

*КРАТКОЕ РУКОВОДСТВО ПО УСТАНОВКЕ И НАСТРОЙКЕ*

1С:Сервер взаимодействия

январь 2020

Пустая страница

## ОГЛАВЛЕНИЕ

Глава 1. Общая информация .....	1
Глава 2. Требования к аппаратуре и программному обеспечению .....	1
Глава 3. Установка сервера взаимодействия .....	1
3.1. Общая информация .....	1
3.2. Установка .....	2
3.2.1. Утилита ring .....	2
3.2.2. Сервер взаимодействия .....	2
Глава 4. Настройка .....	3
4.1. Общая информация .....	3
4.2. Настройка пакетов .....	3
4.2.1. Для ОС Windows .....	3
4.2.2. Для ОС Linux .....	3
4.3. Настройка базы данных .....	4
4.4. Настройка WebSocket .....	4
4.5. Запуск необходимых сервисов .....	5
4.6. Проверка работоспособности .....	5
4.7. Завершение настройки .....	5
Инициализация базы данных сервера .....	6
Подключение к ICE-серверу .....	6
Глава 5. Подключение к установленному серверу .....	6
Приложение 1. Руководство администратора – Управление системой взаимодействия .....	6
6.16.10. Управление системой взаимодействия .....	6
Приложение 2. Руководства разработчика - Система взаимодействия .....	10
30.1. Общая информация .....	10
30.2. Устройство системы взаимодействия .....	12
30.3. Интерактивное взаимодействие .....	13
30.3.1. Параметры пользователей .....	13
30.3.2. Виды обсуждений .....	13
30.3.3. Виды сообщений .....	14
30.3.4. Оповещения .....	15
30.4. Не интерактивное взаимодействие .....	16
30.4.1. Общая информация .....	16
30.4.2. Регистрация/отмена регистрации приложения .....	16
30.4.3. Работа с пользователями .....	17
30.4.4. Работа с обсуждениями .....	20
30.4.5. Совместное использование .....	27
30.5. Администрирование .....	28
Приложение 3. Руководства администратора - Описание и расположение служебных файлов .....	28
3.19. inetcfg.xml .....	28
Атрибут ntlm необязательный .....	29
Атрибут protocols необязательный .....	29

Атрибут user необязательный .....	29
Атрибут password необязательный .....	29
Атрибут bypassOnLocal необязательный .....	29
Атрибут bypassOnAddresses необязательный .....	29
Приложение 4. Руководства разработчика - Описание прав доступа .....	30

## ГЛАВА 1. ОБЩАЯ ИНФОРМАЦИЯ

Данный документ содержит краткое руководство по установке и настройке сервера взаимодействия. После установки и настройки сервер можно использовать для работы системы взаимодействия в «1С:Предприятии». Подробнее про систему взаимодействия можно прочитать в книге «1С:Предприятие 8.3. Руководство разработчика» ([Глава 30](#)).

## ГЛАВА 2. ТРЕБОВАНИЯ К АППАРАТУРЕ И ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ

Поставка сервера системы взаимодействия состоит из следующих компонентов:

- собственно сервер взаимодействия для работы с системой «1С:Предприятие».
- сервер Hazelcast;
- сервер Elasticsearch;
- утилита ring;
- внешняя обработка [CollaborationSystemRegister.epf](#), предназначенная для регистрации информационной базы на сервере взаимодействия.

**ВНИМАНИЕ!** Для работы рекомендуется использовать сервера Hazelcast и Elasticsearch, которые поставляются в комплекте поставки сервера взаимодействия.

Требования к компьютеру:

- операционная система: 64-разрядные версии операционных систем CentOS 7.2 и старше, Windows 7 и старше;
- процессор Intel/AMD с тактовой частотой 1800 МГц и выше;
- оперативная память 4 GB и более;
- 30 GB свободного места на дисковом накопителе (не учитывая место для размещения базы данных);

К установленному серверу взаимодействия может подключаться система «1С:Предприятие» версий [8.3.10](#) и [8.3.11](#). При работе с версией 8.3.10 будут недоступны видеозвонки.

Для работы системы взаимодействия на компьютере необходимо наличие следующих программных продуктов:

- Oracle Java 8 (64-разрядная) – версия 1.8.

Скачать Java: <https://www.java.com/ru/download/>.

- СУБД PostgreSQL – версия 9.5 и старше.

Скачать сервер: <https://www.postgresql.org/download/>.

Документация: <https://postgrespro.ru/docs/postgresql/>.

- Утилита ring (64-разрядная) – версия 0.8 и старше.

Утилита версии 0.8 поставляется:

- в составе системы «1С:Предприятие» версии 8.3.11 и старше;
- в составе сервера взаимодействия.

Документация к утилите ring расположена в книге «1С:Предприятие 8.3. Руководство администратора» (<https://its.1c.ru/db/v83doc#bookmark:adm:TI000000674>).

## ГЛАВА 3. УСТАНОВКА СЕРВЕРА ВЗАИМОДЕЙСТВИЯ

### 3.1. ОБЩАЯ ИНФОРМАЦИЯ

Данная глава описывает развертывание и настройку компонентов сервера взаимодействия. Существенным предположением является то, что все компоненты установлены на одном компьютере. Развертывание и настройка кластерного варианта сервера взаимодействия в данной главе не приводится.

Файл пакета компоненты имеет имя вида `1ce-cs2-<имя пакета>-a.b.c-d.x64`, где:

- [<имя пакета>](#) – имя компонента;
- [server](#) – пакет собственно сервер взаимодействия;
- [elasticsearch](#) – пакет сервера Elasticsearch;
- [hazelcast](#) – пакет сервера Hazelcast.
- [a.b.c-d](#) – это полный номер версии компонента, расположенной в пакете, который соответствует номеру в виде [a.b.c.d](#).

Для ОС Windows пакет имеет расширение [.msi](#), для ОС Linux – [.rpm/.deb](#), в зависимости от используемой ОС.

Для ОС Windows поставляется пакет [1ce-cs2-all-in-one-setup-2.0.14-1.x64.exe](#), который представляет собой программу установки, которая содержит в себе все необходимые компоненты.

### 3.2. УСТАНОВКА

#### 3.2.1. УТИЛИТА RING

В том случае, если на компьютере, где планируется развернуть сервер взаимодействия, уже установлена утилита ring, то следует проверить версию установленной утилиты с помощью команды `ring --version`.

В том случае, если версия утилиты младше требуемой (0.8), то следует удалить установленную версию утилиты и установить ее из поставки «1С:Предприятия» версии 8.3.11 и старше или из состава дистрибутива сервера взаимодействия.

Описание установка утилиты ring описано в книге «1С:Предприятие 8.3. Руководство администратора» (<https://its.1c.ru/db/v83doc#bookmark:adm:TI000000678>).

#### 3.2.2. СЕРВЕР ВЗАИМОДЕЙСТВИЯ

##### ПАКЕТНАЯ УСТАНОВКА ДЛЯ ОС WINDOWS

В том случае, если сервер взаимодействия разворачивается на ОС Windows, имеется возможность выполнить всю установку в автоматическом режиме с помощью специального мастера установки.

Для этого необходимо запустить [1ce-cs2-all-in-one-setup-2.0.14-1.x64.exe](#) и следовать указаниям мастера. Во время установки потребуются указать имя и пароль пользователя, от имени которого сервер взаимодействия будет подключаться к СУБД PostgreSQL. Остальные действия по установке и настройке мастер выполнит автоматически.

При этом установка будет выполнена со следующими параметрами:

- Имя экземпляра сервиса взаимодействия: **CS**.
- Имя экземпляра Hazelcast: **hazelcast**.
- Имя экземпляра Elasticsearch: **elasticsearch**.
- Для СУБД PostgreSQL:
- Создан пользователь **CS** с паролем **cs-pass**.
- Создана база данных **1ce-cs**, владельцем которой является пользователь **CS**.
- К базе данных **1ce-cs** подключено расширение **uuid-oss**.
- Настроен WebSocket.
- Выполнена инициализация базы данных сервиса.
- Выполнено подключение к ICE-серверу **dialog-turn1-gpt-msk.1c.ru:3478**.

Данные созданных экземпляров сервисов будут располагаться в каталоге `%ProgramData%\1C\1CE\instances\<имя экземпляра>`.

После завершения установки будут запущены все необходимые сервисы.

После выполнения установки сервера взаимодействия с помощью мастера, выполнять настройки, описанные в соответствующем разделе (далее) нет необходимости.

##### РУЧНАЯ УСТАНОВКА

Установка сервера системы взаимодействия осуществляется последовательной установкой всех пакетов, входящих в состав поставки.

Установка каждого компонента выполняется с помощью ручного запуска установки соответствующего пакета. Синтаксис команды запуска установки определяется используемой операционной системой и пакетным менеджером (при наличии).

Перед запуском установки пакетов под управлением ОС Windows необходимо в свойства файла **.msi** указать, что он должен запускаться в режиме совместимости с предыдущей версией Windows.

После установки всех пакетов, входящих в комплект поставки, следует перейти к настройке, описанной в следующих разделах.

## ГЛАВА 4. НАСТРОЙКА

## 4.1. ОБЩАЯ ИНФОРМАЦИЯ

Все команды настройки должны выполняться от имени пользователя, обладающего административными правами. Для ОС Windows это означает необходимость запуска консоли, в которой выполняются команды настройки, от имени администратора. Для ОС Linux команды настройки должны выполняться от имени пользователя **root** (или аналогичного).

## 4.2. НАСТРОЙКА ПАКЕТОВ

## 4.2.1. ДЛЯ ОС WINDOWS

## 4.2.1.1. СЕРВЕР ВЗАИМОДЕЙСТВИЯ

Для начальной инициализации сервера взаимодействия необходимо выполнить следующие действия:

```
ring cs instance create --dir %ProgramData%\1C\1CE\instances\<cs_instance>
ring cs --instance <cs_instance> service create --java-home "%JAVA_HOME%" --stopped
```

В данном примере:

- **<cs\_instance>** – название экземпляра сервера взаимодействия.

## 4.2.1.2. HAZELCAST

Для начальной инициализации сервера Hazelcast необходимо выполнить следующие действия:

```
ring hazelcast instance create --dir %ProgramData%\1C\1CE\instances\<hc_instance>
ring hazelcast --instance <hc_instance> service create --java-home "%JAVA_HOME%" --stopped
```

В данном примере:

- **<hc\_instance>** – название экземпляра Hazelcast.

## 4.2.1.3. ELASTISEARCH

Для начальной инициализации сервера Elasticsearch необходимо выполнить следующие действия:

```
ring elasticsearch instance create --dir %ProgramData%\1C\1CE\instances\<elasticsearch
instance>
ring elasticsearch --instance <elasticsearch instance> service create --java-home
"%JAVA_HOME%" --stopped
```

В данном примере:

- где **<elasticsearch instance>** – название экземпляра Elasticsearch.

## 4.2.2. ДЛЯ ОС LINUX

## 4.2.2.1. СЕРВЕР ВЗАИМОДЕЙСТВИЯ

Для начальной инициализации сервера взаимодействия необходимо выполнить следующие действия:

```
sudo useradd <cs_user>
sudo mkdir -p /var/cs/<cs_instance>
sudo chown <cs_user>:<cs_user> /var/cs/<cs_instance>
ring cs instance create --dir /var/cs/<cs_instance> --owner <cs_user>
ring cs --instance <cs_instance> service create --username <cs_user> --java-home
$JAVA_HOME --stopped
```

В данном примере:

- **<cs\_instance>** – название экземпляра сервера взаимодействия.
- **<cs\_user>** – имя пользователя, от имени которого будет функционировать сервер взаимодействия.

## 4.2.2.2. HAZELCAST

Для начальной инициализации сервера Hazelcast необходимо выполнить следующие действия:

```
sudo useradd <hc_user>
sudo mkdir -p /var/cs/<hc_instance>
sudo chown <hc_user>:<hc_user> /var/cs/<hc_instance>
```

## 1С:СЕРВЕР ВЗАИМОДЕЙСТВИЯ: Краткое руководство по установке и настройке

```
ring hazelcast instance create --dir /var/cs/<hc_instance> --owner <hc_user>
ring hazelcast --instance <hc_instance> service create --username <hc_user> --java-home
$JAVA_HOME --stopped
```

В данном примере:

- **<hc\_instance>** – название экземпляра Hazelcast.
- **<hc\_user>** – имя пользователя, от имени которого будет функционировать сервер Hazelcast.

### 4.2.2.3. ELASTISEARCH

Для начальной инициализации сервера Elasticsearch необходимо выполнить следующие действия:

```
sudo useradd <elastic_user>
sudo mkdir -p /var/cs/<elastic_instance>
sudo chown <elastic_user>:<elastic_user> /var/cs/<elastic_instance>
ring elasticsearch instance create --dir /var/cs/<elastic_instance> --owner <elas-
tic_user>
ring elasticsearch --instance <elastic_instance> service create --username <elas-
tic_user> --java-home $JAVA_HOME --stopped
```

В данном примере:

- **<elastic\_instance>** – название экземпляра Elasticsearch.
- **<elastic\_user>** – имя пользователя, от имени которого будет функционировать сервер Elasticsearch.

## 4.3. НАСТРОЙКА БАЗЫ ДАННЫХ

К моменту выполнения действий данного раздела PostgreSQL должен быть установлен на компьютере, где выполняется развертывание и настройка сервера взаимодействия.

Также, перед началом настройки базы данных для работы с сервером взаимодействия, должны быть выполнены следующие настройки:

- Создан пользователь, от имени которого сервер взаимодействия будет взаимодействовать с СУБД. Данный пользователь должен обладать правами создания базы данных. В дальнейшем для определения имени этого пользователя будет использоваться **db\_user**. Пароль данного пользователя будет указываться как **db\_user\_pwd**.

Создание пользователя: <https://www.postgresql.org/docs/current/static/sql-createuser.html>.

- Создана база данных, в которой будут храниться данные сервера взаимодействия. В качестве владельца этой базы должен выступать пользователь **db\_user**. В дальнейшем для определения базы данных будет использоваться **cs\_db**.

Создание базы данных: <https://www.postgresql.org/docs/current/static/manage-ag-createdb.html>.

- Сервер должен поддерживать аутентификацию по паролю.

Методы аутентификации: <https://www.postgresql.org/docs/current/static/auth-methods.html>.

Для базы данных **cs\_db** следует подключить расширения **uuid-oss** (подробнее <https://www.postgresql.org/docs/current/static/uuid-oss.html>). Для этого необходимо выполнить следующую команду базы данных:

```
CREATE EXTENSION IF NOT EXISTS "uuid-oss";
```

Необходимо выполнить настройку параметров JDBC-драйверов PostgreSQL:

```
ring cs --instance <cs_instance> jdbc set-params --url jdbc:postgresql://lo-
calhost:5432/cs_db?currentSchema=public
ring cs --instance <cs_instance> jdbc set-params --username db_user
ring cs --instance <cs_instance> jdbc set-params --password db_user_pwd
ring cs --instance <cs_instance> jdbc-privileged set-params --url jdbc:postgresql://lo-
calhost:5432/cs_db?currentSchema=public
ring cs --instance <cs_instance> jdbc-privileged set-params --username db_user
ring cs --instance <cs_instance> jdbc-privileged set-params --password db_user_pwd
```

## 4.4. НАСТРОЙКА WEBSOCKET

Для взаимодействия сервера взаимодействия и клиентского приложения используется протокол WebSocket. Для настройки работы этого протокола следует выполнить следующие действия:

```
ring cs --instance <cs_instance> websocket set-params --hostname <cs_host>
ring cs --instance <cs_instance> websocket set-params --port <cs_port>
```

В данном примере:



- **<cs\_host>** – имя или IP-адрес компьютера, на котором установлен сервер взаимодействия и к которому будет подключаться клиентское приложение системы взаимодействия (система «1С:Предприятие»).
- **<cs\_port>** – IP-порт, который будет использован для подключения к серверу взаимодействия клиентского приложения системы взаимодействия (системы «1С:Предприятие»).

В том случае, если при работе с сервером взаимодействия будет использоваться протокол **WS**, дальнейшие действия можно не выполнять. Хранилище ключей необходимо создавать только в том случае, когда предполагается использовать протокол **WSS**.

Для настройки защищенного канала необходимо создать хранилище ключей в формате JKS. Описание создания хранилища ключей <https://docs.oracle.com/cd/E19509-01/820-3503/ggfen/> (на английском языке). В созданное хранилище следует добавить сертификат для **<cs\_host>**, подписанный доверенным центром и сертификат этого доверенного центра.

