определяет конфигурацию шаблонов для семантической интерпретации и выделения фреймов – секции <se-templates> и <se-filters>. Атрибут <code>status</code> с возможными значениями <i>on</i> или <i>off</i> позволяет включать или отключать функцию семантической интерпретации. Атрибут <code>validation-mode</code> может принимать значения <i>none</i> или <i>standard</i>. Во втором случае из числа найденных фреймов исключаются те, что соответствуют нереальным ситуациям, а из числа сущностей в слотах фреймов исключаются сущности, не принимавшие реального участия в ситуации, – это производится на основании лингвистических особенностей описания ситуации в тексте (наличия показателей отрицания и нереальности). Данное общее значение, устанавливаемое для всех шаблонов, может быть перекрыто значением одноименного атрибута в каждом конкретном шаблоне. Атрибут <code>min-slot-count</code> может принимать значения от 0 и выше. Его значение определяет, каким должно быть минимальное количество найденных участников ситуации, чтобы соответствующий фрейм был выделен и доступен в результатах анализа текста. Данное общее значение, устанавливаемое для всех шаблонов, может быть перекрыто значением одноименного атрибута в каждом конкретном шаблоне.
<code>./semantic-expounder/se-templates</code>	Содержит перечень шаблонов, используемых для выделения фреймов – набор тегов <template> .
<code>./semantic-expounder/se-templates/template</code>	Идентификатор ресурса, передаваемый в менеджер ресурсов для загрузки шаблона фрейма. Имеет атрибут <code>class</code>, значение которого возвращается библиотекой при выделении фрейма (атрибут <code>FX_ATTR_FRAME_CLASS</code>) и позволяет отождествить ситуации одного класса, выделенные на основе разных шаблонов. Имеет атрибут <code>view</code>, принимающий следующие значения: • <i>always</i> – все фреймы, выделенные по данному шаблону, будут доступны в результатах анализа текста (значение по умолчанию); • <i>never</i> – никакой фрейм, выделенный по данному шаблону, не будет доступен в результатах анализа текста (такие шаблоны либо добавляют связи в семантическую сеть для обработки другими шаблонами, либо описывают общие фреймы для последующей обработки семантическими фильтрами); • <i>unfiltered</i> – фрейм, выделенный по данному шаблону, будет доступен в результатах анализа текста только в том случае, если он не был отфильтрован семантическими фильтрами;

	• <i>unenclosed</i> – фрейм, выделенный по данному шаблону, будет доступен в результатах анализа текста только в том случае, если множество выделенных участников не является подмножеством участников другого выделенного фрейма (любого класса).
<code>./semantic-expounder/se-filters</code>	Содержит перечень семантических фильтров, используемых для последующей обработки фреймов, выделенных на основе шаблонов – набор тегов <code><filter></code> .
<code>./semantic-expounder/se-filters/filter</code>	Определяет структуру семантического фильтра, который позволяет выделить новый класс фреймов на основании уже выделенного класса при выполнении дополнительных ограничений. Фильтры применяются последовательно, в процессе их указания в файле конфигурации, в результате чего их можно применять иерархически, к фреймам, полученным в результате срабатывания других фильтров. Содержит набор тегов <code><dct></code>, каждый из которых задает словарь ограничений, подключаемых к слоту фрейма с определенной ролью. Имеет атрибут <code>source-class</code>, значение которого определяет, к какому классу выделенных фреймов может применяться фильтр. Имеет атрибут <code>target-class</code>, значение которого возвращается библиотекой при выделении нового фрейма (атрибут <code>FX ATTR FRAME CLASS</code>).
<code>./semantic-expounder/se-filters/filter/dct</code>	Идентификатор ресурса, передаваемый в менеджер ресурсов для загрузки словаря семантического фильтра. Каждая строка словаря может содержать слово, словосочетание или указание семантического типа сущности (значение атрибута <code>FX ATTR ENTITY_TYPE</code>), а также символ «*», означающий любую сущность. Кроме того, элемент словаря может содержать оператор отрицания – символ «!» в начале строки. Имеет атрибут <code>role</code>, значение которого задает роль участника ситуации, к которому будет применяться словарь: <i>~Other</i> означает, что словарь будет применяться ко всем участникам, независимо от их ролей. Если перед именем роли стоит префикс <i>_parent</i>: (например, role=" _parent:Goods"), то словарь будет применен ко всем сущностям, синтаксически подчиненным участнику ситуации в указанной роли. К одной и той же роли можно подключать несколько словарей, которые в этом случае просто объединяются в один, т.е. работают как <i>OR</i>. Имеет атрибут <code>comparison-mode</code>, который может иметь значение <i>verify</i> или <i>negate</i>. Если словари с атрибутом <code>comparison-mode="verify"</code> применяются к разным ролям – они работают как <i>AND</i> – для выделения нового фрейма в каждой из ролей должно найтись хотя бы по одному участнику – сущности, имя которой или семантический тип указаны в соответствующем словаре. Словари с атрибутом <code>comparison-mode="negate"</code> всегда работают как <i>OR</i>, даже если применяются к разным ролям: при нахождении участника – сущности, имя которой или семантический тип указаны в соответствующем словаре, новый фрейм не выделяется. Отрицание также можно указать прямо в словаре с атрибутом <code>comparison-mode="verify"</code> оператором «!» перед нужной строкой (например, для борьбы с омонимией в одной строке словаря указывается <i>бумага</i>, а в другой – <i>!ценная бумага</i>).

Ниже приведен пример конфигурационного файла.

```
<?xml version="1.0" encoding="windows-1252"?>
<configuration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="config.xsd">

  <filter-sentence strong-filtering="on">
    <entity>Person:Name</entity>
    <entity>Organization:Name</entity>
  </filter-sentence>

  <text-parser words-in-sentence="200">
    <plain-fragmentation level="paragraph">
      <rule id="spaces" status="on" param="2"/>
      <rule id="spaces-dash" status="on" param="1"/>
      <rule id="newline" status="on" param="2"/>
    </plain-fragmentation>
    <abbreviations>dic\shortcuts.txt</abbreviations>
    <end-sentence-abbreviations>dic\shortcuts_endsent.txt</end-sentence-
      abbreviations>
  </text-parser>

  <text-analysis>
    <options>
      <param id="parse-verb-phrases" value="on"/>
      <param id="set-next-relations" value="off"/>
      <param id="anaphora-for-pronoun" value="on"/>
      <param id="anaphora-distance" value="2"/>
    </options>

    <morphology first-dictionary="MorphDct.dat" second-
      dictionary="MorphDctNames.dat"/>
    <linguistic dictionary="LingDct.dat"/>
    <dictionaries>
      <dictionary>dic\idioms_adverb.dct</dictionary>
      <dictionary>dic\idioms_adverbtime.dct</dictionary>
      <dictionary>dic\idioms_conjunction.dct</dictionary>
      <dictionary>dic\person_position.dct</dictionary>
      <dictionary>dic\latinic.dct</dictionary>
    </dictionaries>
    <cape-filters>
      <filter>dic\prep_date.flt</filter>
      <filter>dic\month.flt</filter>
      <filter>dic\month_nom.flt</filter>
      <filter>dic\month_gen.flt</filter>
    </cape-filters>
    <cape-rules>
      <rule-phase control="optimal">
        <rule>dic\service.rul</rule>
      </rule-phase>
      <rule-phase control="optimal">
        <rule>dic\phone.rul</rule>
        <rule>dic\vehicle_number.rul</rule>
        <rule>dic\e-mail.rul</rule>
      </rule-phase>
      <rule-phase control="optimal">
        <rule>dic\date1.rul</rule>
      </rule-phase>
    </cape-rules>
    <user-objects>
      <object-desc>objects\Putin.xml</object-desc>
```

```
<object-desc>objects\Luzkov.xml</object-desc>
<object-desc>objects\Lukoil.xml</object-desc>
</user-objects>
</text-analysis>

<semantic-expounder status="on" filter="facts">
<se-templates>
<template class="договора">ld\contract1.dot</template>
<template class="договора">ld\contract2.dot</template>
<template class="договора">ld\contract3.dot</template>
<template class="купля-продажа">ld\buy1.dot</template>
<template class="купля-продажа">ld\buy2.dot</template>
<template class="купля-продажа">ld\sale1.dot</template>
<template class="купля-продажа">ld\sale2.dot</template>
</se-templates>
<se-filters>
<filter source-class="купля-продажа" target-class="купля-продажа:
    предприятия">
<dict comparison-mode="verify"
    role="Goods">ld\_predpriyatiya.flt</dict>
</filter>
<filter source-class="купля-продажа" target-class="купля-продажа:
    разное">
<dict comparison-mode="verify" role="Goods">ld\_pokupki.flt</dict>
<dict comparison-mode="negate"
    role="Goods">ld\_predpriyatiya.flt</dict>
<dict comparison-mode="negate" role="Goods">ld\_akcii.flt</dict>
</filter>
</se-filters>
</semantic-expounder>
</configuration>
```

Приложение 2.

Возможные значения кодов ошибки

Код ошибки, возвращаемый функциями библиотеки, имеет тип `FX_ERROR` и может принимать перечисленные ниже значения. Более подробное описание ошибки обычно можно получить в [файле журнала](#).

Код ошибки	Описание
<code>FX_ERR_NOT_ACCESSIBLE_IN_THIS_VERSION</code>	Вызываемая функция или обработка переданного ей параметра не поддерживается в используемой вами конфигурации библиотеки. Обратитесь к разработчику.
<code>FX_SUCCESS</code>	Нет ошибок.
<code>FX_WARN_NO_MORE_RESULTS</code>	Больше нет результатов.
<code>FX_WARN_DATA_TRUNCATED</code>	Функция вернула только часть запрошенных данных.
<code>FX_ERR_BAD_PARAMETER</code>	Задано недопустимое значение параметра.
<code>FX_ERR_BAD_CONFIG_FORMAT</code>	Нарушен формат конфигурационного файла.
<code>FX_ERR_BAD_INPUTTEXT</code>	Входной текст не может быть обработан.
<code>FX_ERR_OUT_OF_MEMORY</code>	Недостаточно памяти.
<code>FX_ERR_BUFFER_TOO_SMALL</code>	Переданный размер буфера недостаточен.
<code>FX_ERR_BAD_IDENTIFIER</code>	Передан неверный идентификатор.
<code>FX_ERR_BAD_HANDLE</code>	Передан неверный дескриптор.
<code>FX_ERR_NO_SUCH_ATTR</code>	Нет атрибута с таким идентификатором.
<code>FX_ERR_NO_VALUE</code>	Атрибут не имеет значения с таким индексом.
<code>FX_ERR_RESOURCE_NOT_FOUND</code>	Указанный ресурс не существует.
<code>FX_ERR_INTERNAL</code>	Внутренняя ошибка приложения.

Приложение 3.

Формат файла журнала

В ходе работы библиотека **RCO FX** периодически записывает в файл журнала сообщения о важных событиях: всевозможного рода ошибках и предупреждениях, начале и завершении отработки вызовов ключевых функций библиотеки, начале и завершении загрузки каждого внешнего ресурса, начале и завершении ключевых этапов анализа текста.

Имя файла журнала и уровень подробности сообщений задаются в файле инициализации библиотеки **fx.ini** (см. раздел «[Инициализация](#)»).

Каждая запись файла журнала начинается с новой строки и имеет формат:

[Дата Время, ИдентификаторПотока, СтатусЗаписи] ТекстЗаписи ,

где *СтатусЗаписи* может иметь значения **ERROR**, **WARNING**, **INFO**, **VERBOSE** (в зависимости от установленного в **fx.ini** уровня подробности записи с тем или иным статусом могут отсутствовать).

