SMEV

Оглавление

[Назначение программного средства 1](#_Toc367950274)

[Командная строка 1](#_Toc367950275)

[Запуск утилиты в диалоговом режиме 1](#_Toc367950276)

[Запуск утилиты в режиме командной строки 2](#_Toc367950277)

[Ключи конфигурационного файла 2](#_Toc367950278)

[Ключи общего назначения 2](#_Toc367950279)

[Ключи, определяемые используемым сервисом 3](#_Toc367950280)

[Взаимодействие с сервисами 3](#_Toc367950281)

[Сервис rkn.gov.ru 3](#_Toc367950282)

[Конфигурационный файл 3](#_Toc367950283)

[Схема обмена данными с сервисом 4](#_Toc367950284)

[Работа в диалоговом режиме 5](#_Toc367950285)

[Работа в режиме командной строки 5](#_Toc367950286)

# Назначение программного средства

Утилита SMEV предназначена для обеспечения во взаимодействии с сервисами облака СМЭВ (Системы Межведомственного Электронного Взаимодействия), в том числе:

* Федеральной службы по надзору в сфере связи, информационных технологий и массовых коммуникаций

# Командная строка

Запуск утилиты производится командной строкой в одном из следующих возможных вариантов:

* Запуск утилиты в диалоговом режиме
* Запуск утилиты в режиме командной строки

## Запуск утилиты в диалоговом режиме

Возможные варианты командной строки:

smev /config:<путь к файлу конфигурации>

По данной команде после запуска утилиты открывается диалоговое окно, которое обеспечивает выполнение необходимых операций по обмену информацией с сервисом в диалоговом режиме.

Например:

smev /config:C:\smev\unittell.cfg

## Запуск утилиты в режиме командной строки

Возможные варианты командной строки:

smev **/silent** /config:<путь к файлу конфигурации>

По данной команде все необходимые операции по обмену с выбранным сервисом производятся без участия пользователя, а итог обмена отображается кодом возврата приложения.

Например:

smev /silent /config:C:\smev\unittell.cfg

# Ключи конфигурационного файла

Конфигурационный файл представляет собой текстовой файл, каждая строка которого имеет следующий формат:

<Имя ключа>=<Значение ключа>

Например:

Service\_name=rkn.gov.ru

Исключение составляет ключ **Parameter**, который имеет следующий формат

Parameter:<Имя параметра>=<Значение параметра>

Например:

Parameter:Operator\_Name=ООО "ЮнитТел"

Порядок следования ключей в конфигурационном файле не имеет значения.

При считывании конфигурационного файла игнорируются пустые строки, а также строки, начинающиеся с символа «точка с запятой».

## Ключи общего назначения

$ROOT$ - Значение макро-подстановки $ROOT$ в значениях остальных ключей (необязательный ключ)

Service\_name - Идентификатор сервиса (обязательный ключ)

Service\_url - URL сервиса (обязательный ключ)

Crypto\_sign\_in - Путь файла для подписывания (обязательный ключ)

Crypto\_sign\_out - Путь файла подписи (обязательный ключ)

Crypto\_sign\_exec - Командная строка запуска утилиты формирования подписи (обязательный ключ)

Log\_path - Путь к файлу лога технологического лога (необязательный ключ)

Список допустимых значений для ключа **Service\_name** ограничен следующим перечнем:

* rkn.gov.ru - Федеральная служба по надзору в сфере связи, информационных технологий и массовых коммуникаций

## Ключи, определяемые используемым сервисом

Cookie\_path - Путь к файлу сохранения состояния

Result\_path - Путь к файлу результата обмена

Parameter - Задание параметра обмена с сервисом

# Взаимодействие с сервисами

## Сервис rkn.gov.ru

Данный сервис используется для получения реестра подлежащих блокировке Интернет-ресурсов, формируемого Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций.

### Конфигурационный файл

Кроме ключей общего назначения обмен с сервисом управляется следующими ключами и параметрами:

Cookie\_path - Путь к файлу сохранения состояния (необязательный ключ)

Result\_path - Путь к файлу результата обмена (обязательный ключ)

Parameter: Operator\_Name - Наименование оператора (обязательный ключ)

Parameter: Inn - ИНН оператора (обязательный ключ)

Parameter: Ogrn - ОГРН оператора (обязательный ключ)

Parameter: Email - Адрес электронной почты оператора (обязательный ключ)

Parameter: Timestamp - Временная метка запроса (необязательный ключ). Данный параметр может задаваться, если необходимо использовать готовый файл подписи вместо ее формирования в режиме реального времени.

**Внимание!** В том случае, если с одного компьютера производятся запросы с разными параметрами конфигурации, то и значения ключей **Cookie\_path** и **Result\_path** в данных конфигурациях должны различаться во избежание взаимовлияния исполнения запросов.

Например:

$ROOT$=D:\\_Projects\_SVN\SMEV

Service\_name=rkn.gov.ru

Service\_url=vigruzki.rkn.gov.ru:80/services/OperatorRequest/

Crypto\_sign\_in=$ROOT$\crypto\_data.txt

Crypto\_sign\_out=$ROOT$\crypto\_sign.txt

Crypto\_sign\_exec=...