После этого в командной строке следует выполнить следующие действия:

```
ring cs --instance <cs_instance> websocket set-params --wss true
ring cs --instance <cs_instance> websocket set-params --keystore-path <cs_keystore_path>
ring cs --instance <cs_instance> websocket set-params --keystore-password: <cs_keystore_pwd>
```

В данном примере:

- **<cs\_keystore\_path>** – полный путь к файлу хранилища сертификатов, к которому есть доступ от имени пользователя **<cs\_user>**.
- **<cs\_keystore\_pwd>** – пароль к файлу хранилища сертификатов.

Рекомендуется:

- Назвать файл хранилища сертификатов **websocket-keystore.jks**.
- Расположить этот файл:
  - на **OC Windows:** **%ProgramData%\1C\1CE\instances\<cs\_instance>\data\security\websocket-keystore.jks**.
  - на **OC Linux:** **/var/cs/<cs\_instance>/data/security/websocket-keystore.jks**.

### 4.5. ЗАПУСК НЕОБХОДИМЫХ СЕРВИСОВ

После окончания настройки необходимо запустить настроенные сервисы:

```
ring hazelcast --instance <hc_instance> service start
ring elasticsearch --instance <elastic_instance> service start
ring cs --instance <cs_instance> service start
```

После запуска серверу взаимодействия требуется некоторое время для инициализации базы данных и прочих настроек.

### 4.6. ПРОВЕРКА РАБОТОСПОСОБНОСТИ

Проверить работоспособность сервера взаимодействия можно с помощью обращения по адресу <http://localhost:8087/rs/health>.

В случае успешной инициализации сервер вернет ответ следующего вида:

```
{
  "status": "UP",
  "mainDbOk": true,
  "allShardsOk": true,
  "hazelcastOk": true,
  "elasticsearchOk": true,
  "mediaClusterOk": false,
  "mediaServers": {}
}
```

### 4.7. ЗАВЕРШЕНИЕ НАСТРОЙКИ

После успешной установки и настройки сервера взаимодействия необходимо выполнить ряд действий, завершающих настройку:

1. Инициализировать базу данных сервера.

## 1С:СЕРВЕР ВЗАИМОДЕЙСТВИЯ: Краткое руководство по установке и настройке

2. Выполнить подключение к ICE-серверу. Это действие необходимо выполнять только в том случае, если предполагается совершать видеозвонки, участники которых будут расположены за разными NAT-ами или будут использованы брандмауэры, которые настроены на запрет входящих соединений.

Для выполнения завершающих настроек необходимо использовать утилиту [curl](https://curl.haxx.se/download.html), которая доступна по адресу <https://curl.haxx.se/download.html>.

### ИНИЦИАЛИЗАЦИЯ БАЗЫ ДАННЫХ СЕРВЕРА

Команда инициализации выглядит следующим образом:

```
curl -Sf -X POST -H "Content-Type: application/json" -d "{ \"url\" : \"jdbc:postgresql://localhost:5432/<cs_db>\", \"username\" : \"<db_user>\", \"password\" : \"<db_user_pwd>\", \"enabled\" : true }" -u admin:admin http://localhost:8087/admin/bucket_server
```

В данном примере:

- **cs\_db** – имя базы данных сервера взаимодействия,
- **db\_user** – имя пользователя, от имени которого сервер взаимодействия подключается к СУБД.
- **db\_user\_pwd** – пароль этого пользователя.

### ПОДКЛЮЧЕНИЕ К ICE-СЕРВЕРУ

Команда подключения к ICE-сервера выглядит следующим образом:

```
curl -Sf -X POST -H "Content-Type: application/json" -d "{ \"url\" : \"turn:dialog-turn1-gpt-msk.1c.ru:3478\", \"username\" : \"video_module\", \"password\" : \"FJkvT1JfZY5RyQFqhrjQ\", \"enabled\" : true }" -u admin:admin http://localhost:8087/admin/ice_server
```

В данном примере выполняется подключение к ICE-серверу фирмы «1С».

## ГЛАВА 5. ПОДКЛЮЧЕНИЕ К УСТАНОВЛЕННОМУ СЕРВЕРУ

В том случае, если прикладное решение используется для своей работы сервер системы взаимодействия, развернутый самостоятельно (не сервер 1С:Диалог), то для регистрации информационной базы в сервисе следует использовать не стандартную функцию [Управление системой взаимодействия](#), а внешнюю обработку [CollaborationSystemRegister.epf](#), которая поставляется вместе с сервером.

Значение поля Адрес сервера взаимодействия состоит из трех частей:

- протокол: **ws:** или **wss:**;
- адрес сервера;
- порт сервера.

Все эти составные части можно получить в конфигурационном файле сервера взаимодействия [com.\\_1c.ecs.websocket.yml](#):

- Протокол определяется по значению параметра файла **wss**. Если этот параметр установлен в значение **true** – значит используется протокол **wss:**, в противном случае – протокол **ws:**.
- Адрес сервера определяется по значению параметра **hostname**.
- Порт сервера определяется по значению параметра **port**.

Конфигурационный файл сервера взаимодействия расположен:

- Для ОС Windows: **%ProgramData%\1C\1CE\instances\<cs\_instance>\config\**.
- Для ОС Linux: **/var/cs/<cs\_instance>/config/**.

## ПРИЛОЖЕНИЕ 1. РУКОВОДСТВО АДМИНИСТРАТОРА – УПРАВЛЕНИЕ СИСТЕМОЙ ВЗАИМОДЕЙСТВИЯ

### 6.16.10. УПРАВЛЕНИЕ СИСТЕМОЙ ВЗАИМОДЕЙСТВИЯ

С помощью данной стандартной функции предоставляется возможность выполнять административные действия по работе с системой взаимодействия:

1. Зарегистрировать текущую информационную базу в сервисе 1С:Диалог для того, чтобы стало возможно использовать систему взаимодействия.
2. Отменить регистрацию приложения в сервисе.

3. Выполнить/отменить объединение приложений системы взаимодействия.
4. Блокировать/разблокировать пользователей системы взаимодействия.

Для того чтобы пользователь мог выполнять регистрацию информационной базы, этот пользователь должен обладать правом **РЕГИСТРАЦИЯ ИНФОРМАЦИОННОЙ БАЗЫ СИСТЕМЫ ВЗАИМОДЕЙСТВИЯ**.

Для работы системы взаимодействия требуется, чтобы на компьютерах, с которых планируется использовать систему взаимодействия, был доступен URL <wss://1cdialog.com> с использованием IP-порта 443.

В том случае, если приложение еще не зарегистрировано в сервисе 1С:Диалог, система сразу предлагает выполнить регистрацию.

← → **Регистрация абонента сервиса и приложения**

Адрес электронной почты абонента:

Приложение будет связано с этим адресом.  
В дальнейшем на этот адрес будут приходить коды подтверждения

Наименование приложения:

Код регистрации:  **Получить код**

**Зарегистрировать**

**Рис. 77. Регистрация приложения**

В диалог вводится адрес электронной почты владельца абонента и представление регистрируемого приложения. После нажатия кнопки **Получить код** на указанный адрес электронной почты поступает письмо, в котором содержится код регистрации, выданный сервисом. Электронное письмо с кодом регистрации будет отправлено с адреса [info@1cdialog.com](mailto:info@1cdialog.com). Данный код следует ввести в поле **Код регистрации** и затем нажать кнопку **Зарегистрировать**.

← → **Регистрация абонента сервиса и приложения**

Адрес электронной почты абонента:

Приложение будет связано с этим адресом.  
В дальнейшем на этот адрес будут приходить коды подтверждения

Наименование приложения:

Код регистрации:  **Получить код**

**Зарегистрировать**

**Рис. 78. Ввод кода регистрации**

Успешное завершение регистрации сопровождается диалогом с информацией об этом.

После окончания регистрации и в том случае, если приложение уже зарегистрировано в системе взаимодействия, открывается форма, которая предлагает выполнить следующие действия: настроить совместное использование приложений, управлять блокировкой пользователей и отменить регистрацию данного приложения.

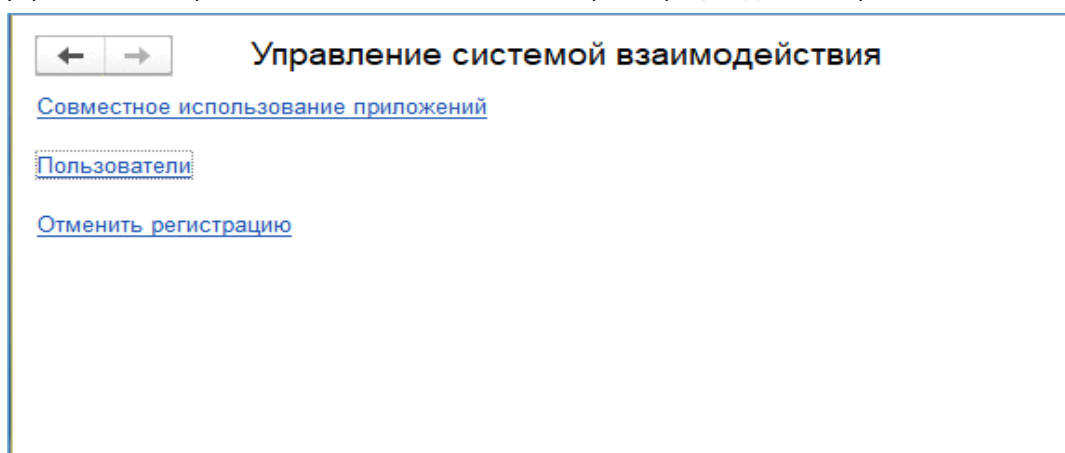


Рис. 79. Стартовая форма

Если пользователь нажимает гиперссылку [Совместное использование приложений](#), то открывается форма настройки совместного использования.

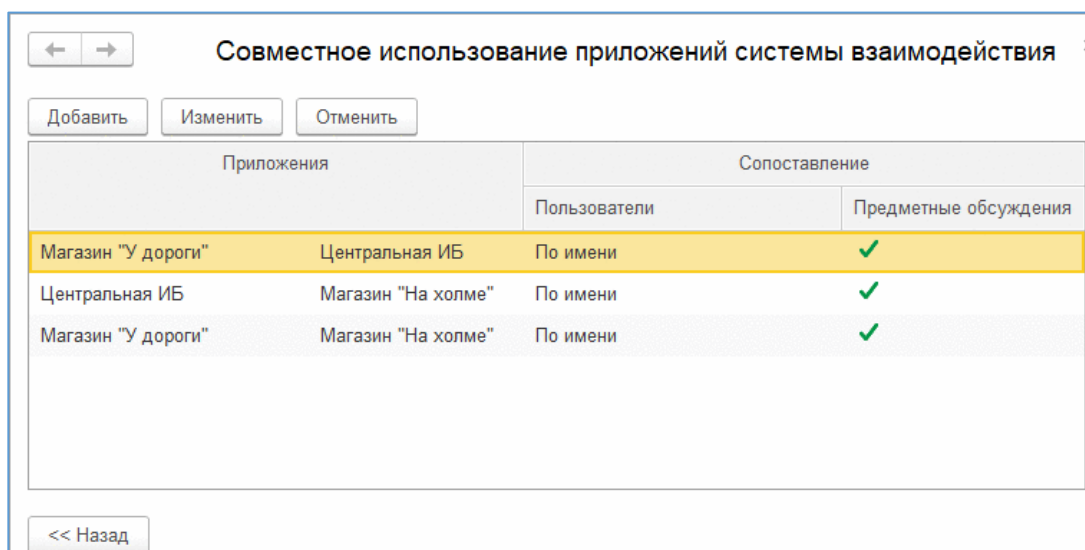
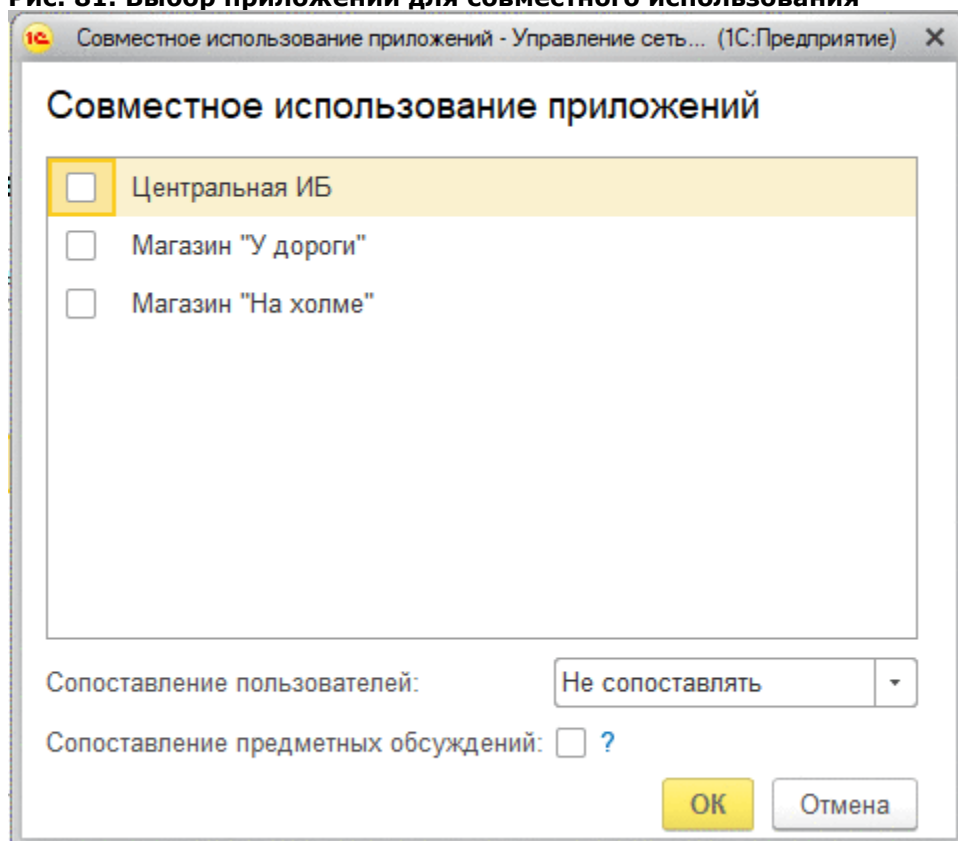


Рис. 80. Совместное использование приложений

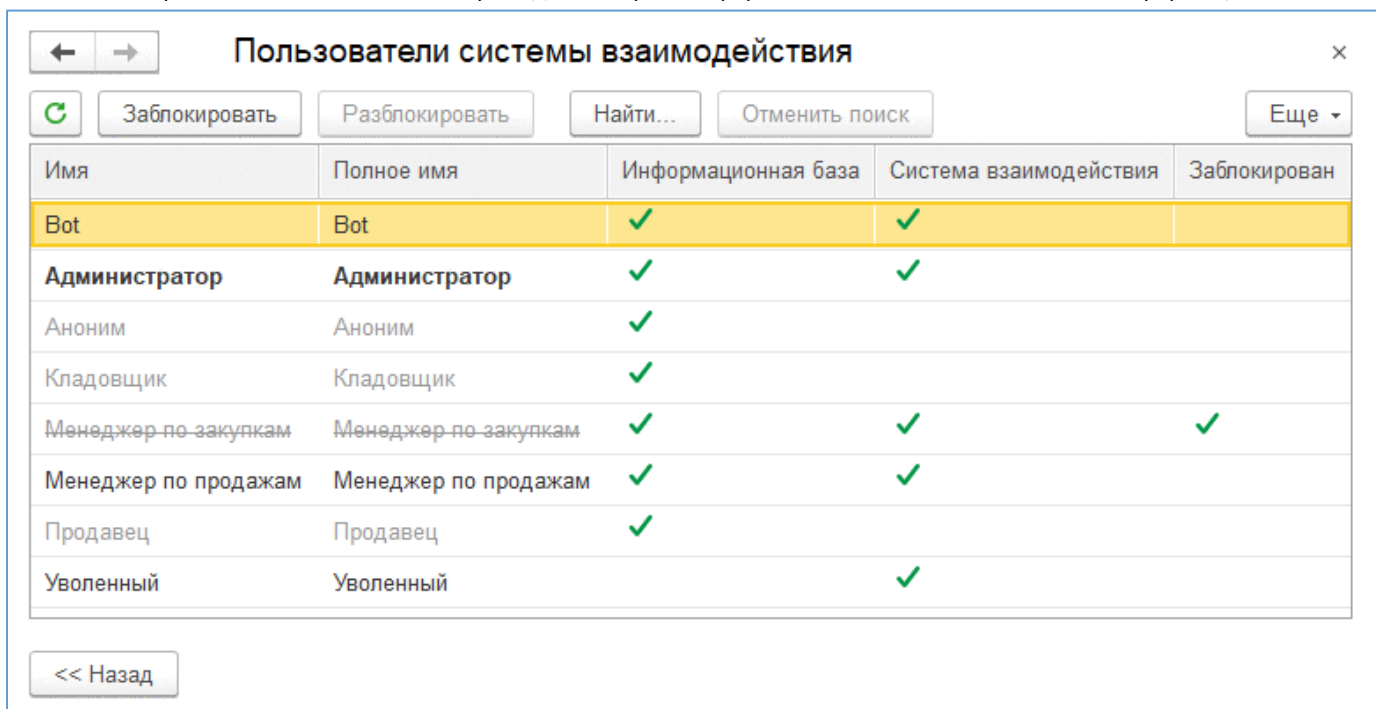
В данной форме перечислены все пары приложений и правила сопоставления пользователей и контекстов обсуждения. Если нужно выполнить объединение приложений, то следует нажать кнопку [Добавить](#).