Ниже приведены примеры записей.

```
[15.12.2006 15:25:14.534, 3372, INFO] Start of
function::FXCreateContext.
[15.12.2006 15:25:15.237, 3372, INFO] Start of parsing dictionary:
dic\organization_key.dct
[15.12.2006 15:25:15.502, 3372, INFO] End of parsing dictionary:
dic\organization_key.dct. Loaded terms count: 2659
[20.12.2006 17:03:37.747, 1836, INFO]
Start of parsing user object description: objects\MinVneshTorg.xml
[20.12.2006 17:03:37.747, 1836, VERBOSE] object have been described
as: <object id="Министерство внешней торговли">
<term class="SimpleTerm" capitalization="Capital">МИНВНЕШТОРГ</term>
<term class="SimpleTerm" capitalization="Capital">МИНВНЕШТОРГА</term>
<term class="SimpleTerm" capitalization="Capital">МИНВНЕШТОРГУ</term>
<term class="SimpleTerm" capitalization="Capital">МИНВНЕШТОРГ</term>
<term class="SimpleTerm"
capitalization="Capital">МИНВНЕШТОРГОМ</term>
<term class="SimpleTerm" capitalization="Capital">МИНВНЕШТОРГЕ</term>
<term class="3WordsTerm" capitalization="Small">МИНИСТЕРСТВО and
ВНЕШНЕЙ and ТОРГОВЛИ</term>
<term class="3WordsTerm" capitalization="Small">МИНИСТЕРСТВА and
ВНЕШНЕЙ and ТОРГОВЛИ</term>
<term class="3WordsTerm" capitalization="Small">МИНИСТЕРСТВУ and
ВНЕШНЕЙ and ТОРГОВЛИ</term>
<term class="3WordsTerm" capitalization="Small">МИНИСТЕРСТВО and
ВНЕШНЕЙ and ТОРГОВЛИ</term>
<term class="3WordsTerm" capitalization="Small">МИНИСТЕРСТВОМ and
ВНЕШНЕЙ and ТОРГОВЛИ</term>
<term class="3WordsTerm" capitalization="Small">МИНИСТЕРСТВЕ and
ВНЕШНЕЙ and ТОРГОВЛИ</term>
</object>
[20.12.2006 17:03:37.747, 1836, INFO]
End of parsing user object description: objects\MinVneshTorg.xml
```

Приложение 4.

Типы выделяемых сущностей

При анализе текста функцией **FXGetMetaData** могут быть выделены базовые типы сущностей, перечисленные далее.

Дополнительно средства администрирования библиотеки позволяют специфицировать новые типы сущностей.

Наиболее просто новые типы определяются в словарях посредством перечисления ключевых слов (словосочетаний) для каждого из типов. При анализе текста все словосочетания, образованные от ключевых слов, будут отнесены к заданному типу. Ключевые слова записываются в файл, где каждая строка имеет вид «слово (словосочетание) MSYN возможный синоним», а в первой строке словаря указывается имя соответствующей категории, например *Special:Conflict*. В качестве примера словаря можно использовать любой из файлов вида *.dct, входящий в набор стандартных словарей.

Кроме того, существует возможность добавлять свои типы посредством задания особых правил для распознавания специальных конструкций, допустим, номеров автомобилей (см. документацию «**RCO Pattern Extractor. Компонент выделения конструкций в тексте**»). Все новые специфицированные сущности должны иметь семантический тип вида *Special:**, например *Special:Address*.

В таблице ниже приведена информация об основных типах выделяемых сущностей и соответствующих строковых значениях, которые возвращаются библиотекой при запросе строкового атрибута сущности `FX_ATTR_ENTITY_TYPE`. Для многих из возвращаемых строковых констант в заголовочных файлах библиотеки определены соответствующие идентификаторы, которые рекомендуется использовать вместо прямого указания строки в тексте программы.

Значение <code>FX_ATTR_ENTITY_TYPE</code>	Описание
<code>SEMANTIC_TYPE_OBJECT_NAMED</code>	Именованный объект, тип которого установить не удастся.
<code>SEMANTIC_TYPE_ORGANIZATION_NAMED</code>	Именованные организации, например: <i>Магнитогорский металлургический комбинат</i> , <i>Мобильные телесистемы</i> , <i>МДМ-банк</i> , <i>Министерство путей сообщения</i> , <i>администрация Воронежа</i> .
<code>"Organization:*"</code>	Прочие обозначения организаций (нереферентные или с неустановленным референтом), например: <i>металлургический комбинат</i> , <i>банки</i> , <i>филиал</i> . Имеют название типа, образованное по шаблону <code>"Organization:*"</code> , например <i>Organization:Part</i> , <i>Organization:Key</i> .
<code>SEMANTIC_TYPE_PERSON_NAMED</code>	Именованные персоны, например: <i>Иванов Александр Петрович</i> , <i>Фкунц С.Е.</i> , <i>Вася</i> .
<code>SEMANTIC_TYPE_PERSON</code>	Прочие обозначения персон и одушевленных предметов, например: <i>политик</i> , <i>экспериментатор</i> , <i>злая собака</i> , <i>брат Иванова</i> .
<code>SEMANTIC_TYPE_GEOGRAPHY_NAMED</code>	Именованные географические объекты, например: <i>Москва</i> , <i>США</i> , <i>Чебоксарский автономный округ</i> , <i>Красное море</i> , <i>Черемушкинский район</i> .

<i>SEMANTIC_TYPE_ARTIFACT_NAMED</i>	Именованные артефакты, товары, например: <i>Nokia 6131, модель Ford Focus, автомобиль марки «ВАЗ-21», водка «Столичная».</i>
<i>SEMANTIC_TYPE_PERSON_POSITION</i>	Названия должностей, например: <i>директор, журналист, исполняющий обязанности министра.</i>
<i>SEMANTIC_TYPE_EVENT</i>	Названия действий, событий, состояний (глаголы и существительные), например: <i>увеличивать, торговаться, аренда земли, болезнь.</i>
<i>SEMANTIC_TYPE_OBJECT</i>	Неодушевленные предметы, например: <i>компьютерная сеть, энергия воздуха.</i>
<i>SEMANTIC_TYPE_FEATURE</i>	Признаки, например: <i>динамический, небрежность, свойство.</i>
<i>SEMANTIC_TYPE_UNKNOWN</i>	Сущности, обозначенные неодушевленными именами существительными и образованными от них словосочетаниями, не отнесенные к другим категориям.
<i>SEMANTIC_TYPE_DATE, SEMANTIC_TYPE_TIME</i>	Даты и обстоятельства времени, например: <i>с 5 по 8 декабря 2003 года, в V веке до нашей эры, четыре года назад, сегодня.</i>
<i>SEMANTIC_TYPE_MONEY</i>	Денежные суммы, например: <i>более 1,2 млрд. долларов, \$10 тыс., в размере 192,5 рублей.</i>
<i>SEMANTIC_TYPE_LATINIC</i>	Латинские слова, встретившиеся в русских предложениях и не отнесенные к другим категориям.
<i>SEMANTIC_TYPE_UNIT</i>	Единицы измерения: метры, килограммы, литры.
<i>“Special:*”</i>	Все прочие виды сущностей, например: номер паспорта, ИНН, номер автомобиля. Согласно установленному соглашению, названия таких типов имеют вид <i>“Special:*”</i> , например <i>Special:Passport</i> . Словари и правила выделения таких сущностей могут опционально включаться в комплект поставки библиотеки, а также могут быть разработаны пользователем самостоятельно.

Приложение 5.

Управление генерацией имен сущностей

Ниже описаны параметры, позволяющие влиять на способ генерации имен сущностей, которые устанавливаются вызовом [FXSetEntityNameGenerationMode](#).

Идентификатор параметра	Значение
FX_PARAM_ENTITYNAME_MAX_LENGTH	Положительное целое число – максимальное допустимое количество слов в имени сущности. Например, при задании значения 2 вместо полного имени сущности <i>последний успех компьютерной лингвистики</i> будет сгенерировано имя <i>успех лингвистики</i> .
FX_PARAM_ENTITYNAME_MAX_COUNT	Положительное целое число – максимальное допустимое количество генерируемых имен сущности. Более одного имени сущности может генерироваться в случае присутствия в предложении однородных членов, в случае неоднозначности слов, при установке значения параметра FX_PARAM_ENTITYNAME_SUBNAMES равного 1.
FX_PARAM_ENTITYNAME_HOMONYMS	0 или 1. При установке значения 1 генерируются все варианты имени сущности для всех слов с неразрешенной омонимией. Например, в результате разбора словосочетания <i>новые техники</i> в неоднозначном предложении <i>новые техники будут эффективны</i> будут построены имена <i>новый техник</i> и <i>новая техника</i> . При установке значения 0 выдается единственный вариант, указанный первым в морфологическом словаре.
FX_PARAM_ENTITYNAME_SUBNAMES	0 или 1. При установке значения 0 для имени сущности генерируется только полное словосочетание, причем зависимые слова в нем сохраняют ту грамматическую форму, которая присутствовала в тексте. Этот режим позволяет получить имена сущностей, оптимальные для представления человеку и максимально приближенные к исходному тексту (параметр FX_PARAM_ENTITYNAME_GENERATION_FLAGS всегда принимает значение NAME_GENERATION_NO_TRANSFORMATION). При установке значения 1 для имени сущности генерируется как полное, так и все входящие в него связные словосочетания. Грамматическая форма зависимых слов определяется значениями параметра FX_PARAM_ENTITYNAME_GENERATION_FLAGS. Этот режим позволяет получить все стандартизованные варианты имени сущности для последующего отождествления при компьютерной обработке, например, при сравнении документов в задачах классификации.
FX_PARAM_ENTITYNAME_RELATION_SET	Битовая маска – дизъюнкция возможных значений атрибута FX_ATTR_RELATIONTYPE (типа связи в семантической сети), которая определяет, связи каких типов будут участвовать в генерации словосочетаний. Рекомендуется использовать пару стандартно определенных значений маски:

	NAME_SIMPLE – генерируются простые именные группы (RELATION_TYPE_ATTR_ADJECTIVE, RELATION_TYPE_ATTR_PARTICIPLE, RELATION_TYPE_ATTR_GENITIVE, RELATION_TYPE_INFINITIVE); NAME_COMPLEX – генерируются глагольные группы и сложные именные группы, включающие предложные формы подчиненных существительных (NAME_SIMPLE, RELATION_TYPE_ARGUMENT, RELATION_TYPE_CIRCUMSTANCE).
FX_PARAM_ENTITYNAME_GENERATION_FLAGS	Битовая маска с нулевым значением (NAME_GENERATION_NO_TRANSFORMATION) или дизъюнкция следующих возможных значений: NAME_GENERATION_DELETE_STOP_WORDS – при генерации имени сущности опускаются малозначимые слова, помеченные в тезаурусе как «удаляемые» (<i>указ нового президента = указ президента</i>). Повышает стандартизацию словосочетаний для их отождествления. NAME_GENERATION_DELETE_COMMON_WORDS – генерируется пустое имя сущности, если в его состав входят только слова, помеченные в тезаурусе как «удаляемые» и «общеупотребимые». Отфильтровывает тематически неопределенную лексику (<i>использование нового свойства</i>), сохраняя тематически окрашенную (<i>использование новой боеголовки</i>). NAME_GENERATION_SYNONIMS – при генерации имен сущностей слова заменяются на синонимы, указанные в тезаурусе. Позволяет повысить стандартизацию имен сущностей для их отождествления. NAME_GENERATION_ADJECTIVE_TO_NOUN – при генерации имен сущностей прилагательные заменяются на существительные, указанные в тезаурусе, с соответствующим изменением формы подчиненных слов в соответствии с русской грамматикой. Позволяет повысить стандартизацию имен сущностей для их отождествления. NAME_GENERATION_VERB_TO_NOUN – при генерации имен сущностей глаголы заменяются на отглагольные существительные, указанные в тезаурусе, с соответствующим изменением формы подчиненных слов в соответствии с русской грамматикой (<i>использовать новое свойство = использование нового свойства</i>). Позволяет повысить стандартизацию имен сущностей для их отождествления. NAME_GENERATION_NOTCHANGE_PLURAL_NUMBER – при генерации имен сущностей подчиненные имена существительные сохраняют форму грамматического числа, указанного в тексте (<i>указ о снижении налогов</i> не будет преобразован в <i>указ о снижении налога</i>). Генерирует более удобные для восприятия человеком словосочетания. NAME_GENERATION_NOTCHANGE_USE_REFERENT – генерация имен сущностей производится с учетом их наиболее полных кореферентных обозначений (<i>этот налог = новый налог на имущество, введение этого налога = введение нового налога на имущество, реформа = реформировать сельское хозяйство</i>). Позволяет сгенерировать более полные словосочетания.