Cookie\_path=$ROOT$\unittel.cookie

Result\_path=$ROOT$\rkn.zip

Log\_path=$ROOT$\smev.log

Parameter:Operator\_Name=ООО "ЮнитТел"

Parameter:Inn=7722553712

Parameter:Ogrn=1057747889605

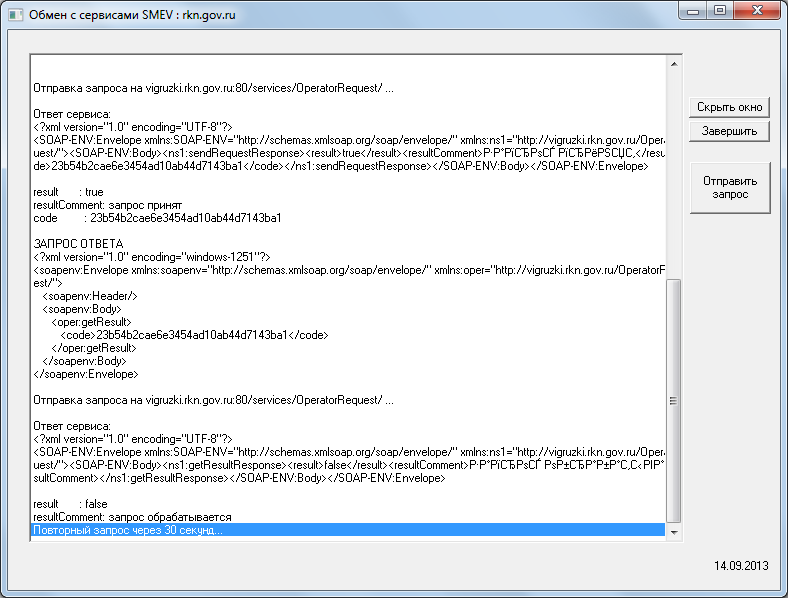
Parameter:Email=vt@jettel.ru

### Схема обмена данными с сервисом

* Для получения контрольной временной метки Реестра на сервис отправляется запрос **getLastDumpDateEx**.Если при обработке запроса сервисом была выдана ошибка, то дальнейший обмен прекращается. Если в файле конфигурации задан ключ **Cookie\_path** и по указонному им пути существует файл, то производится сравнение данных файла с переданным сервисом значением **контрольной временной метки**. Если данные совпадают, то дальнейший обмен прекращается. В случае, если данные не совпадают либо Cookie-файл не указан или не существует - производится переход к следующему шагу обмена.
* На основе заданных конфигурационных параметров производится формирование файла запроса Реестра. В том случае, если параметр **Timestamp** задан – при формировании временной метки файла используется его значение, в противном случае временная метка файла формируется автоматически в виде символьной строки по шаблону yyyy-mm-ddThh:mi:ss.000+04:00, где yyyy, mm, dd – год, месяц и день текущей даты, а hh, mi, ss – часы, минуты и секунды текущего времени.
* Производится формирование подписи для файла запроса Реестра. При этом подлежащие подписыванию данные помещаются в файл, задаваемый ключом **Crypto\_sign\_in**, после чего производится исполнение командной строки, задаваемой ключом **Crypto\_sign\_exec**. Данные подписи считываются из файла, задаваемого ключом **Crypto\_sign\_out**.
* Данные файла запроса Реестра и его подпись передаются на сервис с использованием запроса **sendRequest**. Если при обработке запроса сервисом была выдана ошибка, то дальнейший обмен прекращается. В случае успешного исполнения запроса сервис возвращает **идентификатор выгрузки**.
* Для получения данных Реестра на сервис передается запрос **getResult** с указанием в нем ранее полученного **идентификатора выгрузки**. Если сервис возвращает статус «Запрос обрабатывается», то через 30 секунд запрос **getResult** повторяется. Если сервис возвращает сообщение об ошибке, то дальнейший обмен прекращается. Если сервис возвращает статус «Запрос успешно обработан», то производится прием данныч Реестра, которые записываются в файл, задаваемый ключом **Result\_path**.
* Если в файле конфигурации задан ключ **Cookie\_path**, то по указонному им пути создается файл, в который записывается значение **контрольной временной метки**, полученной ранее с помощью запроса **getLastDumpDate**.
* Обмен завершен.

### Работа в диалоговом режиме

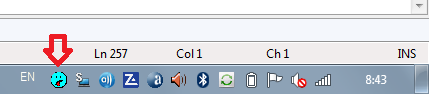
После успешного запуска приложения открывается диалоговое окно:



Для запуска процесса обмена данными с сервисом необходимо нажать кнопку «Отправить запрос». При этом в текстовом окне будет отображаться протокол процесса обмена.

Кнопка «Завершить» служит для завершения работы приложения.

При нажатии кнопки «Скрыть» диалоговое окно приложения закрывается и может быть снова открыто двойным кликом по иконке приложения в трее:



Если приложение было скрыто в процессе обмена, то обмен продолжается обычным порядком.

### Работа в режиме командной строки

В данном режиме приложение работает без какой-либо визуальной индикации. Результат работы приложения сообщаются через код возврата (ERRORLEVEL), при этом детали фиксируются в файле лога, если он задан ключом конфигурационного файла **Log\_path**.

В случае ошибки при запуске приложения используются следующие коды возврата:

1 - ошибка инициализации

2 - ошибка создания диалоговых окон

3 - ошибка считывания файла конфигурации

4 - ошибка считывания файла ключей командной строки

5 - неизвестный сервис СМЭВ

Результат обмена данными с сервисом индицируется следующими кодами возврата:

0 - обмен успешно завершен, данные получены

101 - ошибка при обмене с сервисом, детали – в файле лога

102 - обмен успешно завершен, данные не требуют обновления