**Рис. 81. Выбор приложений для совместного использования**

В данном диалоге отображается список всех приложений текущего владельца абонента. Необходимо отметить объединяемые приложения (не меньше двух), затем указать способ сопоставления пользователей и необходимость сопоставления контекстов обсуждения. Нажатие кнопки **Ок** приведет к выполнению операции и обновлению списка совместно используемых приложений.

Кнопка **Отменить** выполняет отмену совместного использования двух приложений (из строки, на которую установлен курсор).

Нажатие гиперссылки [Пользователи](#) приводит к открытию формы со списком пользователей информационной базы.

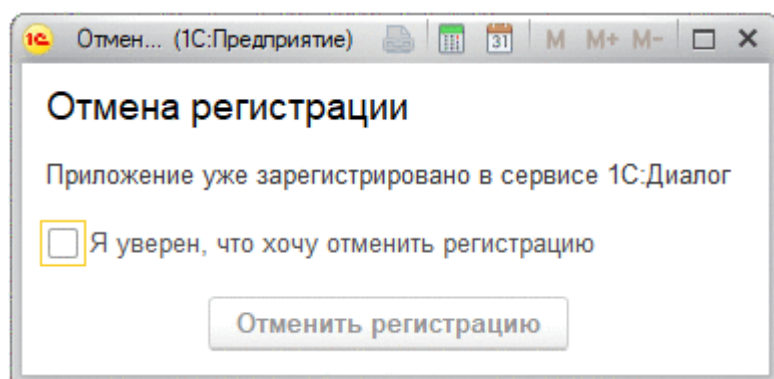
**Рис. 82. Пользователи системы взаимодействия**

В открывшемся списке пользователи выделяются соответственно своему состоянию:

- Полу жирным шрифтом выделяется текущий пользователь. На рис.82 таким пользователем является **Администратор**.
- Серым шрифтом выделяются те пользователи, которых нельзя заблокировать. Такие пользователи присутствуют только в информационной базе. На [рис.82](#) примером такого пользователя является **Аноним**.
- Зачеркнутым шрифтом выделены заблокированные пользователи и примером такого пользователя является **Менеджер по закупкам**.
- Обычным шрифтом отображаются пользователи, которые присутствуют в системе взаимодействия и которых можно заблокировать. На [рис.82](#) примером такого пользователя является **Уволенный**.

Колонки **Информационная база** и **Система взаимодействия**, соответственно, показывают наличие пользователя в самой информационной базе и в системе взаимодействия. Колонка **Заблокирован** отображает состояние блокировки пользователя. Кнопки **Заблокировать** и **Разблокировать** позволяют выполнить одноименные операции с текущим пользователем или с выделенными пользователями.

При нажатии гиперссылки **Отменить регистрацию** пользователю предлагается подтвердить это действие в специальном диалоге.



**Рис. 83. Подтверждение отмены регистрации**

Нажатие кнопки **Отменить регистрацию** отключает возможность использования системы взаимодействия в информационной базе. Однако остается возможность выполнить повторную регистрацию. После повторной регистрации будет восстановлен доступ к обсуждениям и сообщениям.

Смотри также:

- Система взаимодействия (см. [здесь](#)).
- Совместное использование приложений (см. [здесь](#)).

## ПРИЛОЖЕНИЕ 2. РУКОВОДСТВА РАЗРАБОТЧИКА - СИСТЕМА ВЗАИМОДЕЙСТВИЯ

### 30.1. ОБЩАЯ ИНФОРМАЦИЯ

**Система взаимодействия** - это механизм, позволяющий взаимодействовать между собой клиентским приложениям, серверу и пользователям одной или нескольких информационных баз. Одна информационная база (или область данных), однозначно идентифицируемая системой взаимодействия, называется **приложением**. Система взаимодействия призвана упростить автоматизацию бизнес-процессов предприятия и предоставляет пользователям приложения простой способ коммуникации в рамках выполнения должностных обязанностей. Пользоваться системой взаимодействия могут только пользователи приложения. При этом не требуется отдельно регистрироваться в системе взаимодействия. Авторизация пользователя в системе взаимодействия осуществляется системой «1С:Предприятие».

Взаимодействие может быть:

- **Не интерактивным** - в обмене информацией принимают участие только подсистемы прикладного решения, не предполагающие интерактивного взаимодействия. Такое взаимодействие может происходить в том случае, если необходимо передавать какую-либо информацию между подсистемами прикладного решения во время выполнения какой-либо активности.
- **Только интерактивным** - когда все участники взаимодействия являются живыми людьми. Например, два пользователя обсуждают возможность предоставления скидки контрагенту и ее (скидки) размер.
- **Смешанным** - когда взаимодействие выполняется между живым человеком с одной стороны и подсистемой прикладного решения с другой. Простейшим примером такого взаимодействия является робот, отвечающий на определенные запросы.



При интерактивном взаимодействии пользователи приложения могут общаться друг с другом в режиме реального времени с помощью текстовых сообщений и видеозвонков. Под термином «видеозвонок» понимается любой возможный вариант общения двух пользователей, который включает в себя как только голосовой канал общения, так и полноценный видеозвонок (и голосовая информация и изображение). Видеозвонки могут быть только интерактивными (в них участвуют только живые люди). Во время видеозвонка имеется возможность демонстрации своего экрана. Экран может быть продемонстрирован только в том случае, если для данного пользователя окно видеозвонка занимает весь монитор. В противном случае вместо демонстрации экрана отображается специальный значок [Показ экрана](#).

Взаимодействие пользователей осуществляется в рамках обсуждений. **Обсуждение** - это набор сообщений, связанных одной темой. Темой обсуждения может быть все, что угодно: документ, погода, исполнение какого-либо действия и т. д. Обсуждение может быть двух видов:

1. **Неконтекстное** обсуждение - обсуждение, которое не привязано к объекту информационной базы. В таком обсуждении могут принимать участие любые пользователи приложения, которые выбраны участниками обсуждения. Неконтекстных обсуждений может быть произвольное количество.
2. **Контекстное** обсуждение - это обсуждение, которое привязано к какому-либо объекту приложения. В таком обсуждении могут принимать участие только те пользователи приложения, которые имеют доступ к обсуждаемому объекту. Каждый объект может иметь только одно обсуждение.

К сообщениям системы взаимодействия можно прикреплять различные файлы. Для этого необходимо настроить работу сервера взаимодействия с внешним хранилищем. Внешнее хранилище должно поддерживать протокол Amazon S3. Если сервер системы взаимодействия не подключен к хранилищу - возможность прикрепления файлов будет недоступна. Если для передачи файлов используется сервис 1С:Диалог, то механизм передачи и хранения файлов имеет определенные ограничения, которые приведены на странице <https://1cdialog.com/ru/pricing/>.

Доступ к сообщениям возможен с помощью Центра оповещений, элемента [Обсуждения](#) панели открытых, специальной панели сообщений в формах объектов системы (для контекстных сообщений) и с помощью программного интерфейса, предоставляемого свойством глобального контекста [СИСТЕМАВЗАИМОДЕЙСТВИЯ](#).

Система взаимодействия реализована в клиент-серверной архитектуре и состоит из следующих частей:

- **Клиентская часть** системы взаимодействия является составной частью системы «1С:Предприятие».
- **Серверная часть** представляет собой специализированный сервер, представленный сервисом [1CDialog.com](#) или экземпляром продукта **1С:Сервер взаимодействия**. Содержимое обсуждений хранится на сервере системы взаимодействия. Информация о видеозвонках (включая факты совершения) никак не хранится в системе взаимодействия.

Обмен сообщениями между клиентской частью и сервером взаимодействия осуществляется по протоколу WebSocket. Этот протокол обеспечивает защищенную передачу данных, что позволяет безопасно передавать сообщения, с помощью которых общаются пользователи, как в локальной сети, так и через Интернет. Если доступ к серверу системы взаимодействия осуществляется через прокси-сервер, то настройки прокси-сервера получаются из системных настроек операционной системы (не поддерживается использование сценария автоматической настройки) или из конфигурационного файла [inetcfg.xml](#) (подробнее см. [здесь](#)). NTLM-аутентификация на прокси-сервере не поддерживается. Для видеозвонков используется соединение типа «точка-точка» (peer-to-peer), при этом установка соединения происходит через сервер системы взаимодействия.

Система «1С:Предприятие» обеспечивает пользовательский интерфейс для обмена сообщениями (любых поддерживаемых видов), а также предоставляет программный интерфейс, позволяющий выполнить некоторые операции, в число которых входят следующие:

- зарегистрировать или отменить регистрацию приложения в серверной части системы взаимодействия;
- объединять или отменять объединение приложений, а также получать информацию о объединенных приложениях;
- получать и устанавливать данные, позволяющие приложению взаимодействовать с сервером системы взаимодействия (данные регистрации);
- работать с пользователями системы взаимодействия;
- создавать и получать обсуждения и сообщения;
- реализовывать специализированные инструменты для работы с сообщениями.

Не поддерживается работа с несколькими копиями одной информационной базы. В том случае, если информационная база будет скопирована, то такая база сможет подключиться к сервису, однако при работе будут наблюдаться различные особенности, например:

- Обсуждения и оповещения будут доступны во всех копиях информационных баз.
- Будут наблюдаться проблемы при работе с контекстными обсуждениями для тех объектов, которые созданы после клонирования оригинальной информационной базы.
- Будут наблюдаться проблемы при работе с получателями сообщений в том случае, если пользователи в информационных базах создавались после клонирования.

В тоже время, имеется возможность объединить несколько приложений в единое информационное пространство (с точки зрения системы взаимодействия). Для этого необходимо зарегистрировать каждое приложение в системе взаимодействия, а потом объединить необходимые приложения. Объединенные приложения обладают единым списком пользователей (из всех объединенных приложений). При объединении приложений можно указать, каким образом будут сопоставляться пользователи объединяемых приложений.

Видеозвонки поддерживаются:

- В тонком клиенте при работе под управлением:
  - ОС Windows 7 и старше;
  - ОС macOS любой поддерживаемой версии.
- В веб-клиенте:
  - При работе в веб-браузерах Google Chrome, Mozilla Firefox версии 52 и старше.
  - При работе в веб-браузере Google Chrome видеозвонки поддерживаются только в том случае, если доступ к информационной базе выполняется по протоколу HTTPS.
  - Трансляция экрана:
    - Поддерживается при работе в веб-браузерах Google Chrome, Mozilla Firefox.
    - Поддерживается только при использовании протокола HTTPS.
    - При использовании веб-браузера Google Chrome доступна трансляция не только всего экрана, но и отдельной закладки.
    - В веб-браузере Google Chrome требуется установить расширение работы веб-браузера и предоставить расширению возможность выполнять снимок содержимого экрана.

### 30.2. УСТРОЙСТВО СИСТЕМЫ ВЗАИМОДЕЙСТВИЯ

Работа системы взаимодействия построена на клиент-серверном обмене данными между системой «1С:Предприятие» и сервером системы взаимодействия. При этом и клиентское и серверное приложение системы «1С:Предприятие» выступают в роли клиентов сервера системы взаимодействия. Взаимодействие организуется вокруг приложения системы взаимодействия (информационная база или область данных).

Чтобы начать использование системы взаимодействия, приложение, пользователи которого будут участвовать в этом взаимодействии, необходимо зарегистрировать в сервисе системы. Во время регистрации приложения в сервисе системы взаимодействия создается (и сохраняется) следующая информация:

- Уникальный идентификатор приложения в сервисе. Этот идентификатор сохраняется как в самой информационной базе, так на сервере системы взаимодействия. Этот идентификатор нельзя получить и установить самостоятельно.
- Криптографические ключи, которые используются для обеспечения безопасной передачи сообщений между клиентской и серверной частями системы взаимодействия. Эти ключи хранятся как в приложении, так и на сервере системы взаимодействия. Каждая сторона взаимодействия хранит свой набор ключей.
- В сервисе также сохраняется имя регистрируемого приложения и адрес электронной почты, который используется при регистрации информационной базы. Адрес электронной почты, который используется при регистрации приложения в сервисе, будет олицетворять абонента сервиса. Один абонент сервиса может зарегистрировать несколько приложений. Этот абонент будет считаться **владельцем абонента**.

В дальнейшем адрес сервиса взаимодействия, имя приложения, электронная почта и код активации (совокупно) будут называться «параметры регистрации приложения в сервисе». В тоже время параметры регистрации приложения в сервисе, идентификатор приложения в сервисе и криптографические ключи, используемые для обмена, будут совокупно называться «данными регистрации приложения».

Для того чтобы прекратить работу системы взаимодействия, необходимо выполнить отмену регистрации приложения в сервисе. В этом случае в приложении удаляются криптографические ключи, используемые для обеспечения обмена и, как следствие, обмен сообщениями становится невозможным. При повторной регистрации того же приложения будет создана новая пара ключей. В результате обмен сообщениями будет восстановлен и станет доступна вся история сообщений. Для корректного восстановления обмена сообщениями важно, чтобы у приложения остался неизменным уникальный идентификатор этого приложения в сервисе. Если это правило нарушено - восстановить доступ к истории сообщений будет невозможно.

Для того чтобы приложение смогло зарегистрироваться в сервисе системы взаимодействия, в нем должны быть заданы пользователи. В противном случае приложение не сможет зарегистрироваться в сервисе и, как следствие, в нем будет невозможно использовать возможности системы взаимодействия. В процессе обмена информацией могут принимать участие только те пользователи приложения, которые зарегистрированы в сервисе взаимодействия. Для каждого пользователя приложения создается парный пользователь в системе взаимодействия. Создание такого пользователя выполняется при первой аутентификации пользователя приложения после регистрации этого приложения в сервере системы взаимо-



действия или с помощью методов встроенного языка. При последующих аутентификациях пользователя, в системе взаимодействия автоматически актуализируются имя и полное имя пользователя информационной базы. Сопоставление пользователей (приложения и сервиса взаимодействия) выполняется по уникальному идентификатору пользователя информационной базы системы «1С:Предприятие». Если требуется другой алгоритм синхронизации списка пользователей - прикладной разработчик должен реализовать его самостоятельно.

### 30.3. ИНТЕРАКТИВНОЕ ВЗАИМОДЕЙСТВИЕ

#### 30.3.1. ПАРАМЕТРЫ ПОЛЬЗОВАТЕЛЕЙ

После подключения приложения к сервису взаимодействия, появляется возможность указания дополнительных параметров для пользователя, который зарегистрирован в системе взаимодействия.

Поддерживается хранение полного имени пользователя, картинки (аватара), телефонного номера и адреса электронной почты.

#### 30.3.2. ВИДЫ ОБСУЖДЕНИЙ

##### 30.3.2.1. НЕКОНТЕКСТНЫЕ ОБСУЖДЕНИЯ

Неконтекстные обсуждения не привязаны к каким-либо объектам приложения. Любой пользователь может создать произвольное количество таких обсуждений. Неконтекстные обсуждения могут быть двух видов: с любым количеством участников (вариант «группового обсуждения») и только с двумя участниками (вариант «один на один»). При создании обсуждения его участниками являются пользователь, создавший обсуждение, и те пользователи, которых он указал при создании обсуждения. В дальнейшем любой участник обсуждения может:

- просматривать сообщения обсуждения;
- добавлять сообщения в обсуждение;
- следующие возможности доступны только для группового обсуждения (и недоступны для обсуждений «один на один»):
- изменять название (тему) обсуждения;
- добавлять новых участников обсуждения (любых пользователей информационной базы);
- покинуть обсуждение.

Если обсуждение покидает последний участник, то обсуждение становится недоступным. При этом на сервере взаимодействия содержимое обсуждений сохраняется. Неконтекстное обсуждение видимо только для участников обсуждения.