	<i>NAME_GENERATION_SHOW_NEGATION</i> – при наличии отрицания тексте оно включается в имя сущности, при этом вариант без отрицания не выводится. В отсутствие флага отрицания игнорируются. <i>NAME_GENERATION_SHOW_IRREALITY</i> – при наличии сослагательного наклонения в тексте оно включается в имя сущности, при этом вариант без сослагательного наклонения не выводится. В отсутствие флага сослагательное наклонение игнорируются. <i>NAME_GENERATION_NOMINATIVE_NOT_USEROBJECTID</i> – позволяет указывать, что имя сущности в именительном падеже для объектов из XML-описаний составляется из атрибутов объекта в XML-описании, а не из идентификатора объекта в XML-описании (для других падежей всегда составляется из атрибутов объекта). <i>NAME_GENERATION_CASE_SENSITIVE</i> – позволяет указывать, что имя сущности выдается не полностью заглавными буквами, а с учетом регистра букв в XML-описании объектов или в тексте.
FX_PARAM_ENTITYNAME_ MAINWORD_CASE	Флаг позволяет указывать падеж, в который должно быть поставлено имя сущности при вызове FXGetAttributeValue с параметром FX_ATTR_ENTITY_NAME.

Приложение 6.

Описание место-временного контекста

Для каждой сущности семантической сети, соответствующей обозначению события (действия, процесса, состояния), в ряде случаев удастся установить связь с сущностями, соответствующими обозначениям момента времени и места происшествя – место-временному контексту. Изначально сущности-контексты привязываются к сущностям-глаголам на основании синтаксических правил. Далее, на основании особенностей стиля текста (временное согласование глагольных форм в главных клаузах предложений), специальный алгоритм распространения контекста может привязать некоторые контексты к другим глаголам за пределами своего предложения, а также одни контексты к другим. Используемый для этой цели алгоритм распространения контекста чувствителен к стилю текста и ориентирован на тексты стиля СМИ.

Место или время обозначается выделенной из текста сущностью, которая связывается с одной или несколькими сущностями входящей в нее связи специального типа (*RELATION_TYPE_CONTEXT_TIME* или *RELATION_TYPE_CONTEXT_PLACE*). Новые сущности в семантическую сеть не добавляются – одна сущность, обозначающая контекст, может иметь входящие связи указанных типов от сущностей из другого предложения.

Контекст может быть абсолютным (*в 1993 году, 12 мая, в Московской области, недалеко от Смоленской площади*), относительным (*вчера, затем, тремя годами раньше, там, недалеко*).

Также всегда существует актуальный контекст, как правило, соответствующий месту и времени создания текста (или мысленной позиции автора), и обычно известный из внешних источников. Можно его выделить и автоматически из номинативных предложений (*Москва, 13.05.2007*). При этом он маркируется специальным атрибутом и может быть извлечен соответствующими вызовами функций **FXGetFirst** и **FXGetNext** с параметром **FX_INFO_CONTEXT**.

Для относительного контекста точкой отсчета может являться актуальный контекст создания текста (*вчера, пять лет назад*) или некое время, указанное ранее в тексте (*позднее, тремя годами раньше*). Притом обозначение отсылки, зависит от того, на что ссылается относительный контекст:

- На упомянутый в тексте ранее контекст. Отсылка обозначается исходящей из относительного контекста связью типа *RELATION_TYPE_CONTEXT_TIME* или *RELATION_TYPE_CONTEXT_PLACE* с абсолютным или другим относительным контекстом. Так, аналогично прочим сущностям, сущность, обозначающая относительный контекст, может иметь свой собственный контекст (*в мае 1997 ← через шесть месяцев ← потом*);
- На актуальный контекст, не упомянутый в тексте. В этом случае относительный контекст (*вчера, год назад*) просто имеет соответствующую маркировку.

Описание типов выделяемого контекста и их атрибутов приведено ниже.

Показатели места: система описания

Семантические атрибуты

Обстоятельства места, выделяемые в качестве контекста, привязываются к другим сущностям связью типа *RELATION_TYPE_CONTEXT_PLACE*. Они представлены либо наименованиями географических объектов (значение атрибута *FX_ATTR_ENTITY_TYPE* равно *SEMANTIC_TYPE_GEOGRAPHY_NAMED*), либо определенным подклассом наречий (значение атрибута *FX_ATTR_MORPH_SPEECHPART* равно *SP_ADVERB*).

Обстоятельства места подразделяются на классы, которые обозначаются соответствующими значениями атрибута с именем *Semantic* (извлекается в составе строки атрибутов, получаемой запросом *FX_ATTR_ENTITY_ATTRIBUTES*):

- *Semantic=PlaceActual* – актуальное место написания текста, извлекается модулем анализа контекста из номинативного предложения в начальной и конечной частях текста (*Москва, 12 мая 1984 года*). Кроме маркировки таких мест, доступен и их вызов функциями **FXGetFirst** и **FXGetNext** с параметром *FX_INFO_CONTEXT*.
- *Semantic=PlaceAbsolute* – абсолютное обстоятельство места – географический объект в определенном падеже с определенным предлогом или наречие, маркированное в словаре (*в Ульяновской области, рядом с Таганрогом, где-то*).
- *Semantic=PlaceRelative* – относительное обстоятельство места, указывающее на обстоятельство места, упомянутое ранее по тексту (*там, здесь, далеко*). Алгоритм распространения контекста может привязать к такому обстоятельству ранее упомянутое обстоятельство связью типа *RELATION_TYPE_CONTEXT_PLACE* (*в Барнауле ← там*).
- *Semantic=PlaceRelativeActual* – относительное обстоятельство места, указывающее на актуальное место (*Semantic=PlaceActual*). В настоящий момент данный класс обстоятельств пуст (единственным возможным претендентом может быть обстоятельство *у нас*).

Показатели времени: система описания

Семантические атрибуты

1. Обстоятельство образа действия с семантикой времени – значение атрибута *FX_ATTR_ENTITY_TYPE* равно *SEMANTIC_TYPE_TIME_ADVERB*. Включает обозначения таких характеристик событий, которые представляют не его временной контекст (собственно обстоятельства времени *когда это происходит*), а дополнительные временные характеристики события (обстоятельства образа действия *как это происходит*) – длительности, периодичности, возраста (*на 5 минут, 10 лет, часто*).

Такие обстоятельства не выходят за пределы синтаксически ближайшего глагола.

2. Время (обстоятельство времени) – значение атрибута *FX_ATTR_ENTITY_TYPE* равно *SEMANTIC_TYPE_TIME_TIME*. Включает обозначения отрезков времени длительностью менее суток, содержащие в себе числовые величины (12.23, в 3 часа утра, к часу дня по местному времени).

Подобные обстоятельства не распространяются за пределы синтаксически ближайшего глагола, так как с прагматической точки зрения интервалы времени в пределах суток представляются бесполезными, а вероятность ошибки при распространении велика.

3. Дата (обстоятельство времени) – значение атрибута *FX_ATTR_ENTITY_TYPE* равно *SEMANTIC_TYPE_TIME_DATE*. Охватывает даты, интервалы дат и им подобные

обозначения отрезков времени с длительностью более дня, а также с неопределенной длительностью: в V веке нашей эры, 12.08.92, к 5 мая 1998, в 30-е годы, с 1 по 5 ноября 2000г, в марте, иногда, однажды.

Такие обстоятельства времени (за исключением некоторых подклассов) обрабатываются алгоритмом распространения контекста и могут быть распространены на глаголы за пределами своего предложения, привязываясь к ним связью типа `RELATION_TYPE_CONTEXT_TIME`.

Даты подразделяются на классы, которые обозначают соответствующими значениями атрибута `Semantic` (извлекается в составе строки атрибутов, получаемой запросом **FX_ATTR_ENTITY_ATTRIBUTES**):

- `Semantic=TimeActual` – актуальная дата написания текста, извлекается модулем анализа контекста из номинативного предложения в начальной и конечной части текста (*Москва, 12 мая 1984 года; «Коммерсант» от 14.08.92*). Кроме маркировки таких мест атрибутом, их можно получить вызовом функций `FXGetFirst` и `FXGetNext`, с параметром `FX_INFO_CONTEXT`.
- `Semantic=TimeAbsolute` – абсолютное обстоятельство времени: дата конкретной структуры с определенными предлогами или наречие, маркированное в словаре (*5 мая 1998, в 30-е годы, однажды, когда-то*).
- `Semantic=TimeNominative` – дата без предлога, не выступающая в роли обстоятельства времени (*1990 год, 2005-2009 гг, XX век*). Игнорируется алгоритмом распространения контекста.
- `Semantic=TimeRelative` – относительное обстоятельство времени, указывающее на абсолютное или относительное обстоятельство времени, упомянутое ранее по тексту – дата определенной структуры с определенными предлогами или наречие, маркированное в словаре (*затем, пятью годами раньше, через шесть месяцев, 10 лет назад*). Алгоритм распространения контекста может привязать к такому обстоятельству ранее упомянутое обстоятельство связью типа `RELATION_TYPE_CONTEXT_TIME` (*в мае 1997 ← через шесть месяцев ← потом*).
- `Semantic=TimeRelativeActual` – относительное обстоятельство времени, указывающее на актуальную дату (*вчера, сегодня, пять лет назад*). Актуальная дата (`Semantic=TimeActual`) может быть выделена из текста либо получена из внешних источников.
- `Semantic=TimeSituative` – ситуативное обстоятельство времени, выраженное номинативной конструкцией с предлогом либо придаточным предложением с союзом (*после публикации..., в период правления ..., когда начал ...*).
- `Semantic=TimeAttributeAbsolute` – абсолютная дата, стоящая в позиции определения (*указ от 1998 года, поколение 90-х*). Игнорируется алгоритмом распространения контекста.
- `Semantic=TimeAttributeRelative` – относительная дата, стоящая в позиции определения (*бюджет прошлого года*). Игнорируется алгоритмом распространения контекста.
- Пустое значение `Semantic` – прочее (*по сравнению с 1999 годом*). Игнорируется алгоритмом распространения контекста.
- `Semantic=Time` – не ясно, что это за дата. Игнорируется алгоритмом распространения контекста.

Структурные атрибуты

Структурные атрибуты группы *Cape* извлекаются в составе строки атрибутов по запросу `FX_ATTR_ENTITY_ATTRIBUTES`. Помогают стандартизовать даты, написанные в разных форматах, и имеют следующие имена:

- *Cape.Value_WeekDay=**** – день недели: *пятница, в субботу*
- *Cape.Value_Day=**** – число: *17 сентября 1997, 19 декабря*
- *Cape.Value_Month=**** – месяц: *апрель, 15 мая 2000 года*
- *Cape.Value_Year=**** – год: *2004, после 1945 года*
- *Cape.Value_Century=**** – век: *в XX веке, к началу XIX века*
- *Cape.Value_Era=**** – эра: *в 2000 году до нашей эры, в 1500 г. н.э.*
- *Cape.Value_Hour=**** – час: *к 20 часам, 12 часов 15 минут*
- *Cape.Value_Minutes=**** – минуты: *в 15 ч. 45 мин.*
- *Cape.Value_Seconds=**** – секунды: *10 ч 15 мин 45 сек мск*
- *Cape.Value_Prep=**** – предлог, стоящий при дате: *до понедельника, в 2005 г.*
- *Cape.Value_Modifier=**** – прочие модификаторы, стоящие при дате (прилагательные, сущ. *конец, начало* и проч.): *в прошлый вторник, до конца 2006 г., весной 2001 года, в 1-м квартале 1999 г.*

Приложение 7.

Классификация частей речи

Ниже приведены коды частей речи (лексико-грамматических разрядов).

В последнем столбце указан класс парадигмы словоизменения, который определяет возможные грамматические формы слов (см. Приложение 8):

Н – неизменяемое, С – существительное, П – прилагательное, Г – глагол.

Часть речи	Код	Пример	Класс
неизвестна	MO_POS_UNKNOWN		
Вводное слово	MO_POS_INTRODUCTORY	просто	Н
Междометье	MO_POS_INTERJECTION	ах	Н
Предикатив	MO_POS_PREDICATIVE	просто	Н
Предлог	MO_POS_PREPOSITION1	у	Н
Предлог	MO_POS_PREPOSITION2	к	Н
Предлог	MO_POS_PREPOSITION3	в	Н
Предлог	MO_POS_PREPOSITION4	с	Н
Предлог	MO_POS_PREPOSITION5	на	Н
Союз сочинительный	MO_POS_CONJUNCTIONCOORDINATIVE	и	Н
Союз	MO_POS_CONJUNCTION1	что	Н
Союз	MO_POS_CONJUNCTION2	что	Н
Союз	MO_POS_CONJUNCTION3	что	Н
Частица	MO_POS_PARTICLE	же	Н
Наречие	MO_POS_ADVERB	просто	Н
Сущ. сокращенное	MO_POS_ABBREVIATEDNOUN	выкл.	Н
Прил. сокращенное	MO_POS_ABBREVIATEDADJECTIVE	др.	Н
Сокращенное вводное слово	MO_POS_ABBREVIATEDINTRODUCTOR Y	напр.	Н
Обособленная сравнительная степень	MO_POS_COMPARATIVE	побольше	Н
Притяжательное прилагательное	MO_POS_POSSESSIVEADJECTIVE	птичий	П
Мест. муж.	MO_POS_PRONOUN	он	С
Мест. жен.	MO_POS_PRONOUNFEMININE	она	С
Мест. сред.	MO_POS_PRONOUNNEUTER	оно	С
Местоименное прилагательное	MO_POS_PRONOUNADJECTIVE	такой	П
Числ.	MO_POS_NUMERIC	три	С
Собирательное числ.	MO_POS_NUMERICCOLLECTIVE	трое	С
Числ. порядковое	MO_POS_NUMERICORDINAL	третий	П
Аббр. муж	MO_POS_ABBREVIATURE	ВАК	Н
Аббр. жен.	MO_POS_ABBREVIATUREFEMININE	РФ	Н
Аббр. ср.	MO_POS_ABBREVIATURENEUTER	ЦРУ	Н
Имя собств. муж.	MO_POS_NAMEMASCULINE	Саша	С
Имя собств. жен.	MO_POS_NAMEFEMININE	Мария	С
Отчество муж.	MO_POS_MIDDLLENAMEMASCULINE	Васильевич	С
Отчество жен.	MO_POS_MIDDLLENAMEFEMININE	Васильевна	С
Фамилия	MO_POS_LASTNAME	Иванов	П

Географ. название муж.	MO_POS_GEOGRAPHYNAME MASCULINE	Орел	С
Географ. название жен.	MO_POS_GEOGRAPHYNAMEFEMININE	Обь	С
Географ. название сред.	MO_POS_GEOGRAPHYNAMENEUTER	Внуково	С
Географ. название множ.	MO_POS_GEOGRAPHYNAMEPLURAL	Карпаты	С
Глагол переходный несов. вида	MO_POS_VERBIMPERFECT	смотреть	Г
Глагол непереходный несов. вида	MO_POS_VERBINTRANSITIVEIMPERFECT	чихать	Г
Глагол переходный совершенного вида	MO_POS_VERBPERFECT	посмотреть	Г
Глагол непереходный совершенного вида	MO_POS_VERBINTRANSITIVEPERFECT	чихнуть	Г
Глагол переходный двувидовой	MO_POS_VERBTWOWAY	эмитировать	Г
Глагол непереходный двувидовой	MO_POS_VERBINTRANSITIVETWOWAY	эмигрировать	Г
Прил.	MO_POS_ADJECTIVE1	фонетический	П
Прил.	MO_POS_ADJECTIVE2	фонетический	П
Сущ. неодуш. муж.	MO_POS_NOUNINANIMATEMASCULINE	трактор	С
Сущ. одуш. муж.	MO_POS_NOUNANIMATEMASCULINE	зяблик	С
Сущ. одуш.-неодуш. муж.	MO_POS_NOUNANIMATEINANIMATE MASCULINE	кодировщик	С
Сущ. неодуш. жен.	MO_POS_NOUNINANIMATEFEMININE	весна	С
Сущ. одуш. жен.	MO_POS_NOUNANIMATEFEMININE	птичка	С
Сущ. одуш.-неодуш. жен.	MO_POS_NOUNANIMATEINANIMATEFEMININE	матрешка	С
Сущ. неодуш. сред.	MO_POS_NOUNINANIMATENEUTER	озеро	С
Сущ. одуш. сред.	MO_POS_NOUNANIMATENEUTER	дитя	С
Сущ. одуш.-неодуш. сред.	MO_POS_NOUNANIMATEINANIMATE NEUTER	детище	С
Сущ. неодуш. множ.	MO_POS_NOUNINANIMATEPLURAL	ножницы	С
Наименование муж. рода	MO_POS_APPELATION_MderASCULINE	Гарант-Парк-интернет	С
Наименование жен. рода	MO_POS_APPELATION_FEMININE	Формоза	С
Наименование сред. рода	MO_POS_APPELATION_NEUTER	Алеко	С
Наименование множ. числа	MO_POS_APPELATION_PLURAL	Инфо-Технологии	С

Приложение 8.

Описание парадигмы словоизменения

Парадигма словоизменения представляет множество словоформ, которые может принимать слово в определенных грамматических формах. В зависимости от части речи группы слов могут потенциально иметь ограниченный набор грамматических форм. Вследствие этого все парадигмы разделяются на классы таким образом, что каждый класс парадигм описывает изменение слов, относящихся к одному грамматическому разряду.

В принятой здесь системе описания словоизменения существуют три полноценных класса парадигм: существительного, прилагательного и глагола. К четвертому, вырожденному классу, относятся неизменяемые слова. Словоизменение всех прочих частей речи (см. Приложение 7) строится по парадигмам одного из этих классов. Причастия не рассматриваются как самостоятельная часть речи, и все их формы включены в парадигму глагола.

Описание кодов типов парадигм.

<i>MO_PARADIGM_SUBJECT</i>	класс существительных
<i>MO_PARADIGM_DEFINITION</i>	класс прилагательных
<i>MO_PARADIGM_VERB</i>	класс глаголов
<i>MO_PARADIGM_UNCHANGED</i>	неизменяемые слова

Ниже приведен порядок следования грамматических форм в парадигмах словоизменения.

Слова, изменяющиеся по типу существительного (код *MO_PARADIGM_SUBJECT*), могут иметь формы, указанные в таблице ниже.

Грамматическая форма	Пример
имен. падеж, един.	река
род. падеж, един.	реки
дат. падеж, един.	реке
вин. падеж, един.	реку
твор. падеж, един.	рекой, рекою
пред. падеж, един.	реке
2-ой. род. падеж, един.	—
2-ой. пред. падеж, един.	—
им. падеж, множ.	реки
род. падеж, множ.	рек
дат. падеж, множ.	рекам
вин. падеж, множ.	реки
твор. падеж, множ.	реками
пред. падеж, множ.	реках

Слова, изменяющиеся по типу прилагательного (код *MO_PARADIGM_DEFINITION*), могут иметь формы, указанные в таблице ниже.

Грамматическая форма	Пример
имен. падеж, един., муж.	быстрый
род. падеж, един., муж.	быстрого
дат. падеж, един., муж.	быстрому
вин. падеж, един., муж., одуш.	быстрого
вин. падеж, един., муж., неодуш.	быстрый
твор. падеж, един., муж.	быстрым
пред. падеж, един., муж.	быстром
краткая форма, муж.	быстр
имен. падеж, един., жен.	быстрая
род. падеж, един., жен.	быстрой
дат. падеж, един., жен.	быстрой
вин. падеж, един., жен.	быструю
твор. падеж, един., жен.	быстрой, быстрою
пред. падеж, един., жен.	быстрой
краткая форма, жен.	быстра
имен. падеж, един., сред.	быстрое
род. падеж, един., сред.	быстрого
дат. падеж, един., сред.	быстрому
вин. падеж, един., сред.	быстрое
твор. падеж, един., сред.	быстрым
пред. падеж, един., сред.	быстром
краткая форма, сред.	быстро
имен. падеж, множ.	быстрые
род. падеж, множ.	быстрых
дат. падеж, множ.	быстрым
вин. падеж, множ., одуш.	быстрых
вин. падеж, множ., неодуш.	быстрые
твор. падеж, множ.	быстрыми
пред. падеж, множ.	быстрых
краткая форма, множ.	быстры
сравнительная степень	быстрее, быстрее

Слова, изменяющиеся по типу глагола (код *MO_PARADIGM_VERB*), могут иметь формы, указанные в таблице ниже.

Грамматическая форма	Пример
Инф.	перифотграфировать
Возв. инф.	перифотграфироваться
Повел. Накл.	перифотграфируй
Множ. повел.	перифотграфируйте
Буд. 1-е ед.	перифотграфирую
Буд. 1-е мн.	перифотграфируем
Буд. 2-е ед.	перифотграфируешь
Буд. 2-е мн.	перифотграфируете
Буд. 3-е ед.	перифотграфирует
Буд. 3-е мн.	перифотграфируют
Наст. 1-е ед.	перифотграфирую
Наст. 1-е мн.	перифотграфируем
Наст. 2-е ед.	перифотграфируешь
Наст. 2-е мн.	перифотграфируете
Наст. 3-е ед.	перифотграфирует
Наст. 3-е мн.	перифотграфируют
Наст. прич. действ. И. ед.	перифотграфирующий

Наст. прич. страд. И. ед.	перифотографируемый
Наст. дееприч.	перифотографируя
Прош. муж.	перифотографировал
Прош. жен.	перифотографировала
Прош. ср.	перифотографировало
Прош. множ.	перифотографировали
Прош. прич. действ. И. ед.	перифотографировавший
Прош. прич. страд. И. ед.	перифотографированный
Прош. дееприч.	перифотографировав, перифотографировавши
Наст. прич. действ. Р. муж.	перифотографирующего
Наст. прич. действ. Д. муж.	перифотографирующему
Наст. прич. действ. В. муж. одуш.	перифотографирующего
Наст. прич. действ. В. муж. неодуш.	перифотографирующий
Наст. прич. действ. Т. муж.	перифотографирующим
Наст. прич. действ. П. муж.	перифотографирующем
Наст. прич. действ. кф. муж.	–
Наст. прич. действ. И. жен.	перифотографирующая
Наст. прич. действ. Р. жен.	перифотографирующей
Наст. прич. действ. Д. жен.	перифотографирующей
Наст. прич. действ. В. жен.	перифотографирующую
Наст. прич. действ. Т.1 жен.	перифотографирующей
Наст. прич. действ. Т.2 жен.	перифотографирующею
Наст. прич. действ. П. жен.	перифотографирующей
Наст. прич. действ. кф. жен.	–
Наст. прич. действ. И. ср.	перифотографирующее
Наст. прич. действ. Р. ср.	перифотографирующего
Наст. прич. действ. Д. ср.	перифотографирующему
Наст. прич. действ. В. ср.	перифотографирующее
Наст. прич. действ. Т. ср.	перифотографирующим
Наст. прич. действ. П. ср.	перифотографирующем
Наст. прич. действ. кф. ср.	–
Наст. прич. действ. И. мн.	перифотографирующие
Наст. прич. действ. Р. мн.	перифотографирующих
Наст. прич. действ. Д. мн.	перифотографирующим
Наст. прич. действ. В. мн. одуш.	перифотографирующих
Наст. прич. действ. В. мн. неодуш.	перифотографирующие
Наст. прич. действ. Т. мн.	перифотографирующими
Наст. прич. действ. П. мн.	перифотографирующих
Наст. прич. действ. кф. мн.	–
Прош. прич. действ. Р. муж.	перифотографировавшего
Прош. прич. действ. Д. муж.	перифотографировавшему
Прош. прич. действ. В. муж. одуш.	перифотографировавшего
Прош. прич. действ. В. муж. неодуш.	перифотографировавший
Прош. прич. действ. Т. муж.	перифотографировавшим
Прош. прич. действ. П. муж.	перифотографировавшем
Прош. прич. действ. кф. муж.	–
Прош. прич. действ. И. жен.	перифотографировавшая
Прош. прич. действ. Р. жен.	перифотографировавшей
Прош. прич. действ. Д. жен.	перифотографировавшей
Прош. прич. действ. В. жен.	перифотографировавшую