Создание неконтекстного обсуждения выполняется из основной формы системы взаимодействия, которая становится доступной после регистрации приложения в сервисе взаимодействия.

Основная форма системы взаимодействия всегда является второй (после начальной страницы) формой в панели открытых. Эту форму нельзя закрыть и перетащить в другое место панели. Если в прикладном решении отсутствует начальная страница, основная форма системы взаимодействия становится первой формой в панели открытых.

При создании нового обсуждения предлагается указать тему сообщения и участников. Инициатор обсуждения автоматически добавляется в список участников.

Для создания обсуждения «один на один» необходимо в строке поиска формы [Обсуждения](#) указать пользователя, с которым необходимо создать обсуждение. Для таких обсуждений не указывается тема и состав участников. Тема обсуждения «один на один» не может быть задана, а состав участников автоматически формируется системой.

В списке обсуждений показываются все неконтекстные обсуждения, по которым для пользователя есть уведомления. Если таких обсуждений меньше 50, то список дополняется неконтекстными обсуждениями, в которых пользователь является участником, и по которым для пользователя нет уведомлений. Эти обсуждения добавляются в список в порядке убывания даты последнего сообщения в обсуждении, пока размер списка не достигнет 50 элементов.

Поиск в списке обсуждений работает по темам обсуждений и именам пользователей. Поиск в списке сообщений работает только по сообщениям выбранного обсуждения. Не поддерживается возможность поиска обсуждения по тексту сообщений.

При появлении в обсуждении новых сообщений, участники обсуждения получают оповещения.

##### 30.3.2.2. КОНТЕКСТНЫЕ ОБСУЖДЕНИЯ

Контекстные обсуждения (как уже было сказано ранее) привязаны к объектам приложения. Если приложение подключено к сервису взаимодействия, то для форм элементов ссылочных типов и форм записей регистров сведений, становится до-

ступна команда формы **Обсуждение**. Для каждого объекта существует только одно контекстное обсуждение. В обсуждении могут принимать участие все пользователи приложения (или приложений), которые имеют право **ЧТЕНИЕ** для обсуждаемого объекта.

Контекстное обсуждение выполняется в форме обсуждаемого объекта (в правой части). Такое обсуждение не имеет заголовка. Открытие и закрытие списка сообщений выполняется циклическим нажатием кнопки **Обсуждение**.

Несмотря на то, что в обсуждении по умолчанию могут принимать участие все пользователи приложения (с соответствующими правами), при создании сообщения в контекстном обсуждении имеется возможность выбрать адресатов сообщения. В данном случае адресаты выбираются для того, чтобы отобразить им оповещения о новом сообщении. Остальные пользователи получают оповещение только в том случае, если они подпишутся на получение оповещений о новых сообщениях в этом обсуждении.

Команда **Обсуждение** не выводится в формах, которые блокируют окно владельца или весь интерфейс. Эти формы обычно используются для отображения и редактирования данных, не имеющих самостоятельного прикладного значения, поэтому механизм обсуждений для них не предусмотрен.

Для форм, которые не блокируют окно владельца или интерфейс приложения, имеется возможность управлять доступностью команды **Обсуждение** в форме. Управление осуществляется с помощью свойства управляемой формы **ОТОБРАЖЕНИЕОБСУЖДЕНИЙ**. Установка данного свойства в значение **Не отображать** приведет к тому, что команда **Обсуждение** не будет доступна в форме.

Следует, однако, учитывать, что контекстное обсуждение доступно только в той форме, навигационная ссылка которой указывает на данные, которые хранятся в информационной базе, или команду.

Смотри также:

- Права доступа (см. [здесь](#)).

### 30.3.3. ВИДЫ СООБЩЕНИЙ

#### 30.3.3.1. ТЕКСТОВЫЕ СООБЩЕНИЯ

Основным механизмом в системе взаимодействия считается обмен текстовыми сообщениями. Размер текстового сообщения ограничен значением 4 000 символов. Само текстовое сообщение может быть и просто текстом и форматированной строкой (объект типа **ФОРМАТИРОВАННАЯСТРОКА**). Указание форматированного сообщения доступно только при программном формировании сообщения. Когда форматированная строка присваивается свойству **СООБЩЕНИЕСИСТЕМЫВЗАИМОДЕЙСТВИЯ.ТЕКСТ**, то форматирование сохраняется, но картинки (если таковые есть в строке) удаляются. Форматирование сообщения будет сохранено в следующих случаях:

- во всплывающих уведомлениях,
- в центре оповещений,
- в формах обсуждений.

В остальных случаях текст сообщения отображается без форматирования. При отображении форматированного сообщения автоматическое распознавание гиперссылок не выполняется, а «смайлики» распознаются только для следующих комбинаций символов и эмотиконов (специальных символов Unicode):

Код	Текстовый аналог	Из встроенного языка
U+1F610	:	Символ (55357) + Символ (56848)
U+1F600	:D	Символ (55357) + Символ (56832)
U+1F642	:)	Символ (55357) + Символ (56898)
U+1F641	:(	Символ (55357) + Символ (56897)
U+1F62E	:o	Символ (55357) + Символ (56878)
U+1F61B	:P	Символ (55357) + Символ (56859)
U+1F609	;) )	Символ (55357) + Символ (56841)

#### 30.3.3.2. ВИДЕОЗВОНКИ

Для использования видеозвонков должны быть выполнены следующие требования:

1. Сервер взаимодействия должен разрешать видеозвонки для абонента, который выполнял регистрацию приложения.

2. Клиентское приложение, которое применяется для доступа к приложению, позволяет использовать видеозвонки.
3. Компьютеры пользователей - участников видеозвонка должны быть оснащены микрофонами и динамиками. При наличии видеокамеры у участника видеозвонка обеспечивается показ изображения видеокамеры другому участнику видеозвонка.

В системе взаимодействия поддерживаются видеоконференции с более чем двумя участниками. Участников видеоконференции можно добавлять и удалять в любой момент времени.

Максимальное количество участников видеоконференции ограничивается настройками сервера взаимодействия. По умолчанию в видеоконференции не может принимать участия более 10 абонентов.

### 30.3.4. ОПОВЕЩЕНИЯ

Для информирования пользователей о новых сообщениях, система взаимодействия использует оповещения. Оповещение формируется для каждого нового сообщения в рамках каждого обсуждения. Оповещения направляются:

- при добавлении сообщения в контекстное обсуждение - всем пользователям, которых автор сообщения указал в качестве адресатов, а также пользователям, включившим режим наблюдения для этого контекстного обсуждения (с учетом правил оповещения);
- при добавлении сообщения в неконтекстное обсуждение - всем участникам обсуждения (с учетом правил оповещения);
- при пропущенном видеозвонке - пользователю, пропустившему видеозвонок.

Обсуждение содержит сообщения, отправленные пользователями. При появлении нового сообщения возникает оповещение. Настройки системы позволяют управлять как доступностью сообщений, так и отображением оповещений. И получение сообщений, и получение оповещений управляются с помощью некоторых настроек: права пользователя, настройки получения оповещений и настройка отображения оповещений о новых сообщениях. Рассмотрим поведение системы в разрезе этих настроек.

Как известно, обсуждения бывают двух видов: контекстные и неконтекстные. Права пользователя для этих обсуждений формируются по-разному:

- Для контекстного обсуждения: пользователь должен иметь право **ЧТЕНИЕ** на обсуждаемый объект.
- Для неконтекстного обсуждения: должен быть участником обсуждения.

Оповещения о новых сообщениях будут получены пользователем при следующих настройках обсуждений:

- Для контекстного обсуждения: в настройках обсуждения пользователь указал **Оповещать обо всех сообщениях** или является адресатом сообщения.
- Для неконтекстного обсуждения: в настройках обсуждения пользователь указал **Оповещать о новых сообщениях**.

Таким образом, если у пользователя есть права на обсуждение и он решил получать оповещения - оповещения о новых сообщениях будут отображаться в Центре оповещений.

Режим получения оповещений управляется с помощью методов **ПолучитьРежимНаблюдения()** и **УстановитьРежимНаблюдения()**. Параметр, описывающий режим наблюдения, фактически означает следующее (в зависимости от вида обсуждения):

- Контекстное обсуждение:
  - **ИСТИНА** - соответствует режиму **Оповещать обо всех новых сообщениях**.
  - **ЛОЖЬ** - соответствует режиму **Оповещать только о новых сообщениях, адресованных мне**.
- Неконтекстное обсуждение:
  - **ИСТИНА** - соответствует режиму **Оповещать о новых сообщениях**.
  - **ЛОЖЬ** - соответствует режиму **Не оповещать о новых сообщениях**.

Как и установка режима получения оповещений, режим наблюдения может устанавливаться как для текущего, так и для произвольного пользователя. Во втором случае для выполнения операции требуется установка привилегированного режима.

Последняя настройка управляет отображением всплывающих оповещений о новых сообщениях:

- В настройках пользователя выбран режим **Не беспокоить** - оповещения не будут отображаться для всех обсуждений.
- В настройках обсуждения выбран режим **Не беспокоить** - оповещения не будут отображаться для того обсуждения, где указан режим **Не беспокоить**.

Для обсуждений имеется возможность управлять режимом **Не беспокоить** с помощью встроенного языка. Для этого следует использовать методы менеджера системы взаимодействия **ПолучитьОтображениеОповещенийОбсуждения()** и **УстановитьОтображениеОповещенийОбсуждения()**. Режим отобра-

жения оповещений может задаваться как для текущего пользователя, так и для любого пользователя системы взаимодействия. Для того, чтобы работать с режимом отображения оповещений для любого пользователя - необходимо использование привилегированного режима.

Смотри также:

- Привилегированный режим (см. [здесь](#)).
- Права доступа (см. [здесь](#)).

### 30.4. НЕ ИНТЕРАКТИВНОЕ ВЗАИМОДЕЙСТВИЕ

#### 30.4.1. ОБЩАЯ ИНФОРМАЦИЯ

Большинство операций, выполняемых с использованием системы взаимодействия, можно выполнить интерактивно или с использованием методов встроенного языка. В данном разделе будут рассмотрены примеры реализации тех или иных возможностей работы с системой взаимодействия.

**ВНИМАНИЕ!** Примеры, приведенные в данном разделе, не являются законченными, а служат лишь для иллюстрации работы описываемых механизмов.

Для доступа к программному интерфейсу системы взаимодействия предназначен менеджер системы взаимодействия, который представлен свойством глобального контекста `СИСТЕМАВЗАИМОДЕЙСТВИЯ`. Для вызова какого-либо метода следует указывать `СИСТЕМАВЗАИМОДЕЙСТВИЯ.ИмяМетода ( )`. Однако для упрощения текста, во время описания того или иного метода префикс `СИСТЕМАВЗАИМОДЕЙСТВИЯ` будет опускаться. Примеры, естественно, будут содержать корректный синтаксис вызова.

#### 30.4.2. РЕГИСТРАЦИЯ/ОТМЕНА РЕГИСТРАЦИИ ПРИЛОЖЕНИЯ

Для выполнения регистрации приложения на сервере системы взаимодействия служит пара методов: `НачатьРегистрациюИнформационнойБазы ( )` и `ВыполнитьРегистрациюИнформационнойБазы ( )`. Первый метод работает на стороне клиентского приложения и является асинхронным. Второй метод выполняется на стороне сервера и является синхронным.

Оба метода получают на вход параметры регистрации (объекта `ПАРАМЕТРЫРЕГИСТРАЦИИИнформационнойБазыСистемыВзаимодействия`), а возвращают результат регистрации (объект `РЕЗУЛЬТАТРЕГИСТРАЦИИИнформационнойБазыСистемыВзаимодействия`). Клиентский вариант метода также требует ссылку на обработчик оповещения (объект `ОПИСАНИЕОПОВЕЩЕНИЯ`), в который будет передан результат регистрации. В серверном варианте метода результат регистрации будет результатом работы метода.

В объекте `РЕЗУЛЬТАТРЕГИСТРАЦИИИнформационнойБазыСистемыВзаимодействия` статус регистрации отображается свойством `РЕГИСТРАЦИЯВыполнена`, а если регистрация не завершена, то свойство `ТЕКСТСообщения` содержит информацию для пользователя.

Собственно регистрация выполняется за несколько шагов:

1. На первом шаге на сервер системы взаимодействия передаются параметры регистрируемого приложения. Для этого необходимо заполнить свойства `АдресЭлектроннойПочты` и `АдресСервера` объекта `ПАРАМЕТРЫРЕГИСТРАЦИИИнформационнойБазыСистемыВзаимодействия` и вызвать метод `НачатьРегистрациюИнформационнойБазы ( )` / `ВыполнитьРегистрациюИнформационнойБазы ( )`.
2. На следующем шаге сервер системы взаимодействия присылает на адрес электронной почты, который был указан на предыдущем шаге, письмо с кодом активации.
3. На последнем шаге необходимо повторить вызов метода `НачатьРегистрациюИнформационнойБазы ( )` / `ВыполнитьРегистрациюИнформационнойБазы ( )`, но кроме свойств `АдресЭлектроннойПочты` и `АдресСервера`, следует заполнить также свойство `КодАктивации`. В это свойство следует поместить значение, которое пришло в электронном письме от сервера системы взаимодействия.

В качестве адреса сервера системы взаимодействия может выступать адрес сервера [1cDialog.com](https://1cdialog.com) (`wss://1cdialog.com:443`) или адрес самостоятельно развернутого сервера системы взаимодействия

4. Если никаких ошибок не происходило, то к этому моменту регистрация успешно завершена. Если при вызовах метода регистрации происходили какие-либо ошибки, то информация для пользователя будет передана в свойстве `ТЕКСТСообщения` объекта `РЕЗУЛЬТАТРЕГИСТРАЦИИИнформационнойБазыСистемыВзаимодействия`. Способ получения объекта с результатом регистрации зависит от используемого метода регистрации. После окончания регистрации сформированы и сохранены все криптографические ключи, которые необходимы для работы с системой взаимодействия.

Свойство `ПАРАМЕТРЫРЕГИСТРАЦИИИНФОРМАЦИОННОЙБАЗЫСИСТЕМЫВЗАИМОДЕЙСТВИЯ.ИмяИНФОРМАЦИОННОЙБАЗЫ` служит для того, чтобы указать представление регистрируемого приложения.

Чтобы отказаться от регистрации приложения в сервисе, следует использовать методы `НачатьОтменуРегистрацииИнформационнойБазы()` или `ОтменитьРегистрациюИнформационнойБазы()`.

После успешного завершения работы метода пользователи текущего приложения не смогут пользоваться системой взаимодействия. В частности, будут удалены все необходимые криптографические ключи. Однако если выполнить повторную регистрацию приложения, то работа с системой взаимодействия будет восстановлена и будет восстановлен доступ ко всей переписке, которая была сформирована до отключения.

Для полного отключения приложение от сервиса взаимодействия (включая потерю доступа ко всей переписке, сформированной при работе в приложении), следует воспользоваться методом `УстановитьДанныеРегистрацииИнформационнойБазы()`, куда в качестве данных регистрации передать значение `Неопределено`. После выполнения такого действия регистрация приложения отменяется, и доступ к сообщениям теряется безвозвратно.

У метода `УстановитьДанныеРегистрацииИнформационнойБазы()` существует парный метод `ПолучитьДанныеРегистрацииИнформационнойБазы()`, который позволяет получить всю информацию о подключении к серверу системы взаимодействия в виде единого объекта. Данный объект содержит всю информацию, которая указывалась при регистрации, а также все криптографические ключи, которые используются для работы с сервером системы взаимодействия. Если передать эти данные в качестве параметра метода `УстановитьДанныеРегистрацииИнформационнойБазы()`, то система взаимодействия будет считать приложение, в которой было выполнено это действие, полностью эквивалентной тому приложению, в котором были получены данные, использованные для установки. Поэтому основное предназначение пары методов `Получить/УстановитьДанныеРегистрацииИнформационнойБазы()` - это миграция области данных между различными базами данных.

Последнее, о чем стоит упомянуть в контексте регистрации приложения в сервисе взаимодействия, это проверка взаимодействия приложения и сервера взаимодействия. Для этого существуют два метода:

- Метод `ИнформационнаяБазаЗарегистрирована()` проверяет, что приложение зарегистрировано в сервисе и эта регистрация не отключена.
- Метод `ИспользованиеДоступно()` проверяет, что приложение зарегистрировано в сервисе, регистрация не отключена и в приложении есть пользователи.