Прош. прич. действ. Т.1 жен.	перифотографировавшей
Прош. прич. действ. Т.2 жен.	перифотографировавшей
Прош. прич. действ. П. жен.	перифотографировавшей
Прош. прич. действ. кф. жен.	–
Прош. прич. действ. И. ср.	перифотографировавшее
Прош. прич. действ. Р. ср.	перифотографировавшего
Прош. прич. действ. Д. ср.	перифотографировавшему
Прош. прич. действ. В. ср.	перифотографировавшее
Прош. прич. действ. Т. ср.	перифотографировавшим
Прош. прич. действ. П. ср.	перифотографировавшем
Прош. прич. действ. кф. ср.	–
Прош. прич. действ. И. мн.	перифотографировавшие
Прош. прич. действ. Р. мн.	перифотографировавших
Прош. прич. действ. Д. мн.	перифотографировавшим
Прош. прич. действ. В. мн. одуш.	перифотографировавших
Прош. прич. действ. В. мн. неодуш.	перифотографировавшие
Прош. прич. действ. Т. мн.	перифотографировавшими
Прош. прич. действ. П. мн.	перифотографировавших
Прош. прич. действ. кф. мн.	–
Наст. прич. страд. Р. муж.	перифотографируемого
Наст. прич. страд. Д. муж.	перифотографируемому
Наст. прич. страд. В. муж. одуш.	перифотографируемого
Наст. прич. страд. В. муж. неодуш.	перифотографируемый
Наст. прич. страд. Т. муж.	перифотографируемым
Наст. прич. страд. П. муж.	перифотографируемом
Наст. прич. страд. кф. муж.	перифотографируем
Наст. прич. страд. И. жен.	перифотографируемая
Наст. прич. страд. Р. жен.	перифотографируемой
Наст. прич. страд. Д. жен.	перифотографируемой
Наст. прич. страд. В. жен.	перифотографируемую
Наст. прич. страд. Т.1 жен.	перифотографируемой
Наст. прич. страд. Т.2 жен.	перифотографируемой
Наст. прич. страд. П. жен.	перифотографируемой
Наст. прич. страд. кф. жен.	перифотографируема
Наст. прич. страд. И. ср.	перифотографируемое
Наст. прич. страд. Р. ср.	перифотографируемого
Наст. прич. страд. Д. ср.	перифотографируемому
Наст. прич. страд. В. ср.	перифотографируемое
Наст. прич. страд. Т. ср.	перифотографируемым
Наст. прич. страд. П. ср.	перифотографируемом
Наст. прич. страд. кф. ср.	перифотографируемо
Наст. прич. страд. И. мн.	перифотографируемые
Наст. прич. страд. Р. мн.	перифотографируемых
Наст. прич. страд. Д. мн.	перифотографируемым
Наст. прич. страд. В. мн. одуш.	перифотографируемых
Наст. прич. страд. В. мн. неодуш.	перифотографируемые
Наст. прич. страд. Т. мн.	перифотографируемыми
Наст. прич. страд. П. мн.	перифотографируемых
Наст. прич. страд. кф. мн.	перифотографируемы
Прош. прич. страд. Р. муж.	перифотографированного
Прош. прич. страд. Д. муж.	перифотографированному
Прош. прич. страд. В. муж. одуш.	перифотографированного

Прош. прич. страд. В. муж. неодуш.	перифотографированный
Прош. прич. страд. Т. муж.	перифотографированным
Прош. прич. страд. П. муж.	перифотографированном
Прош. прич. страд. кф. муж.	перифотографирован
Прош. прич. страд. И. жен.	перифотографированная
Прош. прич. страд. Р. жен.	перифотографированной
Прош. прич. страд. Д. жен.	перифотографированной
Прош. прич. страд. В. жен.	перифотографированную
Прош. прич. страд. Т.1 жен.	перифотографированной
Прош. прич. страд. Т.2 жен.	перифотографированною
Прош. прич. страд. П. жен.	перифотографированной
Прош. прич. страд. кф. жен.	перифотографирована
Прош. прич. страд. И. ср.	перифотографированное
Прош. прич. страд. Р. ср.	перифотографированного
Прош. прич. страд. Д. ср.	перифотографированному
Прош. прич. страд. В. ср.	перифотографированное
Прош. прич. страд. Т. ср.	перифотографированным
Прош. прич. страд. П. ср.	перифотографированном
Прош. прич. страд. кф. ср.	перифотографировано
Прош. прич. страд. И. мн.	перифотографированные
Прош. прич. страд. Р. мн.	перифотографированных
Прош. прич. страд. Д. мн.	перифотографированным
Прош. прич. страд. В. мн. одуш.	перифотографированных
Прош. прич. страд. В. мн. неодуш.	перифотографированные
Прош. прич. страд. Т. мн.	перифотографированными
Прош. прич. страд. П. мн.	перифотографированных
Прош. прич. страд. кф. мн.	перифотографированы

Приложение 9

Классификация падежей, родов и указаний на число

Ниже приведены обозначения и коды падежей – значения атрибута `FX_ATTR_MORPH_CASE`.

Падеж	Обозначение	Код	Пример
Падеж не определен	GRAMMAR_CASE_UNDEFINED	-1	
Именительный	GRAMMAR_CASE_NOMINATIVE	0	река
Родительный	GRAMMAR_CASE_GENITIVE	1	реки
Дательный	GRAMMAR_CASE_DATIVE	2	реке
Винительный	GRAMMAR_CASE_ACCUSATIVE	3	реку
Творительный	GRAMMAR_CASE_INSTRUMENTAL	4	рекой, рекою
Предложный	GRAMMAR_CASE_PREPOSITIONAL	5	реке

Ниже приведены обозначения и коды родов – значений атрибута `FX_ATTR_MORPH_GENDER`.

Род	Обозначение	Код	Пример
Неопределенный род	GRAMMAR_GENDER_UNDEFINED	0	Примеры таких слов, в частности, наречия
Мужской	GRAMMAR_GENDER_MASCULINE	1	Вокзал
Женский	GRAMMAR_GENDER_FEMININE	2	Пристань
Средний	GRAMMAR_GENDER_NEUTRAL	3	Судно
Род не установлен			

Такая морфологическая характеристика сущности, как множественное и единственное число, описывается значениями `GRAMMAR_NUMBER_PLURAL` и `GRAMMAR_NUMBER_SINGULAR` атрибута `FX_ATTR_MORPH_NUMBER` (коды 1 и 0 соответственно).

Приложение 10

Словари, поставляемые в стандартной конфигурации RCO FX Ru

В расположенной ниже таблице приводится описание используемых библиотекой RCO FX Ru словарей.

Имя файла словаря	Назначение	Семантический тип	Примеры
idioms_pronoun_from.dct	Собирает в одно слово определительные и неопределенные местоимения с последующим предлогом «из», приводит к соответствующему синониму. Требуется для улучшения синтаксического анализа.	Idioms: PronounFrom	один из, любой из, некоторые из
idioms_adverb.dct	Собирает идиоматичные наречные выражения в одну сущность. Требуется для улучшения синтаксического и семантического анализа.	undefined: undefined	без задних мыслей, в действительности, без усталости, в лучшем случае
idioms_adverbtime.dct	Выделяет идиоматичные обстоятельства времени. Требуется для улучшения синтаксического и семантического анализа.	Time:Adverb	вечерком, в молодости, год за годом, за это время
idioms_conjunction.dct	Собирает многословные союзы. Требуется для улучшения синтаксического анализа	undefined: undefined	в то время(,) как, если бы не, до тех пор(,) пока
idioms_deleted.dct	Выделяет идиомы, от удаления которых смысл не меняется и которые можно исключить из семантико-синтаксического разбора.	Idioms:Deleted	так называемый, тот или иной, все тот же, вот уже
idioms_intro.dct	Выделяет вводные слова и выражения. Требуется для улучшения синтаксического и семантического анализа	Introductory: Introductory	мягко говоря, как сообщалось ранее, на самом деле
idioms_particle.dct	Собирает многословные частицы в одну сущность, присваивает частицам семантические и грамматические атрибуты. Требуется для улучшения синтаксического, семантического, дискурсивного анализа.	undefined: undefined	чуть было не, не только и не столько, пускай и, все же
idioms_misc.dct	Выделяет прочие идиоматичные выражения, при необходимости присваивает им грамматические атрибуты. Требуется для улучшения синтаксического и семантического анализа.	undefined: undefined	представлять собой, сам себя, все что угодно, какой бы ни было

geoplace_name.dct	Географические наименования. Как правило требуется для выделения многословных географических наименований без ключевого слова, так как многие однословные наименования имеются в морфологическом словаре и не требуют дублирования, а наименования с ключевыми словами выявляются правилами, или для задания синонимов (столица России MSYN Москва).	Geoplace:Name	Дальний Восток, Коми АССР, Васильевский остров, сектор Газа
geoplace_city.dct	Список городов РФ (неомонимичных другим словам).	Geoplace:Name	Абакан, Александровск-Сахалинский, Боровичи, Саратов, Гусь Хрустальный
geoplace_subject.dct	Субъекты РФ.	Geoplace:Name	Адыгея, Алтай, Саха (Якутия), Северная Осетия - Алания
geoplace_keypost.dct	Выделяет ключевые слова для географии (полные), стоящие после названия. Используется в правилах на выделение географических наименований.	Geoplace:KeyPost	область, регион, автономный округ
person_position.dct	Выделяет должности и профессии. Используется для семантического анализа.	Person:Position	еврокомиссар, горнорабочий, вице-мэр, главный бухгалтер
person_position_dignity.dct	Словарь духовных титулов. Используется для семантического анализа.	Person:Position	архидиакон, иерей, викарий, брахман
person_position_herorank.dct	Словарь геройских званий. Используется для семантического анализа.	Person:Position	герой Советского Союза, герой труда, герой России
person_position_military_rank.dct	Словарь воинских званий. Используется для семантического анализа.	Person:Position	адмирал, генерал армии, капитан-лейтенант
person_position_rank.dct	Словарь почетных званий. Используется для семантического анализа.	Person:Position	народный артист, заслуженный врач, кавалер ордена Святого апостола Андрея Первозванного
person_position_sport_title.dct	Словарь спортивных титулов. Используется для семантического анализа.	Person:Position	олимпийский чемпион, чемпион мира, серебряный призер

person_position_sportprof.dct	Словарь спортивных профессий. Используется для семантического анализа.	Person:Position	нападающий, биатлонист, борец сумо, бомбардир
person_position_titles.dct	Словарь аристократических титулов. Используется для семантического анализа.	Person:Position	ага-хан, виконт, великий князь, эсквайр, лорд
organization_key.dct	Словарь ключевых слов для организаций. Используется в правилах для выделения наименований организаций.	Organization:Key	завод, компания, агентство
organization_key_to_name.dct	Словарь омонимичных ключевых слов, которые могут являться частью названия организации. Используется в правилах для выделения наименований организаций.	Organization:KeyToName	группа, центр, партия
organization_keyabbr.dct	Словарь ОПФ организаций. Используется в правилах для выделения наименований организаций.	Organization:KeyAbbr	ООО, ЗАО, автономная некоммерческая организация
organization_inokey.dct	Словарь ключевых слов для иностранных организаций. Используется в правилах для выделения наименований организаций.	Organization:InoKey	Лтд., Инк., Ллс
organization_part.dct	Словарь ключевых слов для подразделений организаций. Используется при семантическом анализе.	Organization:Part	Филиал, дочерняя компания, отдел, представительство
organization_noname.dct	Словарь ключевых слов для организаций, в названии которых не допускается употребление имени собственного. Используется в правилах для выделения наименований организаций.	Organization:Noname	таможня, областной суд, городская администрация
organization_full.dct	Словарь полных наименований организаций. Как правило требуется для выделения наименований организаций, которые по какой-либо причине сложно выявить общими правилами (например, не содержат ключевого слова в названии).	Organization:Full	Эйр Франс, Скай Линк, РАО "ЕЭС России", Би Лайн
organization_latinic.dct	Словарь названий иностранных организаций, пишущихся латиницей. Задаёт терминам необходимые грамматические атрибуты.	undefined:undefined	NASA, Microsoft, Hilton, Russia Today
preposition_about.dct	Словарь предлогов, выражающих содержание. Используется при семантико-синтаксическом анализе.	Preposition:About	о, про, насчет, касательно
preposition_accordance.dct	Словарь предлогов, выражающих сравнение, согласование, соотносённость с другим объектом. Используется при семантико-синтаксическом анализе.	Preposition:Accordance	в соответствии с, по сравнению с, наподобие
preposition_anti.dct	Словарь предлогов, выражающих противопоставление, антагонизм. Используется при семантико-синтаксическом анализе.	Preposition:Anti	наперекор, вопреки, в противоположность

preposition_exception.dct	Словарь предлогов, выражающих исключение. Используется при семантико-синтаксическом анализе.	Preposition: Exception	без, кроме, исключая
preposition_instrument.dct	Словарь предлогов, выражающих способ, инструмент действия. Используется при семантико-синтаксическом анализе.	Preposition: Instrument	при помощи, посредством
preposition_limit.dct	Словарь предлогов, выражающих предел, ограничение. Используется при семантико-синтаксическом анализе.	Preposition: Limit	сверх, менее чем на, более чем в
preposition_location.dct	Словарь пространственных предлогов. Используется при семантико-синтаксическом анализе.	Preposition: Location	далеко от, рядом с
preposition_misc.dct	Словарь предлогов смешанной семантики. Используется при семантико-синтаксическом анализе.	Preposition: Misc	на сумму, в интересах, под предлогом
preposition_purpose.dct	Словарь предлогов, выражающих цель. Используется при семантико-синтаксическом анализе.	Preposition: Purpose	в целях, ради
preposition_reason.dct	Словарь предлогов, выражающих причину. Используется при семантико-синтаксическом анализе.	Preposition: Reason	вследствие, по причине
preposition_situation.dct	Словарь предлогов, выражающих ситуативную характеристику, условия. Используется при семантико-синтаксическом анализе	Preposition: Situation	в зависимости от, в условиях, по случаю
preposition_time.dct	Словарь временных предлогов. Используется при семантико-синтаксическом анализе.	Preposition: Time	в годы, по окончании
preposition_way.dct	Словарь предлогов, выражающих направление действия. Используется при семантико-синтаксическом анализе.	Preposition: Way	по направлению к, вслед за, в направлении
terms.dct	Пользовательский словарь для занесения произвольных терминов с целью объединения нескольких слов в один термин без присвоения семантического типа либо для задания грамматических характеристик слова. Используется для улучшения синтаксического и морфологического анализа.	undefined: undefined	IT-технология, GSM-лицензия, экологически чистый, средства массовой информации, ФЗ РФ, доллар США
events.dct	Служит для выделения событийных существительных – известных именованных событий либо нестандартных наименований (напр., содержащих латинские слова). Требуется для улучшения морфологического, синтаксического и семантического анализа.	Event: Event	День Победы, Холодная война, Формула-1, PR-компания, IPO, vip-вечеринка
latinic.dct	Содержит слова на латинице, часто встречающиеся в русскоязычных текстах и не относящиеся к семантическим категориям других словарей. Задает терминам грамматические характеристики. Требуется для улучшения морфологического и синтаксического анализа.	Unknown: Unknown	Internet, SMS, Linux, Master Card, e-mail

metro.dct	Словарь станций Московского метрополитена. Используется при семантико-синтаксическом анализе	Geoplace:Name	Александровский сад, ВДНХ, Улица 1905 года
person_vocative.dct	Словарь распространенных обращений. Используется для улучшения семантико-синтаксического анализа, чтобы обращения ошибочно не становились именами.	Person:Vocative	Дорогой, Уважаемый, Многоуважаемый
cars.dct	Содержит названия марок машин, состоящие из нескольких слов, а также некоторые однословные неомонимичные названия.	Artifact:Cars	мерседес, лэнд ровер, ролс ройс
cars_latin.dct	Словарь марок машин на латинице (многословные и однословные неомонимичные). Присваивает терминам грамматические атрибуты.	Artifact:Cars	acura, bentley, chevrolet, hyundai accent, land rover
geoplace_country.dct	Словарь стран мира.	Geoplace:Name	Россия, Абхазия, Содружество Багамских Островов, Республика Гвинея-Бисау
geoplace_inocity.dct	Словарь иностранных городов (наиболее крупных и известных). Содержит порядка 540 терминов.	Geoplace:Name	Орландо, Пальма-де-Майорка, Нью-Йорк, Ливерпуль, Куала-Лумпур, Катманду

Приложение 11

Фильтры, поставляемые в стандартной конфигурации RCO FX Ru

В расположенной ниже таблице приводится описание используемых библиотекой RCO FX Ru фильтров.

Имя файла словаря	Назначение	Семантический тип	Примеры	Комментарии
prep_date.flt	Содержит перечень временных предлогов, встречающихся в датах. Используется в правилах на даты.	Preposition:Date	в, до, с, в период, в середине	Используется в date*.rul
month.flt	Содержит перечень месяцев во всех падежах. Используется в правилах на даты.	Date:Month	январь, февраля, марте	Используется в date*.rul
month_nom.flt	Содержит перечень месяцев только в именительном падеже. Используется в правилах на даты.	Month:Nominative	январь, февраль, март	Используется в date*.rul
month_gen.flt	Содержит перечень месяцев только в родительном падеже. Используется в правилах на даты.	Month:Genetive	января, февраля, марта	Используется в date*.rul
time_item_full.flt	Содержит полный список единиц времени. Используется в правилах на даты.	Date:TimeItemFull	год, месяц, день, секунда, мгновение, миг, период, промежуток	Используется в date*.rul
time_item_brief.flt	Содержит только некоторые обозначения единиц времени. Используется в правилах на даты.	Date:TimeItemBrief	год, месяц, день, секунда	Используется в date*.rul
time_item_pr1.flt	Содержит единицы времени, встречающиеся только с определенными предлогами в определенных конструкциях. Используется в правилах на даты (AdverbTime1_Rule, ЗА ДО Rule, ЧЕРЕЗ Rule).	Date:TimeItemPr1	лет, весна, вечер	Используется в date*.rul
time_item_dur.flt	Содержит единицы времени, которые могут обозначать длительности. Используется в правилах на даты.	Date:TimeItemDur	лет, год, день, десятилетие	Используется в date*.rul

time_item_pr5.flt	Содержит единицы времени, встречающиеся только с определенными предложениями в определенных конструкциях. Используется в правилах на даты (EveryRule, AGOAct_Rule, AGORel_Rule).	Date:TimeItemPr5	лет, год, день, десятилетие	Используется date*.rul	в
time_item_acc.flt	Содержит единицы времени в винительном падеже. Используется в правилах на даты.	Date:DurationAcc	год, года, годы, минуту, секунду	Используется date*.rul	в
time_item_prepadj.flt	Содержит единицы времени, встречающиеся с временными прилагательными (прошлый, настоящий, будущий и т.д.). Используется в правилах на даты.	Date:TimeItemPrepAdj	весна, год, день	Используется date*.rul	в
adj_time.flt	Содержит прилагательные, встречающиеся с единицами времени. Используется в правилах на даты.	Adjective:Time	сегодняшний, настоящий, ближайший, прошлый, будущий	Используется date*.rul	в
adj_time1.flt	Содержит прилагательные, встречающиеся с единицами времени. Используется в правилах на даты.	Adjective:TimeRel	тот, некоторый, предыдущий, последний	Используется date*.rul	в
geo_keyprev_full.flt	Содержит ключевые слова для географических названий, полные, стоящие в начале. Используется для выделения географических наименований.	Geo:KeyPrev_full	город, республика, село, деревня	Используется geoplace.rul	в
geo_keyprev_brief.flt	Содержит ключевые слова для географических названий, краткие, стоящие в начале. Используется для выделения географических наименований.	Geo:KeyPrev_brief	г., гор., д., дер., пос., п., п.г.т.	Используется geoplace.rul	в
geo_keypost.flt	Содержит ключевые слова для географических названий, краткие, стоящие в конце. Используется для выделения географических наименований.	Geo:KeyPost	обл., р-н, ао, респ.	Используется geoplace.rul	в
organization_latinic.flt	Содержит латинские названия брендов. Используется в правиле для выделения именованных артефактов.	Organization:Latinic	Panasonic, Ford, Toshiba, Ecco	Используется artifact.rul	в
artifact_key.flt	Содержит ключевые слова для артефактов. Используется в правиле для выделения именованных артефактов.	Artifact:Key	авиалайнер, автобус, видеокамера, ветчина, вино, шампунь	Используется artifact.rul	в

artifact_intellectual.flt	Содержит ключевые слова для продуктов интеллектуальной деятельности. Используется в правиле для выделения именованных артефактов.	Artifact: KeyIntellectual	вальс, картина, кинофильм, книга, роман	Используется artifact.rul	в
geo_prep.flt	Содержит существительные, которые могут быть частью составных пространственных предлогов (на берегу чего-либо, на границе чего-либо, в окрестностях чего-либо). Используется в правиле на выделение данных предлогов.	Geo:Prep	берегу, берегах, границе, границах, окрестности, окрестности	Используется idioms3.rul	в
for_prepos_gen.flt	Содержит существительные, которые могут быть частью составных предлогов смешанной семантики (в области чего-либо, в духе чего-либо, в форме чего-либо). Используется в правиле на выделение данных предлогов.	For:PreposGen	духе, области, форме, силу	Используется idioms.rul	в
metro.flt	Содержит перечень станций Московского метрополитена, омонимичных другим словам. Используется для выделения названий соответствующих станций.	Metro:Name	Алтуфьево, Аэропорт, Белорусская, Беляево, Динамо	Используется metro.rul	в
appellation_prefix.flt	Содержит приставки, с которых часто начинаются наименования организаций. Используется в правиле для выделения названий организаций.	Prefix:Appellation	ГОС, МИН, РОС, ТРАНС	Используется appellation.rul	в
cars.flt	Содержит однословные названия марок машин. Используется в правилах на выделение марок автомобилей	Artifact:Cars	акура, Альфа, ауди, семерка, Ягуар, Москвич, audi, BMW, citroen	Используется cars.rul	в
adjective_unit.flt	Содержит прилагательные, образованные от единиц измерения. Используется для распознавания прилагательных вида: 10-метровый, 5-литровый	Adjective:Unit	секундный, сантиметровый, килограммовый, литровый	Используется tokenization.rul	в
kladr_STREET.flt	Содержит сокращенные ключевые слова, обозначающие улицу, выгруженные из КЛАДРа. Используется в правилах для распознавания названий улиц.	AddrElemType: STREET_SHORT	бр, б-р, бульв., пер., пер-к, п-к, просп., пр-т	Используется address1.rul	в
kladr_STREET_full.flt	Содержит полные ключевые слова, обозначающие улицу, выгруженные из КЛАДРа. Используется в правилах для распознавания названий улиц.	AddrElemType: STREET_FULL	аллея, бульвар, переулок, площадь, проезд	Используется address1.rul	в

Приложение 12

Правила, поставляемые в стандартной конфигурации RCO FX Ru

В расположенной ниже таблице приводится описание используемых библиотекой RCO FX Ru правил.