### 30.4.3. РАБОТА С ПОЛЬЗОВАТЕЛЯМИ

#### 30.4.3.1. СОЗДАНИЕ И ОБНОВЛЕНИЕ ПОЛЬЗОВАТЕЛЕЙ СИСТЕМЫ ВЗАИМОДЕЙСТВИЯ

После того, как выполнена регистрация приложения в сервисе взаимодействия, можно выполнить синхронизацию списка пользователей между пользователями приложения и пользователями системы взаимодействия. Для выполнения этой операции платформа предоставляет целый набор методов.

Текущий пользователь уже создан в системе взаимодействия - это происходит автоматически, при первом интерактивном входе в систему «1С:Предприятие» после регистрации приложения в сервисе. Но может потребоваться перенести всех существующих пользователей приложения в сервис сразу после регистрации. Для этого следует воспользоваться методом `СоздатьПользователя()`. В качестве параметра метода выступает объект `ПользовательИнформационнойБазы`. Для получения списка пользователей информационной базы можно использовать метод `ПользователиИнформационнойБазы.ПолучитьПользователей()`.

*Пример:*

```
УстановитьПривилегированныйРежим(Истина);
ПользователиИБ = ПользователиИнформационнойБазы.ПолучитьПользователей();
Для каждого ПользовательИБ Из ПользователиИБ Цикл
    Попытка
        ИдентификаторПользователяСВ = СистемаВзаимодействия.ПолучитьИдентификаторПользователя(ПользовательИБ.УникальныйИдентификатор);
    Исключение
        ПользовательСВ = СистемаВзаимодействия.СоздатьПользователя(ПользовательИБ);
        // возможно дозаполнение свойств пользователя системы взаимодействия
        ПользовательСВ.Записать();
    КонецПопытки;
КонецЦикла;
УстановитьПривилегированныйРежим(Ложь);
```



## 1С:СЕРВЕР ВЗАИМОДЕЙСТВИЯ: Краткое руководство по установке и настройке

В данном примере выполняется попытка получения идентификатора пользователя системы взаимодействия для каждого пользователя приложения. Если пользователь приложения еще не создавался в системе взаимодействия - будет сгенерировано исключение. В этом случае на основании пользователя приложения будет создан пользователь системы взаимодействия.

При работе с пользователями системы взаимодействия следует обратить внимание на свойство **КлючСопоставления**. Данное свойство может использоваться для сопоставления пользователей при объединении приложений. При работе с пользователями системы взаимодействия следует обратить внимание еще на одно полезное свойство: **ИдентификаторПользователяИнформационнойБазы**. Данное свойство содержит уникальный идентификатор пользователя информационной базы, который можно использовать для поиска пользователя **ПользователяИнформационнойБазы**. **НайтиПоУникальномуИдентификатору()**. Если свойство **ИдентификаторПользователяИнформационнойБазы** равно значению **Неопределено**, это значит, что пользователь информационной базы, соответствующий пользователю системы взаимодействия, был удален из информационной базы. Повторно (после создания пользователя системы взаимодействия) сопоставить пользователя информационной базы и пользователя системы взаимодействия невозможно.

Когда пользователю информационной системы необходимо запретить использование системы взаимодействия, можно воспользоваться свойством **ПользовательИнформационнойБазы.Заблокирован**. Такая потребность может возникнуть, например, в том случае, когда пользователь уволился из компании, где использовалась система взаимодействия. Если пользователь системы взаимодействия заблокирован, то такой пользователь:

- Не может устанавливать соединение с сервером системы взаимодействия.
- Перестает получать уведомления.
- Не отображается в подборе пользователей системы взаимодействия.
- Такому пользователю нельзя сделать видеозвонок.
- Отображается с использованием специального значка.
- При отправке сообщения такому пользователю, добавлению его в список участников обсуждения или в видеоконференцию, возникает исключение.

Смотри также:

- Совместное использование приложений (см. [здесь](#)).

### 30.4.3.2. ВЫБОР ПОЛЬЗОВАТЕЛЕЙ СИСТЕМЫ ВЗАИМОДЕЙСТВИЯ

#### 30.4.3.2.1. ОБЩАЯ ИНФОРМАЦИЯ

При указании пользователя в системе взаимодействия, платформа предоставляет возможность управлять этим процессом. Для этого предоставляется обработчик события **АвтоПодборПользователейСистемыВзаимодействия**. Этот обработчик события может располагаться в модуле формы клиентского приложения или в модуле приложения. Имеется возможность реализовать собственную форму выбора пользователей системы взаимодействия. Для этого существует обработчик события **ОбработкаПолученияФормыВыбораПользователейСистемыВзаимодействия**. Этот обработчик также может располагаться в модуле формы клиентского приложения или модуле приложения. Также, для указания формы выбора пользователей системы взаимодействия можно использовать свойство конфигурации **ОсновнаяФормаВыбораПользователейСистемыВзаимодействия**.

Система позволяет обработать выбор пользователя в следующих случаях:

- выбор получателя сообщения в контекстном обсуждении;
- выбор участника обсуждения в неконтекстном обсуждении;
- выбор участника видеоконференции.

Назначение выбора пользователей описывается системным перечислением **НазначениеВыбораПользователейСистемыВзаимодействия**. Значение данного перечисления передается в обработчики событий автоподбора пользователя и выбора формы в параметре **НазначениеВыбора**. С его помощью разработчик может понять, где выполняется то или иное действие.

Общая схема выбора пользователя выглядит следующим образом:

- При создании нового сообщения, пользователь начинает вводить имя своего корреспондента. При этом срабатывает событие автоподбора пользователя системы взаимодействия. Выбранные пользователи отображаются в виде списка выбора у поля ввода имени пользователя.
- После того, как сформирован список выбора, платформа определяет, существует ли в данном прикладном решении специальная форма выбора пользователей системы взаимодействия. Если такая форма есть, то она отображается в виде гиперссылки **Показать все** в нижней части выпадающего списка пользователей. Определение формы выбора выполняется с помощью специального события, а также свойства конфигурации.

- Из списка выбора можно выбрать одного пользователя, а из формы выбора - сразу несколько. Выбранные пользователи попадают в соответствующее поле ввода и выбор завершается. Можно начинать выбирать нового пользователя.

### 30.4.3.2.2. ОПРЕДЕЛЕНИЕ СПИСКА ПОЛЬЗОВАТЕЛЕЙ

Более подробно рассмотрим последовательность событий, возникающих при вводе пользователей:

1. Выполняется вызов обработчика события **АвтоПодборПользователейСистемыВзаимодействия** формы клиентского приложения. Обработчик данного события может быть вызван только в том случае, если происходит выбор адреса в контекстном обсуждении.

На вход обработчик получает сформированный системой список пользователей. Данный список может быть модифицирован или полностью заменен.

Если при завершении обработчика параметр **СтандартнаяОбработка** установлен в значение **Ложь**, то платформа будет использовать тот список пользователей, который получен из данного обработчика. Если параметр **СтандартнаяОбработка** имеет значение **Истина** - будет вызван обработчик автоподбора из модуля приложения (следующий шаг).

2. Выполняется вызов обработчика события **АвтоПодборПользователейСистемыВзаимодействия** модуля клиентского приложения.

На вход обработчик получает список пользователей, который сформирован системой или обработчиком события **АвтоПодборПользователейСистемыВзаимодействия** формы клиентского приложения. Данный список может быть модифицирован или полностью заменен.

Для формирования выпадающего списка пользователей система будет использовать в точности тот список пользователей, который получен из данного обработчика.

В обработку автоподбора пользователей передаются следующие параметры:

- Текст, введенный пользователем (параметр **Текст**).
- Список пользователей системы взаимодействия, которых по введенному тексту выбрала сама платформа (параметр **ДанныеВыбора**). Разработчик может переопределить данный список.
- Для чего выполняется подбор пользователей (параметр **НазначениеВыбора**).
- Описание формы клиентского приложения, в которой расположено контекстное обсуждение, для которого выполняется подбор получателей сообщения (параметр **Форма**).
- Идентификатор обсуждения, в которое добавляются пользователи (параметр **ИдентификаторОбсуждения**).

Данные, которые находятся в параметре обработчика события **ДанныеВыбора** после выхода из обработчика, будут использованы в дальнейшей цепочке обработчиков или непосредственно системой взаимодействия.

В качестве значения параметра **ДанныеВыбора** выступает список значений. В качестве значения элемента этого списка может выступать:

- Значение типа **ИдентификаторПользователяСистемыВзаимодействия**. Если значение данного типа будет «пустым», то при формировании списка получателей сообщения контекстного обсуждения, такое значение будет интерпретироваться как **Никого не оповещать**. В остальных случаях такое значение будет игнорироваться. «Пустое» значение типа **ИдентификаторПользователяСистемыВзаимодействия** можно сформировать с помощью конструктора объекта **ИдентификаторПользователяСистемыВзаимодействия** без параметров.
- Значение типа **УникальныйИдентификатор**. Платформа будет «считать», что это идентификатор пользователя информационной базы. По данному идентификатору будет выполнен поиск пользователя системы взаимодействия.
- Массив значений типа **ИдентификаторПользователяСистемыВзаимодействия** или **УникальныйИдентификатор**. В этом случае в списке выбора формируется «группа» пользователей, для которой можно задать свое представление. В качестве примера можно отметить группировку пользователей по функциональному признаку: в группе Продавцы будут перечислены все пользователи, которые выполняют функцию продавцов. В этом случае выбор группы приведет к добавлению в список получателей сразу большого количества пользователей, без необходимости выбора каждого пользователя по одному.

Если для элемента списка значений задана картинка - она будет выводиться в списке выбора. Если картинка не задана, то в случае массива будет выводиться картинка группы пользователей, а для конкретных пользователей будет выводиться картинка заданного пользователя.

Представление элемента списка значений задается или в соответствующем свойстве элемента списка значения или формируется системой автоматически:

- Для конкретного значения будет выводиться полное имя пользователя.
- Для группы - полные имена пользователей, входящих в группу, разделенные запятой.

### 30.4.3.2.3. ОПРЕДЕЛЕНИЕ ФОРМЫ ВЫБОРА ПОЛЬЗОВАТЕЛЕЙ

После того, как сформирован выпадающий список пользователей, система определяет, существует ли в прикладном решении форма выбора пользователей системы взаимодействия. Это определение выполняется после каждого формирования списка пользователей. Для этого используется следующая последовательность действий:

1. Выполняется вызов обработчика события `ОБРАБОТКАПОЛУЧЕНИЯФОРМЫВЫБОРАПОЛЬЗОВАТЕЛЕЙСИСТЕМЫВЗАИМОДЕЙСТВИЯ` модуля формы клиентского приложения. Обработчик данного события может быть вызван только в том случае, если происходит выбор адресата в контекстном обсуждении.

Для указания имени формы выбора служит параметр обработчика `ВЫБРАННАЯФОРМА`. Дальнейшее поведение, как обычно, зависит от значения параметра обработчика `СТАНДАРТНАЯОБРАБОТКА`:

- Параметр имеет значение `ИСТИНА`: вызывается обработчик выбора формы модуля приложения.
  - Параметр имеет значение `ЛОЖЬ`: платформа использует в качестве формы выбора пользователей системы взаимодействия ту форму, которая указана в параметре `ВЫБРАННАЯФОРМА`.
2. Выполняется вызов обработчика `ОБРАБОТКАПОЛУЧЕНИЯФОРМЫВЫБОРАПОЛЬЗОВАТЕЛЕЙСИСТЕМЫВЗАИМОДЕЙСТВИЯ` модуля приложения.

Поведение обработчика подобно поведению аналогичного обработчика модуля формы. Однако имеется отличие, которое заключается в следующем: указание в данном обработчике значения `ИСТИНА` для параметра `СТАНДАРТНАЯОБРАБОТКА` приведет к тому, что система попытается использовать в качестве формы выбора пользователей форму, которая указана в свойстве конфигурации `ОСНОВНАЯФОРМАВЫБОРАПОЛЬЗОВАТЕЛЕЙСИСТЕМЫВЗАИМОДЕЙСТВИЯ`. Если это свойство конфигурации не указано - форма выбора пользователя системы взаимодействия не будет использована. Это означает, что в выпадающем списке пользователей не будет гиперссылки [Показать все](#).

В процедуру обработки события выбора формы выбора пользователей системы взаимодействия передаются следующие параметры:

- Для чего выполняется подбор пользователей (параметр `НАЗНАЧЕНИЕВЫБОРА`).
- В какой форме расположено контекстное обсуждение, если происходит подбор получателей сообщения (параметр `ФОРМА`).
- Идентификатор обсуждения, в которое добавляются пользователи (параметр `ИДЕНТИФИКАТОРОБСУЖДЕНИЯ`).
- Параметры, которые будут использованы для открытия формы выбора пользователей системы взаимодействия (параметр `ПАРАМЕТРЫ`). Платформа формирует в данной структуре следующие параметры:
  - Параметр `РЕЖИМВЫБОРА` установлен в значение `ИСТИНА`.
  - Параметр `МНОЖЕСТВЕННЫЙВЫБОР` установлен в значение `ИСТИНА`.
  - Параметр `ЗАКРЫВАТЬПРИВЫБОРЕ` установлен в значение `ИСТИНА`.
- Форма, которая будет использоваться для выбора пользователей системы взаимодействия (параметр `ВЫБРАННАЯФОРМА`). Если данный параметр при выходе из обработчика не равен пустой строке - в списке подбора пользователей системы взаимодействия добавляется гиперссылка [Показать все](#) и при нажатии на эту гиперссылку открывается выбранная форма. Параметры открытия формы задаются значением параметра `ПАРАМЕТРЫ` обработчика и могут быть модернизированы в обработчике определения формы выбора. Владелец открываемой формы выбора будет указана форма, в которой отображается обсуждение.

Форма выбора, используемая для выбора пользователей системы взаимодействия, должна возвращать выбранных пользователей в качестве параметра метода `ЗАКРЫТЬ()`. В качестве параметра метода `ЗАКРЫТЬ()` может выступать:

- Значение типа `ИДЕНТИФИКАТОРПОЛЬЗОВАТЕЛЯСИСТЕМЫВЗАИМОДЕЙСТВИЯ`.
- Значение типа `УНИКАЛЬНЫЙИДЕНТИФИКАТОР`. Платформа будет «считать», что это идентификатор пользователя информационной базы. И по данному идентификатору будет выполнен поиск пользователя системы взаимодействия.
- Массив из значений вышеперечисленных типов.

### 30.4.4. РАБОТА С ОБСУЖДЕНИЯМИ

#### 30.4.4.1. БАЗОВЫЕ ВОЗМОЖНОСТИ

Для работы с обсуждением необходимо это обсуждение создать. Для этого предназначен метод `СОЗДАТЬОБСУЖДЕНИЕ()`. В результате работы метода создается объект, с помощью которого имеется возможность настраивать обсуждение и работать с ним. Созданное обсуждение еще не записано. Для записи следует явным образом вызвать метод `ЗАПИСАТЬ()` созданного объекта. Рассмотрим некоторые свойства создаваемого обсуждения.



Свойство **ИДЕНТИФИКАТОР** необходимо для однозначной идентификации обсуждения. В частности, для создания сообщения (методом **СоздатьСообщение()**) необходимо знать, в какое обсуждение это сообщение попадет. Идентификация обсуждения выполняется с помощью свойства **ИДЕНТИФИКАТОР**.

Свойство **Ключ** позволяет использовать для идентификации обсуждения любой уникальный идентификатор, который, например, можно предусмотреть заранее. Так, если обсуждение используется для коммуникации между сервером и клиентом, то для идентификации требуемого обсуждения можно создать ключ обсуждения на этапе разработки прикладного решения, и потом везде использовать этот ключ для получения обсуждения. Ключ обсуждения нельзя изменить для уже записанного обсуждения.

Существенное отличие между свойствами **ИДЕНТИФИКАТОР** и **Ключ** заключается в том, что **ИДЕНТИФИКАТОР** становится известен в момент записи обсуждения, а **Ключ** можно определить заранее, до записи обсуждения.

Управление доступностью обсуждения для интерактивной работы с ним выполняется с помощью свойства **ОТОБРАЖАЕМОЕ**. Если данное свойство установлено в значение **ИСТИНА**, то такое обсуждение будет доступно в основной форме обсуждений и пользователи смогут работать с этим обсуждением. Соответственно, если свойство установлено в значение **ЛОЖЬ**, то такое обсуждение будет недоступно для интерактивной работы и доступ к такому обсуждению будет возможен только через программный интерфейс. Контекстное обсуждение следует всегда создавать так, чтобы свойство **ОТОБРАЖАЕМОЕ** было установлено в значение **ИСТИНА**.

Свойство **КОНТЕКСТОВСЖУДЕНИЯ** управляет тем, какой вид обсуждения будет создаваться: контекстное или неконтекстное. Данное свойство доступно для записи только для нового обсуждения. После записи сменить тип обсуждения уже невозможно. Также надо помнить, что у объекта может быть только одно контекстное обсуждение.

Когда создается неконтекстное обсуждение, свойство **ГРУППОВОЕ** позволяет указать, какое количество участников будет у обсуждения. Если свойство **ГРУППОВОЕ** установлено в значение **ИСТИНА**, то у обсуждения может быть произвольное количество участников. Если это свойство установлено в значение **ЛОЖЬ**, то участников всегда два: автор обсуждения и один его собеседник.

С помощью свойства **УЧАСТНИКИ** задается перечень пользователей, которые будут иметь доступ к созданному неконтекстному(!) обсуждению. Если в качестве списка участников будет задано значение **СИСТЕМАВЗАИМОДЕЙСТВИЯ.СТАНДАРТНЫЕПОЛЬЗОВАТЕЛИ.ВСЕПОЛЬЗОВАТЕЛИПРИЛОЖЕНИЯ**, то к данному обсуждению будут иметь доступ все пользователи. Если состав участников не задан для неконтекстного обсуждения, то оно будет недоступно для всех пользователей системы. Доступ к контекстному обсуждению регулируется правом **ЧТЕНИЕ** на объект контекста (свойство **КОНТЕКСТОВСЖУДЕНИЯ**). При создании контекстного обсуждения значение свойства **УЧАСТНИКИ** устанавливать не следует. При создании обсуждения «один на один»:

- не допускается использование значения **СИСТЕМАВЗАИМОДЕЙСТВИЯ.СТАНДАРТНЫЕПОЛЬЗОВАТЕЛИ.ВСЕПОЛЬЗОВАТЕЛИПРИЛОЖЕНИЯ** в списке участников;
- размер списка участников должен быть равен 2.

Как уже было сказано выше, для того, чтобы создать сообщение - необходимо знать идентификатор обсуждения, в которое данное сообщение попадет. Метод **СоздатьСообщение()** формирует объект, который необходимо заполнить данными и записать. Для созданного сообщения свойства **ДАТА** и **АВТОР** заполняются автоматически. Для изменения этих свойств требуется переход в привилегированный режим. После записи новое сообщение будет «отправлено» участникам. Более подробно рассмотрим свойства сообщения.

Свойство **ИДЕНТИФИКАТОР** является уникальным идентификатором созданного сообщения. Позволяет уникально идентифицировать каждое сообщение в системе.

Свойство **ОВСЖУДЕНИЕ** прикрепляет сообщение к какому-либо существующему обсуждению. Невозможно создать сообщение, которое не связано с каким-либо обсуждением.

Свойство **ДААННЕ** позволяет прикрепить к сообщению произвольные данные. У этих данных имеется два ограничения: прикрепляемые данные должны поддерживать XDTO-сериализацию и объем данных в сериализованном виде не может превышать 100 Кбайт. Это свойство недоступно из пользовательского интерфейса. Данное свойство имеет смысл использовать в том случае, когда обсуждение используется как технологическое, для взаимодействия различных подсистем прикладного решения.

С помощью свойства **ПОЛУЧАТЕЛИ** предоставляется возможность задавать, какие пользователи получают оповещение о сообщении для отображаемого контекстного сообщения. Это свойство не определяет доступность сообщения!

Если сообщение отправляется в отображаемое обсуждение, то с помощью свойства **ДЕЙСТВИЯ** можно задать перечень «команд», которые будут отображаться в форме сообщения. Выбор таких команд будет приводить к переходу по гиперссылке или к вызову соответствующего клиентского обработчика. С помощью этой возможности можно, например, реализовывать фиксированный список ответов робота в специальном обсуждении.

Следует помнить, что во всех случаях, когда требуется указание ссылки на обсуждение (при создании сообщения, при подключении какого-либо обработчика и т. д.), это обсуждение должно существовать к моменту указания ссылки. При

этом не важно, каким образом будет выполняться идентификация обсуждения. Важно, что моменту выполнения требуемого действия, обсуждение, на которое требуется ссылка, уже существовало в базе данных системы взаимодействия.

### 30.4.4.2. ПОИСК ОБСУЖДЕНИЙ И СООБЩЕНИЙ

В процессе работы может возникать необходимость поиска (или получения) какого-либо конкретного обсуждения. Например, через обсуждение с известным ключом выполняется передача данных о длительном процессе, выполняемом на сервере. Получение конкретного обсуждения может выполняться несколькими способами:

1. По идентификатору обсуждения (метод `ПолучитьОбсуждение ( )`). В этом случае получается конкретное обсуждение.
2. По ключу обсуждения (другой вариант метода `ПолучитьОбсуждение ( )`). В этом случае также получается конкретное обсуждение.
3. С помощью отбора (метод `ПолучитьОбсуждения ( )`). В этом случае получается список обсуждений, которые соответствуют критериям отбора. Результат выполнения метода может включать и несколько обсуждений и их пустой список.
4. Получение контекстного обсуждения какого-либо объекта. Для этого используется метод `НачатьПолучениеОбсуждения ( )` формы клиентского приложения.

Также система предоставляет возможность получения списка сообщений по некоторым критериям. Для этого предназначен метод `ПолучитьСообщения ( )`. Результатом выполнения метода выступает массив сообщений, которые удовлетворяют указанным (в параметре отбора) критериям.

### 30.4.4.3. РАСШИРЕННЫЕ ВОЗМОЖНОСТИ

#### 30.4.4.3.1. ОБЩАЯ ИНФОРМАЦИЯ

Прикладное решение может включать возможности автоматического ответа в некоторые обсуждения или может требовать реакции на данные, которые поступают с сервера «1С:Предприятия». Для того чтобы реализовать это, нужно иметь возможность реагировать на изменения, которые происходят в обсуждениях. Тогда прикладное решение может оперативно обрабатывать новые сообщения в обсуждениях и адекватно на них реагировать.

Такое поведение можно условно разделить на две больших класса задач:

- реализация автоматической реакции на не интерактивные сообщения;
- реализация автоматической реакции на интерактивные сообщения пользователя.

Реакция на сообщения реализуется путем подписки клиентского приложения на появление новых сообщений в каком-либо обсуждении. В данный класс задач могут попадать следующие задачи: оповещения клиентского приложения о выполнении некоторых служебных действий (например, перезапуск клиентского приложения), отображение состояния выполнения длительных задач, выполняемых на сервере и т. д. Подключение обработчика новых сообщений выполняется с помощью метода `НачатьПодключениеОбработчикаНовыхСообщений ( )`. Для выполнения этой операции необходимо знать, каким образом идентифицировать обсуждение, на которое подписывается клиентское приложение. Вопрос однозначной идентификации обсуждения (и сообщения, на которое требуется среагировать) очень сильно зависит от специфики реализуемого алгоритма. Перед тем, как выбрать конкретную схему реализации, рекомендуется ответить на несколько очень простых вопросов, которые позволят более четко понять, каким образом будет выполняться взаимодействие. Рассмотрим эти вопросы подробнее:

1. Что мы хотим сделать? Ответ на этот вопрос позволит определить, каким образом будет передаваться указание на выполняемое действие. Действие можно передать с помощью следующих свойств объекта `СообщениеСистемыВзаимодействия`: `Данные`, `Текст` или с помощью свойства `ОбсуждениеСистемыВзаимодействия.Ключ`.
2. Кто будет выполняться требуемое действие? Ответ на этот вопрос определяет адресацию выполняемого действия. Адресация может быть реализована следующими способами:
  - свойства `Получатели` или `Данные` объекта `СообщениеСистемыВзаимодействия`;
  - свойства `Участники` или `Ключ` объекта `ОбсуждениеСистемыВзаимодействия`.
3. Каким образом будет параметризоваться выполняемое действие? Выполняемые действия могут быть простыми и однозначными. Простейший пример - необходимость выполнить перезапуск клиентского приложения в момент получения команды на это. В то же время, действия могут требовать некоторых параметров. Например, требуется выполнить перезапуск клиентского приложения в некоторое время или по какому-то условию. В этом случае параметры действия можно передавать с помощью свойств `Данные` и `Текст` объекта `СообщениеСистемыВзаимодействия`.

Рассмотрим несколько подробнее ответы на указанные вопросы.

На способ выдачи команд может существенно влиять выполняемая задача: реализуется обработка сообщений в интерактивном или не интерактивном обсуждении. Так, если рассматривается реакция на интерактивные команды, то логично

предположить, что команды будут вводиться пользователем с помощью текстовых сообщений (свойство `СообщениеСистемыВзаимодействия.Текст`), а для не интерактивного взаимодействия лучше подходят свойства `СообщениеСистемыВзаимодействия.Данные` или `ОбсуждениеСистемыВзаимодействия.Ключ`. При этом работа через свойство сообщения `Данные` выглядит достаточно оптимальным способом передачи данных, т. к. обеспечивает структурированность передаваемой информации и позволяет более формально описать контракт, по которому взаимодействуют отправитель и получатель сообщений.

Очевидно, что программная обработка сообщений подразумевает подписку на появление новых сообщений. Т.к. система не позволяет подписаться на новые сообщения всех обсуждений, то необходимо определить, какое обсуждение следует обрабатывать. В этом смысле ответ на вопрос об адресации является даже более приоритетным, по сравнению с выбором способа передачи команды. Таким образом, определив нужное обсуждение, определен и то, как будет определяться состав обрабатываемых сообщений: автоматически (если используется адресация через свойства обсуждения) или программная (если используется адресация через свойства каждого, конкретного, сообщения).

С точки зрения данных, передаваемых в сообщение, видно, что самым универсальным способом ответить на все заданные вопросы является использование свойства `Данные` каждого сообщения. С точки зрения чисто не интерактивных действий, выглядит вполне разумным такое построение схемы взаимодействия, когда в системе формируется служебное обсуждение (с заранее известным ключом обсуждения), в свойстве сообщения `Данные` указываются все данные, необходимые для выполнения действия: команда, участники команды, параметры команды и т. д. Очевидно, что такой подход не подойдет для интерактивных команд. В этом случае придется решать проблему с передаваемыми данными с помощью разбора текста сообщения.

### 30.4.4.3.2. ПРИМЕРЫ РЕАЛИЗАЦИИ

#### ОБЩАЯ ИНФОРМАЦИЯ

Рассмотрим несколько примеров реализации обработки сообщений на встроенном языке. Следует помнить, что приводятся примеры, демонстрирующие те или иные особенности механизма, а не законченные инструменты решения какой-либо задачи.

#### ПРИМЕР 1

Общая схема подписки на какое-либо обсуждение выглядит достаточно просто. Для этого следует знать только значение, которое однозначно идентифицирует обсуждение. В этой роли выступает идентификатор обсуждения или его (обсуждения) ключ. Таким образом, подписка на обработку новых сообщений существующего обсуждения выглядит следующим образом:

```
Процедура ПодписатьсяНаСлужебноеОбсуждение (КлючСлужебногоОбсуждения) Экспорт
    ОбработкаПодключения = Новый ОписаниеОповещения ("ЗавершениеПодключения", Сообщение-
яКлиент, , "ОшибкаПодключения", СообщенияКлиент);
    ОбработкаСообщенийСервера = Новый ОписаниеОповещения ("ОбработкаСообщенийСервера",
СообщенияКлиент);
    СистемаВзаимодействия.НачатьПодключениеОбработчикаНовыхСообщений (ЗавершениеПодклю-
чения, КлючСлужебногоОбсуждения, ОбработкаСообщенийСервера);
КонецПроцедуры
Процедура ОбработкаСообщенийСервера (Сообщение, ДополнительныеПараметры) Экспорт
    // в данном обработчике будет выполняться обработка новых сообщений
КонецПроцедуры
Процедура ЗавершениеПодключения (ДополнительныеПараметры) Экспорт
    // в данном обработчике будет выполняться обработка завершения подключения обработ-
чика
КонецПроцедуры
Процедура ОшибкаПодключения (ИнфоОбОшибке, СтандартнаяОбработка, ДополнительныеПара-
метры) Экспорт
    // в данном обработчике будет выполняться обработка ошибок подключения обработчика
КонецПроцедуры
```

В данном примере нет разницы (с точки зрения синтаксиса метода) каким образом выполняется идентификация обсуждения. Хотя формальный параметр и называется `КлючСлужебногоОбсуждения`, его фактическим значением может быть и ключ и идентификатор обсуждения.

#### ПРИМЕР 2

Данный пример демонстрирует создание обсуждения, предназначенного для выполнения единственного действия, а также реализация этого действия и отправку команды на выполнение действия.

Будет создано обсуждение, которое предназначено только для отправки команды на перезапуск клиентского приложения для определенного пользователя (или пользователей).

## 1С:СЕРВЕР ВЗАИМОДЕЙСТВИЯ: Краткое руководство по установке и настройке

Команда будет выполнена в том случае, если нужный пользователь указан в сообщении. В качестве идентификатора обсуждения будет выступать ключ.

Создание такого обсуждения будет выполняться на сервере следующим образом:

```
Обсуждение = СистемаВзаимодействия.СоздатьОбсуждение ();
Обсуждение.Отображаемое = Ложь;
Обсуждение.Ключ = "ПерезапускКлиентов";
Обсуждение.Участники.Добавить (СистемаВзаимодействия.СтандартныеПользователи.ВсеПользователиПриложения);
Обсуждение.Записать ();
```

Отправка сообщения в такое обсуждение может выглядеть следующим образом:

```
ИдентификаторОбсуждения = СистемаВзаимодействия.ПолучитьОбсуждение ("ПерезапускКлиентов");
АдресатыКоманды = Новый Массив;
АдресатыКоманды.Добавить (<ИдентификаторПользователя1>);
АдресатыКоманды.Добавить (<ИдентификаторПользователяN>);
Сообщение = СистемаВзаимодействия.СоздатьСообщение (ИдентификаторОбсуждения);
Сообщение.Данные = АдресатыКоманды;
Сообщение.Записать ();
```

Сообщение всегда должно содержать или текст или данные, даже если эти компоненты не анализируются системой.

В обработчике новых сообщений может быть следующий код на встроенном языке:

```
ТекущийПользователь = СистемаВзаимодействия.ИдентификаторТекущегоПользователя ();
Если Сообщение.Данные.Содержит (ТекущийПользователь) Тогда
    ЗавершитьРаботуСистемы (Ложь, Истина);
КонецЕсли;
```

**ПРИМЕЧАНИЕ.** Плохо в этом примере то, что он перезапускает приложение «не глядя» на то, что делает пользователь. В реальной системе так делать не следует.

### ПРИМЕР 3

Еще одним вариантом реализации обработки сообщений будет рассмотрена реализация одного обсуждения, которое предназначено для всего служебного трафика внутри системы. Обсуждение идентифицируется ключом, а «подробности» заказываемого действия находятся в свойстве сообщения **Данные**:

- Свойство **Команда** содержит команду, которая должна быть выполнена;
- Свойство **Адресат** - содержит пользователя, который выполняет команду.

Отправка команды будет следующим образом:

```
ИдентификаторОбсуждения = СистемаВзаимодействия.ПолучитьОбсуждение ("КомандныйКанал");
Параметры = Новый Структура;
Параметры.Вставить ("Команда", "Перезапуск");
Параметры.Вставить ("Адресат", ИдентификаторПолучателя);
Сообщение = СистемаВзаимодействия.СоздатьСообщение (ИдентификаторОбсуждения);
Сообщение.Данные = Параметры;
Сообщение.Записать ();
```

Обработка сообщения будет выглядеть аналогично предыдущему примеру:

```
ТекущийПользователь = СистемаВзаимодействия.ИдентификаторТекущегоПользователя ();
Если Сообщение.Данные.Адресат = ТекущийПользователь Тогда
    Если Сообщение.Данные.Команда = "Перезапуск" Тогда
        ЗавершитьРаботуСистемы (Ложь, Истина);
    КонецЕсли;
КонецЕсли;
```

### 30.4.4.3.3. РЕАЛИЗАЦИЯ РОБОТА

Если требуется реализовать некоторого робота, который будет реагировать на сообщения («бота»), которые интерактивно формирует пользователь, то логика работы будет несколько другой. В этом случае необходимо подписываться не на получение нового сообщения, а на факт отправки сообщения. Для этого существует метод **ПодключитьОбработчикПослеОтправкиСообщения ()**. Подключенный обработчик будет срабатывать каждый раз после того, как пользователь отправит интерактивное сообщение. Срабатывать обработчик будет на том клиентском компьютере, на котором пользователь ввел сообщение. В подключенный обработчик события поступит введенное сообщение и обсуждение, в рамках которого отправлено сообщение.