Имя файла	Назначение	Наименование правил	Описание правил	Семантический тип	Примеры	Комментарии
service.rul	Содержит полезные макросы, использующиеся в других правилах.				Содержит макросы, выделяющие запятую, любой предлог, кавычки, любую сущность и т.д.	Должно всегда стоять первым, т.к. другие правила опираются на него.
phone.rul	Содержит правила для выделения телефонных номеров.	Phone_Rule1	Опирается на ключевое слово (тел., факс, моб. и т.д.).	Special: PhoneNumber	тел.: 123-34-34	
		Phone_Rule2	Опирается на код страны или города.		+7 956 123456, (123) 123-9456	
		DigitsOnly_Rule	Опирается на числовую последовательность.		8 921 1234567, 1-123-123-1234	
inn.rul	Содержит правила для выделения ИНН, опирающиеся на контрольное число.	INN10_Rule	Выделяет 10-значные ИНН.	Special:INN	ИНН 6107200243, 4218414767	
		INN12_Rule	Выделяет 12-значные ИНН.			
ogrn.rul		OGRN_Rule	Выделяет ОГРН.	Special:OGRN	ОГРН 1037700015924, 1107846200951	

		OGRNIP_Rule	Выделяет ОГРНИП.	Special:OGRNIP		
okpo.rul	Содержит правило для выделения ОКПО, опирающееся на контрольное число.	OKPO_Rule		Special:OKPO	ОКПО 68945620, 7951448362	
vehicle_number.rul	Содержит правила для выделения номеров автомобилей	VehicleRegistrNumber_Rule	Опирается на ключевое слово, номер начинается с буквы.	Special:VehicleNumber	гос. номер А 123 АВ, гос. номер А123АВ	
		VehicleRegistrNumber_Rule1	Опирается на последовательность букв и цифр с пробелом, номер начинается с буквы.		А 123 АА 77	
		VehicleRegistrNumber_Rule2	Ключевое слово есть, номер написан без пробела, номер начинается с цифр.		гос. номер 1234АВ77	
		VehicleRegistrNumber_Rule3	Ключевое слово и номер написаны слитно, без пробелов, номер начинается с цифр.		гос.№1234АВ77	
registr_number.rul	Содержит правило для выделения регистрационных номеров.	RegistrNumber_Rule		Special:RegistrNumber	РЕГИСТРАЦИОННЫЙ НОМЕР 4-01-00099-А	
e-mail.rul	Содержит правила для выделения адресов электронной почты.	EMail_Rule, EMail_Rule1, EMail_Rule2, EMail_Rule3		Special:Email	SALES@AKM.RU, INVEST@MDMBANK.COM, INFO@FASMO.ARBITR.RU	

inn_ wrong.rul	Содержит правила для выделения ИНН с неправильным контрольным числом, опирающиеся на ключевое слово.	WrongINN10_Rule	Выделяет 10-значные ИНН с неверным контрольным числом.	Special:INN	ИНН 0205004246	
		WrongINN12_Rule	Выделяет 12-значные ИНН с неверным контрольным числом.		ИНН 123456789012	
		WrongINN_Rule	Выделяет ИНН прочей длины.		ИНН 12345670	
okpo_ wrong.rul	Содержит правило для выделения ОКПО с неверным контрольным числом.	WrongOKPO_Rule		Special:OKPO	ОКПО 1234556	
ogrn_ wrong.rul	Содержит правила для выделения ОГРН и ОГРНИП с неверным контрольным числом.	WrongOGRN_Rule	Выделяет неправильный ОГРН.	Special:OGRN	ОГРН 1234567	
		WrongOGRNIP_Rule	Выделяет неправильный ОГРНИП.	Special:OGRNIP	ОГРНИП 123456788	
date_of_ birth_ pre.rul	Содержит правило для выделения года рождения, где год обозначен только двумя цифрами.	DateOfBirth_Rule4		Special:DateOfBirth	88 года рождения	Должно подключаться до правил на даты.
appellation. rul	Содержит вспомогательные правила для выделения названий некоторых организаций.	AppellationM_Rule, AppellationF_Rule, AppellationN_Rule		не присваивают семантический тип.	Росбалт, Госрегюст, Росприроднадзор	Опираются на фильтр.
		AirlineAppellationM_Rule, Airline	Помогают выделять названия некоторых авиакомпаний.	не присваивают семантический тип.	Авиакомпания Сибирь, а/к Ямал	

		AppelationF_Rule, Airline AppelationN_Rule				
		Science Academy_Rule1, Science Academy_Rule2, Science Academy_Rule2_1	Выделяют названия академий наук.	Organization:Full	АН России, Российская АН, Национальная АН Беларуси	
		UnionRule	Определяет случаи, когда слово «объединение» обозначает организацию.	Organization:KeyToName	объединения «Мебельторг»	
		Department Rule1	Собирает управления и головную организацию в единую сущность.	Organization:Full	Амурского Управления МЧС	
		CentralBank Rule1, CentralBank Rule2	Выделяет центробанки.	Organization:Full	Европейский Центробанк, Европейский ЦБ	
numeric.rul	Содержит правила для выделения числительных.	NumericRule1, NumericRule2, DigitRule2	Числительные, выраженные словами.	Numeric:Numeric	ПЕРВЫЙ, около тысячи, пять, ДВАДЦАТЬ ВТОРОЕ	
		DigitRule1, DigitRule1_2, DigitRule3, DigitRule4	Числительные, выраженные цифрами.	Numeric:Digit	2, 2.381 2381, около 30	

numeric1.rul	Собирает подряд идущие числительные в одну сущность.	Complex NumericRule1		Numeric:Numeric	первых 38305, 6 из 8761, 10 первых, тысяч 50-100	Опирается на правила numeric.rul
date1.rul	Содержит основные правила для выделения дат.	WeekDayRule	Дни недели.	Time:Date	в пятницу, в этот четверг	Опирается на numeric*.rul и на фильтры
		MonthOnlyRule, NightDateRule, MonthDateRule, MonthShortDate Rule	Месяца.	Time:Date	в июне, в середине октября, в конце апреля	
		CenturyRule, CenturyNom Rule, CenturyGenAbs Rule, CenturyGenRel Rule	Века.	Time:Date	XX века, 19 века, в XXI веке	
		RelativeActual DateRule, RelativeDate Rule, RelativeActual DateRule2, EveryRule, RelativeNome Rule	Относительные даты.	Time:Date	В прошлом месяце, На этой неделе, В последние годы, каждый день, каждые два года	
		YearRule, YearEraRule, YearInterval Rule, DayMonth YearRule, YearNomRule, YearGenRule1,	Даты, содержащие год.	Time:Date	в 2012 г., 1959- 1983 гг., 1959-1983 гг., В декабре 2011 года, в начале 2003 года, в 1996-м году	

		YearGenRule2, OrdYearRule, YearDashRule				
		DigitDateRule1, DigitDateGen Rule1, DigitDateRule2, DigitDate GenRule2	Числовые даты.	Time:Date	26.09.11, от 09.04.2012 г., 15.02.2012 года	
		TimeRule1, TimeRule2, TimeRule3	Время.	Time:Time	В 17:30 мск, в 15 часов 46 минут, в 17:20 по местному времени, 10:20, к 12 часам дня	
date1_1. rul	Собирает даты с невременными предлогами.	MonthAnyPrep Rule, CenturyAny PrepRule, YearAnyPrep Rule, UniRule		Time:Date	по сравнению с февралем, за 2011 г., на 2012-2014 годы, на декабрь	Опирается на date1.rul
date_row. rul	Склеивает части дат в одну дату.	AbsDateRule, RelDateRule, GenDate AbsoluteRule, GenDate RelativeRule		Time:Date	с 2001 года по 2009 год, в четверг 10 мая, 7 мая в 9.50, С мая 2004 г. по ноябрь 2005 г., В ночь на 9 мая 2010 года, с 27 по 31 августа 2011 года, С сентября - октября прошлого года, в марте сего года	Опирается на date1.rul
date2.rul	Содержит правила для выделения дат.	DateRule	Склеивает цепочку дат в одну сущность.	Time:Date	за январь-март текущего года, на 1 января текущего года, на начало 2012 г. и на 1 мая 2012 г., за период с 21 декабря 2009 года по 15	Опирается на date1.rul, date1_1.rul, date_row.rul

					января 2010 года, за 2010-2011 гг. и I квартал 2012 г.	
		DurationRule1, DurationRule2, DurationRule3, ПОСЛЕДНИЕ_ ЧИСЛ_Rule, В_ ПОСЛЕДНИЙ_ МОМЕНТ_Rule, Earlier_Rule	Наречные обороты со значением времени.	Time:Adverb	последние 5 лет, часом раньше, В течение года, несколько лет подряд, на некоторое время, за первые два дня	Опираются на фильтры, numeric*.rul, date1.rul
		ЧЕРЕЗ_Rule, ЗА_ДО_Rule, ЧИСЛ_ ПОСЛЕДНИХ_ Rule, ПРЕДЛОГ_ ПРИЛАГ_Act_ Rule, ПРЕДЛОГ_ ПРИЛАГ_ Rel_Rule, В_ЭТОТ_ МОМЕНТ_Rule, В_ДАННЫЙ_ МОМЕНТ_Rule, ПРЕДЛ_ПРИЛ_ ВРЕМЯ_Rule, ПРЕДЛ_ПРИЛ_ ВРЕМЯ_Rel_ Rule, SOVIET_Rule, Season_Rule, SeasonAdv_Rule,	Относительные даты.	Time:Date	через год, за два дня до, пять последних дней, в настоящее время, в ближайшие 5 лет, в тот год, в советский период, ранней весной, этим вечером, предыдущим вечером, год назад, пятью годами раньше, до сих пор, до недавнего времени	Опираются на фильтры, numeric*.rul, date1.rul

		ActualDay_Rule, RelativeDay_ Rule, AGOAct_Rule, AGORel_Rule, TimeIdioms_ Rule, CO_ДНЯ_НА_ ДЕНЬ Rule				
		Midday_Rule	Временные выражения со словами «полдень», «полночь».	Time:Time	около полудня	
date2_1.rul	Содержит правила для выделения дат.	AdverbTime1_ Rule, AdverbTime2_ Rule	Наречные обороты со значением времени.	Time:Adverb	все пять лет, 10 лет, по пять лет, с каждым днем, после четырех лет, более двух месяцев, почти два года	Опираются на фильтры, numeric*.rul, date*.rul
		TimeOfDay_Rule	Время суток с прилагательными «ранний», «поздний».	Time:Time	ранним вечером, поздней ночью	
date_of_birth.rul	Содержит правила для выделения дат рождения.	DateOfBirth_ Rule1, DateOfBirth_ Rule2, DateOfBirth_ Rule3, DateOfBirth_ Rule4		Special: DateOfBirth	1961 года рождения, 12.08.1980 г.р., 15 февраля 1974 г.р.	Опирается на date*.rul
artifact.rul	Правила для выделения именованных артефактов (марки, модели различных товаров).	Artifact_Rule1, Artifact_Rule2, Artifact_Rule3, Artifact_Rule4, Artifact_Rule5, Artifact_Rule6		Artifact:Name	самолетов ТУ- 154, конфет «Привет», водки «Медаль 0,75», пива «Хайнекен», самолет Ан-2, телефонами Samsung, лайнера «Сухой Суперджет 100», фильм	Опирается на фильтры

					«Служу Советскому Союзу», устройство Virtual Steelhead 150	
idioms.rul	Содержит правила для выделения устойчивых оборотов различных частей речи. Требуется для улучшения семантико-синтаксического анализа.		Именованный объект.	Object: Object	группа «А»	Опирается на numeric*.rul
			Вводный оборот.	Introductory: Introductory	в итоге, в целом, более того, благодаря этому, между тем	
			Предлог смешанной семантики.	Preposition:Misc	на благо, в размере, в области, по вопросу, на смену	
			Пространственный предлог.	Preposition:Place	в районе	
			Предлог, выражающий содержание.	Preposition: About	на тему,	
			Предлог, выражающий способ действия.	Preposition: Instrument	путем	
			Пространственный предлог.	Preposition: Location	сбоку, сзади, внутри, среди	
			Предлог, выражающий направление.	Preposition: Way	вслед, навстречу, мимо, параллельно	
			Предлог, выражающий сравнение, согласование, соотнесенность с другим объектом.	Preposition: Accordance	взамен, относительно, подобно, вроде	
			Предлог, выражающий предел, ограничение.	Preposition: Limit	свыше	
			Наречный оборот со значением времени.	Time:Date	в прошлом	
			Наречный оборот со значением следования во времени.	Time:Adverb	после того, перед тем	

			Количественное наречие.	Adverb:Quantity	5 раз подряд, в 10 раз	
idioms2.rul	Содержит правила для выделения устойчивых оборотов различных частей речи. Требуется для улучшения семантико-синтаксического анализа.		Название закона.	Low:Name	Закону «Об оценочной деятельности»	Должно следовать после idioms.rul
			Ученая степень.	Person:Position	кандидат технических наук	
numeric2.rul	Содержит правила для выделения процентов, единиц измерения, денежных сумм.	PersentNumeric Rule1, PersentDigit Rule1, PersentDigit Rule1	Проценты, единицы измерения.	Numeric:Persent	7%, до 25%, более 90%, 51 процента, около 1 м, в 600 метрах	Опирается на numeric.rul, numeric1.rul
		MoneyRule1, MoneyRule2, MoneyRule3	Денежные суммы.	Numeric:Money	за \$1,45 млрд., 300 рублей, 15р, 40 миллионов рублей, до 20.510 руб.	
codex_partition.rul	Собирает разделы кодекса.	Statya_Rule, ChastStatya_Rule	Статья.	Law:Statya	статья 1, ст. 3, статьями 176 , 284 - 289	Опирается на numeric.rul, numeric1.rul
		Chast_Rule	Часть.	Law:Chast	частями 5 и 6, ч. 3	
		Punkt_Rule	Пункт.	Law:Punkt	пункта 2.4, п. а, п. 1	
		Podpunkt_Rule	Подпункт.	Law:Podpunkt	подпункт 3	
codex_assemble.rul	Собирает цепочку из разделов кодекса в одну сущность.	CodexAssemble_Rule1, CodexAssemble_Rule2		Special:Codex	части 10 статьи 41.8; статьи 65 АПК; п. 3 ч. 1 ст. 287	Опирается на codex_partition.rul
person.rul	Содержит правило для выделения титулованных персон, в имени	VIP_1		Person:Full	Алексий II, Петр I	Опирается на numeric.rul, numeric1.rul