## 1С:СЕРВЕР ВЗАИМОДЕЙСТВИЯ: Краткое руководство по установке и настройке

Дальнейшая реакция опять зависит от логики прикладного решения. В обработчике можно проанализировать текст сообщения и что-то ответить в том случае, если в этом тексте обнаружены ключевые слова. В обработчике можно проанализировать участников обсуждения и ответить только в том случае, если пользователь, который олицетворяет робота, входит в состав участников и т. д.

```
// === Код на стороне клиентского приложения
Процедура СоздатьРобота () Экспорт
    Обработчик = Новый ОписаниеОповещения ("ОбработкаОтправкиСообщения", СообщенияКлиент);
    СистемаВзаимодействия.ПодключитьОбработчикПослеОтправкиСообщения (Обработчик);
КонецПроцедуры
Процедура ОбработкаОтправкиСообщения (Сообщение, Обсуждение, ДополнительныеПараметры)
Экспорт
    Если СтрНачинаетсяС (Сообщение.Текст, "?") Тогда
        // здесь вызывается реализация робота (бота)
        СоздатьСообщениеРобота (Обсуждение.Идентификатор, "Я робот!");
    КонецЕсли;
КонецПроцедуры
// === Код на стороне серверного приложения
Процедура СоздатьСообщениеРобота (ИдентификаторОбсуждения, ТекстСообщения) Экспорт
    Обсуждение = СистемаВзаимодействия.ПолучитьОбсуждение (ИдентификаторОбсуждения);
    Сообщение = СистемаВзаимодействия.СоздатьСообщение (Обсуждение);
    Сообщение.Текст = ТекстСообщения;
    Сообщение.Записать ();
КонецПроцедуры
```

Вышеуказанный пример подключает обработчик отправки сообщения, в котором выполняется анализ сообщения, введенного пользователем. Если сообщение начинается с символа «?», то считается, что такое сообщение адресовано роботу. В этом случае робот ответит в обсуждение сообщением **Я робот!**.

### 30.4.4.4. УСТАНОВКА ДЕЙСТВИЙ В СООБЩЕНИЕ

При разработке роботов может возникнуть потребность разместить в сообщении некоторые «команды», чтобы пользователь мог быстро выполнить необходимое действие. Для реализации этого в объекте сообщения есть свойство **ДЕЙСТВИЯ**. Данное свойство представляет собой список значений, который содержит информацию о «командах». В качестве значения элемента списка значений указывается значение, которое будет передано в обработчик и позволит идентифицировать выполняемое действие, а в качестве представления - представление этого действия в сообщении. Если в системе не подключен обработчик действия сообщения (метод **ПодключитьОбработчикДействияСообщения ()**), то никакие действия выполнены не будут. Если обработчик подключен, то в качестве действия может выступать любое значение, которое будет проанализировано в обработчике. В случае подключенного обработчика система будет передавать заданные для действий значения в обработчик действия. И уже в обработчике следует реализовать конкретную реализацию того или иного действия.

Для отображения гиперссылки (с возможностью перехода) не требуется использовать действия, достаточно просто разместить гиперссылку в тексте сообщения.

### 30.4.4.5. ВЛОЖЕНИЯ

Система взаимодействия позволяет присоединять вложения к сообщениям системы взаимодействия. Эта возможность доступна только в том случае, если сервер системы взаимодействия соответствующим образом настроен. Для того, чтобы определить возможность работы с вложениями, следует использовать метод **ПоддерживаетсяРаботаСВложениями ()**.

Собственно работа с вложениями осуществляется с помощью свойства **Вложения** объекта **СообщениеСистемыВзаимодействия**. При работе с вложениями сервер системы взаимодействия контролирует три параметра:

- размер одного загружаемого файла;
- размер всех загружаемых файлов, за сутки;
- размер всех получаемых файлов, за сутки.

Данные параметры настраиваются на стороне сервера взаимодействия. При нарушении какого-либо из этих параметров формируется исключение.

Чтобы добавить новый объект вложения, следует использовать метод **Вложения.Добавить ()**. При этом предполагается, что объект, помещаемый во вложение, не обязательно является файлом на диске. Можно, например, поступить следующим образом для отправки печатной формы документа в формате PDF:

- Создать поток в памяти (объект **ПотокВПамяти**).



- Сохранить табличный документ в созданный поток в формате PDF (метод `Записать ( )` объекта `ТабличныйДокумент`).
- Добавить вложение из созданного потока.

При создании вложения имеет смысл придерживаться следующих рекомендаций:

- Если вложение предполагается использовать в интерактивных сообщениях, то рекомендуется в качестве значения свойства `ВложениеСистемыВзаимодействия . Наименование` указывать имя файла с расширением, соответствующим фактическому типу содержимого. При этом свойство `ВложениеСистемыВзаимодействия . ТипСодержимого` должно содержать реальный MIME-тип вложения.
- Если не предполагается использовать вложение для интерактивных операций, то значения свойств `Наименование` и `ТипСодержимого` объекта `ВложениеСистемыВзаимодействия` формально могут быть любыми, но рекомендуется в качестве значения свойства `ТипСодержимого` указывать реальный MIME-тип, максимально корректно описывающий тип передаваемых данных.

Если свойство `ВложениеСистемыВзаимодействия . ТипСодержимого` не задано, то платформа пытается определить тип вложения самостоятельно:

- При интерактивном прикреплении файла в сообщение:
- В тонком и толстом клиентах тип определяется по расширению файла.
- В веб-клиенте - на основании информации, предоставляемой веб-браузером.
- При вставке картинки из буфера обмена устанавливается тип `IMAGE/PNG`.
- Если объект `ВложениеСистемыВзаимодействия` формируется из встроенного языка и его свойство `ТипСодержимого` не указывается, то считается, что свойство объекта `Наименование` содержит имя прикрепляемого файла и выполняется попытка определить тип на основании расширения этого файла.

При отображении сообщения с вложениями в интерфейсе клиентского приложения, платформа отображает известные вложения специальными картинками:

- Для файлов стандартных платформенных типов (текстовый документ, табличный документ, графическая схема, географическая схема, HTML-документ, внешний отчет, внешняя обработка) отображается стандартная платформенная картинка соответствующего типа файла.
- Для файлов Microsoft Office, Open Office, Libre Office, PDF-файлов и архивов отображается картинка соответствующего типа файла.
- Для вложения с типом содержимого `audio/*` отображается картинка «аудио».
- Для вложения с типом содержимого `video/*` отображается картинка «видео».
- Для вложения с типом содержимого `text/*` отображается картинка «текст».
- Для вложения с типом содержимого `image/*` отображается картинка «картинка».
- Для остальных типов содержимого отображается картинка «неизвестный тип файла».

---

### 30.4.4.6. ДЕЙСТВИЯ ДЛЯ ВЛОЖЕНИЙ

Если в системе взаимодействия настроена работа с вложениями, то в прикладном решении может потребоваться реализовать специальные действия для вложений. Для этой цели предназначен метод `ПодключитьОбработчикФормированияКоманд ( )`.

Данный обработчик срабатывает каждый раз, когда в интерфейсе клиентского приложения требуется получить команду по умолчанию (при нажатии левой кнопки мыши) или контекстное меню (при нажатии правой кнопки мыши). При нажатии правой кнопки мыши обработчик формирования команд будет вызван при нахождении курсора мыши над ссылкой, вложением, действием или самим сообщением. При нажатии левой кнопки мыши список команд будет затребован только для ссылки, вложения или действия. Так как обработчик формирования команд вызывается в интерактивном режиме и достаточно часто, в нем не рекомендуется размещать какие-либо длительные и интерактивные операции (например, задавать вопросы или открывать формы).

Обработчик формирования команд получает на вход следующие параметры:

- `Параметры` - содержит информацию о том, где в сообщении нажата кнопка мыши. Свойство `Параметры . Источник` позволит понять, где нажата кнопка мыши. Свойства `Обсуждение` и `Сообщение` задают «контекст» действия.
- Параметр `Команды` содержит массив объектов типа `ОписаниеКомандыСистемыВзаимодействия`. При вызове обработчика этот параметр содержит команды, которые платформа по умолчанию размещает в контекстном меню для данного источника. Разработчик может сформировать это меню полностью самостоятельно, удалив команды, созданные по умолчанию. Стоит отметить, что конструкция встроенного языка `Команды . Добавить (Новый ОписаниеКомандыСистемыВзаимодействия (Неопределено) )`; добавит в контекстное меню разделитель. Предоставляется возможность добавить стандартные команды системы взаимодействия. В этом случае необходимо добавление команды выглядит следующим образом `Команды . Добавить (Новый`

;

- Параметр **КомандаПоУмолчанию** содержит описание команды, которая будет вызвана, если в сообщении нажали левую кнопку мыши. Если данный параметр равен значению **Неопределено**, то контекстное меню будет открыто в любом случае: и при нажатии на правую и на левую кнопки мыши.
- Параметр **ДополнительныеПараметры** содержит данные, которые может потребоваться передать в обработчик формирования команд из того контекста, в котором подключается этот обработчик.

При необходимости, обработчик формирования команд может быть отключен с помощью метода **ОтключитьОбработчикФормированияКоманд()**.

### 30.4.5. СОВМЕСТНОЕ ИСПОЛЬЗОВАНИЕ

Система взаимодействия позволяет объединить несколько приложений, зарегистрированных в системе взаимодействия, в единое пространство. В случае объединения приложений, в качестве пользователей системы взаимодействия выступают пользователи всех объединенных приложений. При объединении приложений, также, становится доступной возможность обмена сообщениями между приложениями. Объединять приложения может только владелец абонента всех объединяемых приложений. Совместное использование поддерживается:

- Сервисом **1cDialog.com**,
- Продуктом **1С:Сервер взаимодействия**.

Для объединения приложений можно использовать стандартную функцию системы «1С:Предприятие» (при использовании сервиса **1cDialog.com**) или реализовать нужный код на встроенном языке (при использовании экземпляра продукта **1С:Сервер взаимодействия**).

В качестве примера необходимости такого объединения можно привести работу территориально-распределенного предприятия, например сети магазинов и центральный офис сети. Каждый магазин и центральный офис имеют собственную информационную базу. В центральном офисе можно развернуть экземпляр продукта **1С:Сервер взаимодействия** или воспользоваться сервисом **1cDialog.com**. Базы данных регистрируются в выбранном сервисе (каждый магазин и база центрального офиса). Затем происходит объединение получившихся приложений. В результате получается единое информационное пространство, позволяющее обмениваться сообщениями пользователям всех баз данных в реальном масштабе времени.

Для объединения приложений необходимо использовать объект **СовместноеИспользованиеПриложенийСистемыВзаимодействия**. Перечень объединяемых приложений задается с помощью свойства этого объекта **Приложения**. Данное свойство можно заполнить на основании результата использования функции **ПолучитьПриложенияАбонента()**. Данный метод следует вызывать при работе от имени пользователя, являющегося владельцем абонента.

Кроме списка объединяемых приложений, объект **СовместноеИспользованиеПриложенийСистемыВзаимодействия** позволяет указать, каким образом будут сопоставляться пользователи системы взаимодействия объединяемых приложений:

- По имени (значение **Имя**) - в этом случае одинаковыми будут считаться пользователи системы взаимодействия, у которых одинаковое свойство **Имя**.
- По полному имени (значение **ПолноеИмя**) - в этом случае одинаковыми будут считаться пользователи системы взаимодействия, у которых одинаковое свойство **ПолноеИмя**.
- По ключу сопоставления (значение **КлючСопоставления**) - в этом случае одинаковыми будут считаться пользователи системы взаимодействия, у которых одинаковое свойство **КлючСопоставления**. Таким образом, можно выполнить выборочное сопоставление пользователей приложений. Для тех пользователей, которых необходимо сопоставить между разными приложениями, следует указать свойство **КлючСопоставления**. В этом случае пользователи с одинаковыми ключами будут прозрачны для объединенных приложений, а остальные пользователи будут доступны только в рамках «своих» приложений системы взаимодействия.

Выбор способа сопоставления пользователей осуществляется с помощью свойства **СопоставлениеПользователей**.

Наконец свойство **СопоставлениеКонтекстовОбсуждений** объекта **СовместноеИспользованиеПриложенийСистемыВзаимодействия** позволяет указать, что объекты, которые в разных приложениях обладают одинаковыми идентификаторами - будут считаться одинаковыми в рамках информационного пространства объединенных приложений. Так, если в рамках распределенной информационной системы установлено сопоставление контекстов обсуждений, то открывается возможность обсуждать один документ из разных приложений (информационных баз).

При объединении нескольких приложений, эти приложения начинают работать по схеме «все со всеми». Т.е. каждое из объединенных приложений получает возможность взаимодействовать с любым другим приложением из списка объединенных. В тоже время, система взаимодействия не поддерживает использование такой схемы: **Приложение1** объединяется

с **Приложение2**, а **Приложение2** объединяется с **Приложение3** и при этом **Приложение3** не может взаимодействовать с **Приложение1**.

Для отмены сопоставления приложений служит метод **ОтменитьСовместноеИспользованиеПриложенийАбонента()**. Отмена совместного использования выполняется аналогично установке сопоставления: формируется список приложений, которые больше нет необходимости использовать совместно (но не меньше двух), и вызывается метод **ОтменитьСовместноеИспользованиеПриложенийАбонента()**.

С помощью метода **СовместноеИспользованиеПриложенийВзаимодействия()** можно получить список совместно используемых приложений и параметров совместного использования. При этом следует учитывать тот факт, что совместно используемые приложения возвращаются парами.

### 30.5. АДМИНИСТРИРОВАНИЕ

Описание администрирования системы взаимодействия см. [здесь](#).

## ПРИЛОЖЕНИЕ 3. РУКОВОДСТВА АДМИНИСТРАТОРА - ОПИСАНИЕ И РАСПОЛОЖЕНИЕ СЛУЖЕБНЫХ ФАЙЛОВ

### 3.19. INETCFG.XML

Файл **inetcfg.xml** позволяет задавать настройки прокси по умолчанию и имеет больший приоритет над настройками прокси по умолчанию в ОС Windows или параметрами, указанными в переменных окружения в ОС Linux и macOS.

Файл располагается в каталоге конфигурационных файлов системы «1С:Предприятие», и его наличие не является обязательным.

- Для ОС Windows:
  - При наличии файла - настройки берутся из файла.
  - При отсутствии файла - настройки берутся из настроек Microsoft Internet Explorer. Не поддерживается использование сценария автоматической настройки.
- Для ОС Linux:
  - При наличии файла - настройки берутся из файла.
  - При отсутствии файла - происходит попытка получения настроек из переменных окружения **http\_proxy**, **https\_proxy**, **ftp\_proxy**, **https\_proxy** (в соответствии с используемым протоколом). Если эти переменные среды не указаны, производится попытка получить настройки прокси из переменной окружения **all\_proxy**. Если в переменных окружения указаны некорректные параметры - они игнорируются.
- Для ОС macOS:
  - При наличии файла - настройки берутся из файла.
  - При отсутствии файла - происходит попытка получения настроек из переменных окружения **http\_proxy**, **https\_proxy**, **ftp\_proxy**, **https\_proxy** (в соответствии с используемым протоколом). Если эти переменные среды не указаны, производится попытка получить настройки прокси из переменной окружения **all\_proxy**. Если в переменных окружения указаны некорректные параметры - они игнорируются.

При настройке прокси можно использовать информацию **User-Agent** из HTTP-запроса:

- тонкий клиент - **1CV8C**;
- Web-сервис - **1C+Enterprise/8.3**;
- веб-клиент - данный параметр формирует веб-браузер.

*Пример:*

```
<InternetProxy
  protocols="http=10.1.0.8:8080 10.1.0.9:8080"
  user="proxyUser"
  password="proxyPassword"
  bypassOnLocal="true"
  bypassOnAddresses="127.0.0.1 *.master"
/>
```

Корневой элемент **INTERNETPROXY**, задающий настройки прокси по умолчанию, имеет рассмотренную ниже структуру (атрибуты).



### АТРИБУТ NTLM НЕОБЯЗАТЕЛЬНЫЙ

Тип: **БУЛЕВО**. Признак использования механизма NTLM-аутентификации:

- **TRUE** - NTLM-аутентификация включена;
- **FALSE** - выключена.

По умолчанию NTLM-аутентификация включена.

### АТРИБУТ PROTOCOLS НЕОБЯЗАТЕЛЬНЫЙ

Тип: **СТРОКА**. Задаёт имя и порт хоста для протоколов. Имеет следующий формат:

```
ПараметрыПроксиПротокола1ПараметрыПроксиПротокола2
...ПараметрыПроксиПротоколаN
ПараметрыПроксиПротокола := [Протокол] "="хост": "порт
```

Список параметров прокси-протоколов разделён пробелами. Каждый параметр состоит из необязательного имени протокола, знака "=", имени хоста и порта прокси-сервера, разделённых двоеточием. Если имя протокола не указано, то параметры прокси используются для всех протоколов, для которых они явно не указаны. Протоколы могут иметь следующие имена:

- HTTP,
- HTTPS,
- FTP.

Регистр является значимым, другие имена протоколов не поддерживаются, например:

```
protocols="http=10.1.0.8:8080 10.1.0.9:8080"
```

Где:

- для протокола HTTP определены параметры прокси: хост - 10.1.0.8, порт - 8080;
- для остальных протоколов (HTTPS, FTP): хост - 10.1.0.9, порт - 8080.

### АТРИБУТ USER НЕОБЯЗАТЕЛЬНЫЙ

Тип: **СТРОКА**. Имя пользователя для аутентификации на прокси-сервере, например:

```
user="proxyUser"
```

### АТРИБУТ PASSWORD НЕОБЯЗАТЕЛЬНЫЙ

Тип: **СТРОКА**. Пароль пользователя для аутентификации на прокси-сервере. Пример:

```
password="proxyPassword"
```

### АТРИБУТ BYPASSONLOCAL НЕОБЯЗАТЕЛЬНЫЙ

Тип: **БУЛЕВО**. Признак того, использовать ли прокси-сервер для локальных адресов:

- **TRUE** - не использовать;
- **FALSE** - использовать.

Локальность адреса определяется по наличию точки в DNS имени адреса (т. е. все IP-адреса не являются локальными). Поэтому может получиться так, что фактически локальный адрес не распознаётся как локальный.

Например, <пользователь>.<домен> является в Windows XP локальным адресом, но не распознаётся как локальный. Для того чтобы запретить использовать прокси для адресов, которые воспринимаются как локальные, используется следующий параметр:

```
bypassOnLocal="true"
```

Для всех остальных адресов нужно использовать параметр **BYASSONADDRESSES**.

### АТРИБУТ BYPASSONADDRESSES НЕОБЯЗАТЕЛЬНЫЙ

Тип: **Строка**. Список адресов, для которых прокси не используется. Имеет следующий формат:

```
хост1 хост2 ... хостN
```

Имена хостов разделяются пробелами. Имя хоста может содержать специальные символы маски: \* - любое количество символов, ? - любой символ. Например, чтобы заблокировать прокси для всех хостов домена, нужно использовать: \*.<имя домена>, например:

```
bypassOnAddresses="127.0.0.1 *.master"
```

В примере для адреса 127.0.0.1 (**localhost**) и для всех адресов домена **master** прокси не используется.

### ПРИЛОЖЕНИЕ 4. РУКОВОДСТВА РАЗРАБОТЧИКА - ОПИСАНИЕ ПРАВ ДОСТУПА

Данное приложение содержит описание всех прав доступа, которыми можно управлять при редактировании ролей в «1С:Предприятии». Символом "\*" помечены права доступа, которые поддерживаются в мобильной платформе.

- **Automation (Automation)** - использование automation-сервера;
- **Администрирование (Administration)**:
  - Административные действия над всей информационной базой и редактирование списка пользователей (право для конфигурации). Требуется наличия права **АдминистрированиеДанных**.
  - Возможность настройки параметров соединения (для внешнего источника данных).
  - Отображение списка лицензий, использовавшихся при работе с конфигурацией и информационной базой, в окне **О программе**.
  - Удаление областей данных (включая удаление всех областей сразу).
  - Возможность выполнения фонового обновления конфигурации базы данных на стороне клиента.
  - Возможность устанавливать имя профиля безопасности в диалогах управления расширениями.
  - Более подробная информация при просмотре журнала регистрации (см. [здесь](#)).
  - Управление копиями базы данных (см. [здесь](#)).
- **АдминистрированиеДанных (DataAdministration)** - данное право регулирует административные действия над данными. К этим действиям относятся:
  - Пользователь получает возможность просмотра следующих событий журнала регистрации: **Аутентификация**, **Ошибка аутентификации** (группы **Сеанс**), **Доступ**, **Отказ в доступе** (группы **Доступ**), **Подтверждено**, **Отклонено** (группы **Провайдер OpenID**).
  - Пользователь получает возможность получать значения отборов событий без ограничений.
  - Однако для просмотра журнала регистрации требуется наличие права **ЖурналРегистрации**.
  - Установка часового пояса информационной базы (области данных) - вызов метода **УСТАНОВИТЬЧАСОВОЙПОЯСИНФОРМАЦИОННОЙБАЗЫ()**.
  - Создание начального образа подчиненного узла распределенной информационной базы (вызов метода **СОЗДАТЬНАЧАЛЬНЫЙОБРАЗ()**).
  - Обновление нумерации объектов (вызов метода **ОБНОВИТЬНУМЕРАЦИЮОБЪЕКТОВ()**).
  - Выполнение методов объекта **СТАНДАРТНОЕХРАНИЛИЩЕНАСТРОЕКМЕНЕДЖЕР** в тех случаях, если выполняется работа с настройками пользователя, отличными от текущего.
  - Выполнение методов **ПолучитьДанныеРегистрацииИнформационнойБазы()** / **УСТАНОВИТЬДАННЫЕРЕГИСТРАЦИИИнформационнойБазы()**.
  - Управление составом таблиц, входящих в копию базы данных (см. [здесь](#)).
- **АдминистрированиеРасширенийКонфигурации (ConfigurationExtensionsAdministration)** - управляет доступом к интерфейсу управления расширениями (как интерактивному, так и программному) в режиме «1С:Предприятие». В безопасном режиме администрирование расширений невозможно.
- **АктивныеПользователи (ActiveUsers)** - просмотр списка активных пользователей.
- **ВводПоСтроке\* (InputByString)** - использование режима ввода по строке.
- **ВебКлиент (WebClient)** - право запуска веб-клиента.
- **ВнешнееСоединение (ExternalConnection)** - внешнее соединение.
- **Вывод\* (Output)** - вывод на печать, запись и копирование в буфер обмена.
- **Выполнение (Execute)** - выполнение задачи.
- **Добавление (Insert)** - добавление (основное).
- **ЖурналРегистрации (EventLog)** - право просмотра журнала регистрации. Полнота отображения журнала регистрации управляется правами **Администрирование** и **АдминистрированиеДанных**.
- **Изменение (Update)** - изменение (основное).
- **ИзменениеСтандартнойАутентификации (StandardAuthenticationChange)** - пользователь имеет право изменять свои сохраненные параметры стандартной аутентификации (свойства **Аутентификация стандартная**, **Имя пользователя**, **Пароль**) внешнего источника данных.

- **ИзменениеАутентификацииОССеанса** (`SessionOSAuthenticationChange`) - пользователь имеет право изменять параметры аутентификации ОС внешнего источника данных в своих настройках и в параметрах сеанса.
- **ИзменениеИсторииДанных** (`UpdateDataHistory`) - предоставляет возможность изменять историю данных (создавать новую версию или удалять существующую версию).
- **ИзменениеИсторииДанныхОтсутствующихДанных** (`UpdateMissingDataHistory`) - предоставляет возможность изменять историю данных (создавать новую версию или удалять существующую версию) для отсутствующих данных. Отсутствующие данные - это удаленные данные или данные, которые недоступны благодаря ограничениям доступа к данным.
- **ИзменениеНастроекИсторииДанных** (`UpdateDataHistorySettings`) - позволяет изменять настройки истории данных.
- **ИзменениеКомментарияВерсииИсторииДанных** (`UpdateDataHistoryVersionComment`) - предоставляет возможность изменять комментарий к версии данных.
- **ИзменениеСтандартнойАутентификацииСеанса** (`SessionStandardAuthenticationChange`) - пользователь имеет право изменять параметры стандартной аутентификации (свойства **Аутентификация стандартная**, **Имя пользователя**, **Пароль**) внешнего источника данных для текущего сеанса.
- **ИнтерактивнаяАктивация** (`InteractiveActivate`) - интерактивная активация.
- **ИнтерактивнаяОтменаПроведения\*** (`InteractiveUndoPosting`) - интерактивная отмена проведения.
- **ИнтерактивнаяПометкаУдаления\*** (`InteractiveSetDeletionMark`) - интерактивная пометка на удаление.
- **ИнтерактивнаяПометкаУдаленияПредопределенныхДанных\*** (`InteractiveSetDeletionMarkPredefinedData`) - разрешает интерактивную пометку удаления предопределенных данных.
- **ИнтерактивноеВыполнение** (`InteractiveExecute`) - интерактивное выполнение задачи.
- **ИнтерактивноеДобавление\*** (`InteractiveInsert`) - интерактивное добавление.
- **ИнтерактивноеИзменениеПроведенных\*** (`InteractiveChangePosted`) - интерактивное редактирование проведенного документа. Если право не установлено, то пользователь не может удалить проведенный документ, установить пометку удаления, перепровести или сделать непроведенным. Форма такого документа открывается в режиме просмотра.
- **ИнтерактивноеОткрытиеВнешнихОбработок** (`InteractiveOpenExtDataProcessors`) - интерактивное открытие внешних обработок.
- **ИнтерактивноеОткрытиеВнешнихОтчетов** (`InteractiveOpenExtDataReports`) - интерактивное открытие внешних отчетов.
- **ИнтерактивноеПроведение\*** (`InteractivePosting`) - интерактивное проведение.
- **ИнтерактивноеПроведениеНеОперативное** (`InteractivePostingRegular`) - интерактивное проведение (стандартными командами форм) документа в неоперативном режиме.
- **ИнтерактивноеСнятиеПометкиУдаления\*** (`InteractiveClearDeletionMark`) - интерактивное снятие пометки на удаление.
- **ИнтерактивноеСнятиеПометкиУдаленияПредопределенныхДанных\*** (`InteractiveClearDeletionMarkPredefinedData`) - разрешает интерактивно снимать пометку удаления с предопределенных данных.
- **ИнтерактивноеУдаление** (`InteractiveDelete`) - интерактивное непосредственное удаление.
- **ИнтерактивноеУдалениеПомеченных** (`InteractiveDeleteMarked`) - интерактивное удаление помеченных объектов.
- **ИнтерактивноеУдалениеПомеченныхПредопределенныхДанных\*** (`InteractiveDeleteMarkedPredefinedData`) - разрешает интерактивное удаление помеченных предопределенных данных.
- **ИнтерактивноеУдалениеПредопределенныхДанных** (`InteractiveDeletePredefinedData`) - разрешает интерактивное удаление предопределенных данных.
- **ИнтерактивныйСтарт** (`InteractiveStart`) - интерактивный старт бизнес-процесса.
- **Использование** (`Use`) - использование того или иного объекта конфигурации, для функции внешнего источника данных - возможность использовать из встроенного языка и языка запросов.
- **МонопольныйРежим** (`ExclusiveMode`) - использование монопольного режима.
- **ОбновлениеКонфигурацииБазыДанных** (`UpdateDataBaseConfiguration`) - позволяет выполнять обновление конфигурации базы данных. Для выполнения обновления конфигурации на сервере или фоновому обновлению конфигурации на сервере достаточно наличия только этого права, право **Администрирование** в этом случае не требуется.

- **ОтменаПроведения** (`UndoPosting`) - отмена проведения документов.
- **ПереходНаВерсиюИсторииДанных** - предоставляет интерактивное право восстановить состояние версионизируемых реквизитов объекта на момент времени, соответствующий выбранной версии. Отсутствие данного права делает недоступной команду **Перейти на версию** в форме истории изменений объекта.
- **Получение** (`Get`) - получение значения, не хранящегося в базе данных.
- **Проведение** (`Posting`) - проведение документов.
- **Просмотр\*** (`View`) - просмотр. В диалоге **Выбор типа** будут недоступны те типы, для которых у текущего пользователя отсутствует это право. Для функции внешнего источника данных - возможность использовать функцию в списке полей системы компоновки данных.
- **ПросмотрИсторииДанных** (`ViewDataHistory`) - предоставляет интерактивное право просмотра истории данных. При отсутствии данного права становится недоступна команда **История изменений** формы объекта и формы списка.
- **РегистрацияИнформационнойБазыСистемыВзаимодействия** (`CollaborationSystemInfoBaseRegistration`) - пользователь, обладающий данным правом, имеет возможность выполнять регистрацию информационной базы в сервисе 1С:Диалог.
- **Редактирование\*** (`Edit`) - редактирование.
- **РедактированиеКомментарияВерсииИсторииДанных** (`EditDataHistoryVersionComment`) - предоставляет интерактивное право изменения комментария версии данных. Отсутствие данного права не позволит редактировать комментарий версии в списке версии истории данных.
- **РежимВсеФункции\*** (`AllFunctionsMode`) - управляет доступностью в меню **Сервис и настройки** (главном меню) команды **Все функции** и соответствующего пункта настроек клиентского приложения.
- **РежимОсновногоОкнаВстроенноеРабочееМесто** (`MainWindowModeEmbeddedWorkplace`) - право на запуск основного окна клиентского приложения в режиме встроенного рабочего места.
- **РежимОсновногоОкнаКиоск** (`MainWindowModeKiosk`) - право на запуск основного окна клиентского приложения в режиме киоска.
- **РежимОсновногоОкнаОбычный** (`MainWindowModeNormal`) - право на запуск в обычном режиме основного окна клиентского приложения.
- **РежимОсновногоОкнаПолноэкранноеРабочееМесто** (`MainWindowModeFullscreenWorkplace`) - право на запуск основного окна клиентского приложения в режиме полноэкранного рабочего места.
- **РежимОсновногоОкнаРабочееМесто** (`MainWindowModeWorkplace`) - право на запуск основного окна клиентского приложения в режиме рабочего места.
- **СохранениеДанныхПользователя** (`SaveUserData`) - управляет возможностью сохранения настроек пользователя. Если право выключено, то:
  - Недоступна кнопка **История**, обращение к истории из встроенного языка вызывает исключение, интерактивные операции записи не фиксируются в истории.
  - В панели системных команд недоступна команда вызова списка избранного. В формах **Получение ссылки** и **Переход по ссылке** недоступна кнопка **Добавить в избранное**. При обращении к избранному из встроенного языка вызывается исключение.
  - Недоступна настройка форм (отсутствует команда **Изменить форму**).
  - Недоступны команды настройки панели разделов, панели навигации и начальной страницы.
  - В отчетах недоступны команды сохранения пользовательских настроек и вариантов отчета. Не выдается предложение запомнить изменения, выполненные пользователем в текущем варианте отчета. Изменения не сохраняются. Команда выбора настроек отчета доступна только в том случае, если у отчета или конфигурации установлено хранилище пользовательских настроек отчета. При вызове метода **УСТАНОВИТЬТЕКУЩИЕПОЛЬЗОВАТЕЛЬСКИЕНАСТРОЙКИ()** выдается исключение.
  - Недоступны команды сохранения и восстановления данных форм (команды **Сохранить параметры** и **Восстановить параметры**). Не выполняется автоматическое сохранение данных форм.
  - Настройки окон (размер и положение) сохраняются только на время сеанса.
  - Настройки размера панелей главного окна сохраняются только на время сеанса.
  - При программном сохранении настроек выдается исключение.
  - Настройки клиентского приложения сохраняются только на время сеанса. Не отображается флажок **Устанавливать режим разрешения отладки при запуске** (для тонкого и толстого клиента). Не отображается флажок **Устанавливать режим низкой скорости соединения при запуске** (для веб-клиента).
  - Не сохраняется настройка периода в списках. В диалоге настройки периода недоступен флажок **Использовать эту настройку периода при открытии**.
- **Старт** (`Start`) - старт бизнес-процесса.

- ТолстыйКлиент (ThickClient) - право запуска толстого клиента.
- ТонкийКлиент (ThinClient) - право запуска тонкого клиента.
- Удаление (Delete) - удаление (основное).
- УправлениеИтогами (TotalsControl) - управление итогами регистра бухгалтерии и регистра накопления (установка периода, по который рассчитаны итоги, и пересчет итогов).
- Установка (Set) - установка значения, не сохраняемого в базе данных.
- Чтение (Read) - чтение (основное).
- ЧтениеИсторииДанных (ReadDataHistory) - предоставляет возможность читать записи истории данных.
- ЧтениеИсторииДанныхОтсутствующихДанных (ReadMissingDataHistory) - предоставляет возможность читать записи истории данных для отсутствующих данных. Отсутствующие данные - это удаленные данные или данные, которые недоступны благодаря ограничениям доступа к данным.