	которых содержатся римские цифры.					
geoplace.rul	Правила для выделения географических наименований, в составе которых есть ключевое слово.	FullGeoName RuleKeyFirst1, FullGeoName RuleKeyFirst2, FullGeoName RuleKeyFirst3, FullGeoName RuleKeyFirst Long, FullGeoName RuleKeyFirst Long2, FullGeoName RuleKeyFirst Delim, FullGeoName RuleKeyFirst Delim2, FullGeoName RuleLastName1, FullGeoName RuleLastName2	Ключевое слово в начале.	Geoplace: KeyFirst	г. Москва, город Новый Шарой, город Арсеньев, Республике Саха, реки Фонтанки	Опирается на фильтры. Различать, в начале стоит ключевое слово или в конце, требуется для корректной работы синтанизатора
		FullGeoName RuleKeyLast, FullGeoName RuleWithAdj, RepOfSSSR	Ключевое в конце.	Geoplace: KeyLast	Киргизской ССР, Ханты Мансийский АО, Ненецкий авт. окр.	
geoplace_for_address.rul	Сворачивает географические наименования с полным ключевым	FullGeoName RuleForAddress1, FullGeoName RuleForAddress2		Geoplace: KeyLast	МОСКОВСКИЙ ОКРУГ, Свердловской области, Щербиновский район, Краснодарского края	Правила стоит подключать, только если требуется

	словом, стоящим после названия. Требуется для выделения адресов.	, FullSubjName RuleForAddress1 , FullSubjName RuleForAddress2				выделять адреса, т.к. они мешают работе синтанизатора. Опираются на фильтры и geoplace.rul
person_name.rul	Правила выделения имен персон в особых конструкциях.	PersonName_Rule1, PersonName_Rule2		Person:Name	фамилия: Иванов имя: Иван отчество: Иванович	
idioms3.rul	Содержит правило для выделения составных пространственных предлогов.	НА_УЛИЦАХ_ GEO_Rule		Preposition:Place	на юге (чего-либо), в центре (чего-либо), на улицах (чего-либо)	Опирается на фильтр
introductory.rul	Содержит правила для выделения вводных оборотов.	IntroPreRule1, IntroPreRule1_1, IntroPreRule2, IntroPreRule2_1, IntroPreRule3, IntroPreRule5 IntroPostRule1, IntroPostRule2, IntroWholeSentRule1, IntroWholeSentRule2, IntroWholeSentRule3, IntroSourceRule1, IntroSourceRule1_1,	Вводные обороты, указывающие на источник информации.	Introductory:Source	Как сообщал во вторник Минтранс,... Как рассказал агентству неназванный сотрудник спецназа, Как сообщили в Главном управлении по делам ГО и ЧС ЯНАО,... Пресс-секретарь столичного департамента ЖКХ Марина Орлова сообщила «Коммерсанту», По данным, полученным корреспондентом ИА REGNUM в пресс-службе компании МАКС, сообщает пресс-служба столичного метрополитена. Об этом	

		IntroSource Rule1_2, IntroSource Rule2			сообщает телеканал «Россия».	
		IntroAssurance Rule, IntroAssurance Rule, IntroConfirmation Rule2, IntroReminding Rule, IntroRealEvents Rule, IntroAssumption Rule, IntroMisbelief Rule, IntroSubjectivity Rule1, IntroSubjectivity Rule2, IntroSubjectivity Rule3, IntroSubjectivity Rule4, IntroReasoning Rule1	Вводные обороты различной семантики.	Introductory: Introductory	не сомневайтесь, как по- вашему, как вы думаете, насколько мы можем судить, видите ли в чем дело,	
address1.rul			Названия улиц.	Special: AddressPart_Street	Кутузовский проспект, ул. Лесная	

	Содержит правила для выделения элементов адреса.	Street_Rule, Street_Rule1, Street_Rule2 House_Rule Apartment_Rule Doorway_Rule Floor_Rule	Номера домов.	Special: AddressPart_ House	дом 3; д. 3, корп. 5	Опирается на фильтры, numeric*.rul
			Номера квартир, офисов.	Special: AddressPart_ Apartment	кв. 7; 21 офис; комн.1	
			Подъезд.	Special: AddressPart_ Doorway	1 подъезд; подъезд 1	
			Этаж.	Special: AddressPart_ Floor	1 этаж; эт. 11-й	
address2.rul	Собирает адрес целиком из цепочки элементов.	FullAddress_Rule, PartAddress_Rule, ReverseAddress_Rule, Route_Rule, HouseStreet_Rule		Special:Address	г. Томск, ул. Лусная, д.7; ПОСЕЛОК ЛОБВА НОВОЛЯЛИНСКОГО РАЙОНА СВЕРДЛОВСКОЙ ОБЛАСТИ, УЛ. ЗАВОДСКАЯ, 4; 656015, БАРНАУЛ, ПР. ЛЕНИНА, Д. 76; ЛЕНИНА ПРОСПЕКТ, Д. 32 / 27, ЕКАТЕРИНБУРГ, 620075	Опирается на geoplace.rul и address1.rul
address_ino.rul	Выделяет иностранные адреса, опираясь на ключевое слово «адрес:».	ContextBased Latinic_Rule, ContextBased Translit_Rule		Special:Address	адрес офиса: 16 Iton Muse South, London, SW1W 9HP адрес: Итон Мьюз Саут 16, Лондон	
passport.rul	Выделяет данные общегражданских паспортов.	PassportNumber_Rule		Special:Passport	паспорт: 0203 №123456, выдан ОВД г. Томска 13.12.2000г. паспорт серия 42 03 номер 123456, выдан 31.12.2001 года ОВД г. Москвы	Опирается на date*.rul, geoplace.rul

zagan.rul	Данные заграничного паспорта.	Zagan_Rule1		Special: ExternalPassport	заграничный паспорт номер 01 1234567 загранпаспорт 01 №123456	
person_position1.rul	Содержит правила для выделения должностей.	PersonPosition_Rule1, PersonPosition_Rule3, PersonPosition_Rule4, PersonPosition_Rule5, PersonPosition_Rule6, CandidateRule1, FatherRule1, SenatorRule		Person:Position	председатель правления, Стажер по должности оперуполномоченного уголовного розыска, полпред президента в Северо-Кавказском федеральном округе, министр РФ по развитию Дальнего Востока, кандидат в президенты Франции от Социалистической партии, сенатор от Башкирии	
person_position2.rul	Содержит правила для выделения должностей.	PersonPosition_Rule2, PersonPosition_Rule2_1, PersonPosition_Rule2_2		Person:Position	заместитель мэра, и.о. директора, помощник прокурора, заместителя главного бухгалтера	Должно следовать после person_position1.rul
metro.rul	Содержит правила для выделения названий станций Московского метрополитена.	MetroStation Name_Rule, MetroStation Name_Rule1		Geoplace:Metro	метро «Арбатская», м. «Арбатская», станции «Битцевский парк», станции метро «Битцевский парк», метро «Охотный ряд»	Опирается на фильтр
last.rul	Содержит дополнительные правила для выделения процентов,	PersentFreeIn Sentence_Rule	Обрабатывает проценты и единицы измерения, которые не входят в состав другой именной группы.	Numeric:Persent	5,4%, не более 15%, около 80 метров, на 95 метров	Опирается на numeric*.rul

	должностей. Должно подключаться последним.	Position_Rule1	Составные должности вида: помощник/ заместитель/ ио + название должности.	Person:Position		Правило избыточно, дублирует person_position2.rul
tokenization.rul	Содержит правила, улучшающие токенизацию.	MergeNumericHyphenWord	Распознавание прилагательных и существительных, в состав которых входят числа.	не присваивают семантический тип.	10-метровый, 100-литровый, 5-летие, 2-й	Должно подключаться вначале, перед другими правилами, опирается на фильтр.
cars.rul	Содержит правила для выделения марок машин	Cars_Rule1, Cars_Rule2, Cars_Rule2_1,		Artifact:Cars	Машина Лада, Автомобиль «Богдан», автомобиле ВАЗ-2115, автомобиле «Ваз-2115», а / м КАМАЗ	Опирается на фильтр.
car_to_artifact.rul	Заменяет семантический тип Artifact:Cars на Artifact:Name. Используется в случае, если не требуется отделять автомобили от прочих артефактов.	CarToArtifact_Rule		Artifact:Name		Должно следовать после cars.rul.
number.rul	Выделяет любые номера, кроме тех что распознаются прочими правилами.	NumberRule1, NumberRule2		Special:Number	N1, N УГЗ-11/539, № 212/11309, №Р-92/12, #619, #31300420650-03	Должно стоять в конце, чтобы не конфликтовать с другими правилами
_special.rul	Содержит правила, которые сворачивают	Complex Geoplace1,		Geoplace:Name		Опирается на geoplace.rul.

	цепочку из географических наименований в единую сущность. Правило отключено во всех конфигурациях за ненадобностью. Установить, с какой целью оно было создано, не представляется возможным.	Complex Geoplace2				
_vehicle_identification_number.rul	Содержит правило для выделения ВИН автомобилей.	VehicleId Number_Rule		Special:VIN	ВИН: 1M8GDM9AXKP042788; 4T1BE32K42U512832	
airbase.rul	Тестовое правило, которое попало в конфигурацию случайно и должно быть удалено.					
codex_assemble1.rul	Узкоспециализированное правило, которое сворачивает номер статьи уголовного кодекса и название статьи в скобках в одну сущность.	CodexExpanded_Rule		Special:Codex	ст.ст. 77 (бандитизм), 79 (массовые беспорядки), 102 (убийство), 126-1 (захват заложников), 213-3 (терроризм) УК РСФСР; ст.ст.205 (терроризм), 222 (незаконное приобретение, хранение оружия, взрывчатых веществ) УК РФ; ст. 279 (вооруженный мятеж) УК РФ.	Опирается на codex_*.rul

license_number.rul	Содержит правила для выделения номеров водительских удостоверений.	LicenseNumber_Rule		Special: LicenseNumber	водительское удостоверение: №АА-111-222 (категория А); номер водительского удостоверения ВВ-12-123 (АА).	
name_eraser.rul	Содержит правила, препятствующие ошибочному распознаванию персон, не позволяющие выделять имена персон из различных названий. Требуется для повышения точности семантико-синтаксического анализа.	FalsePerson Eraser_Rule, FalsePerson Eraser_Rule2, FalsePerson Eraser_Rule3, Slava_Rule1, Slava_Rule2		не присваивает семантический тип	имени Станиславского, им. М.В. Ломоносова, Слава России, Орден Славы	
name_in_artifact_eraser.rul	Удаляет имена персон из названий артефактов, чтобы избежать ошибочного выделения персон	PersonName EraserRule1, PersonName EraserRule2		не присваивает семантический тип	постановку «Скупого рыцаря», «Смоленский», журнал «Эксперт-Сибирь», «Крузенштерн»	
place_of_birth.rul	Содержит правило для выделения места рождения.	PlaceOfBirth_Rule1		Special: PlaceOfBirth		место рождения: Ростов-на-Дону; место рождения: Казахстан
url.rul	Содержит правила для выделения веб-адресов.	URL_Rule1, URL_Rule2		Special:URL	http://somewhere.com; www.cnn.com; http://www.site.gov; http://www.un.org/sdf/dfgh/fgiy/erty.htm7;	

					http://voyager.ocs.mq.edu.au/ cgi-bin/Pwebrecon.cgi?DB= local&PAGE=First	
--	--	--	--	--	---	--