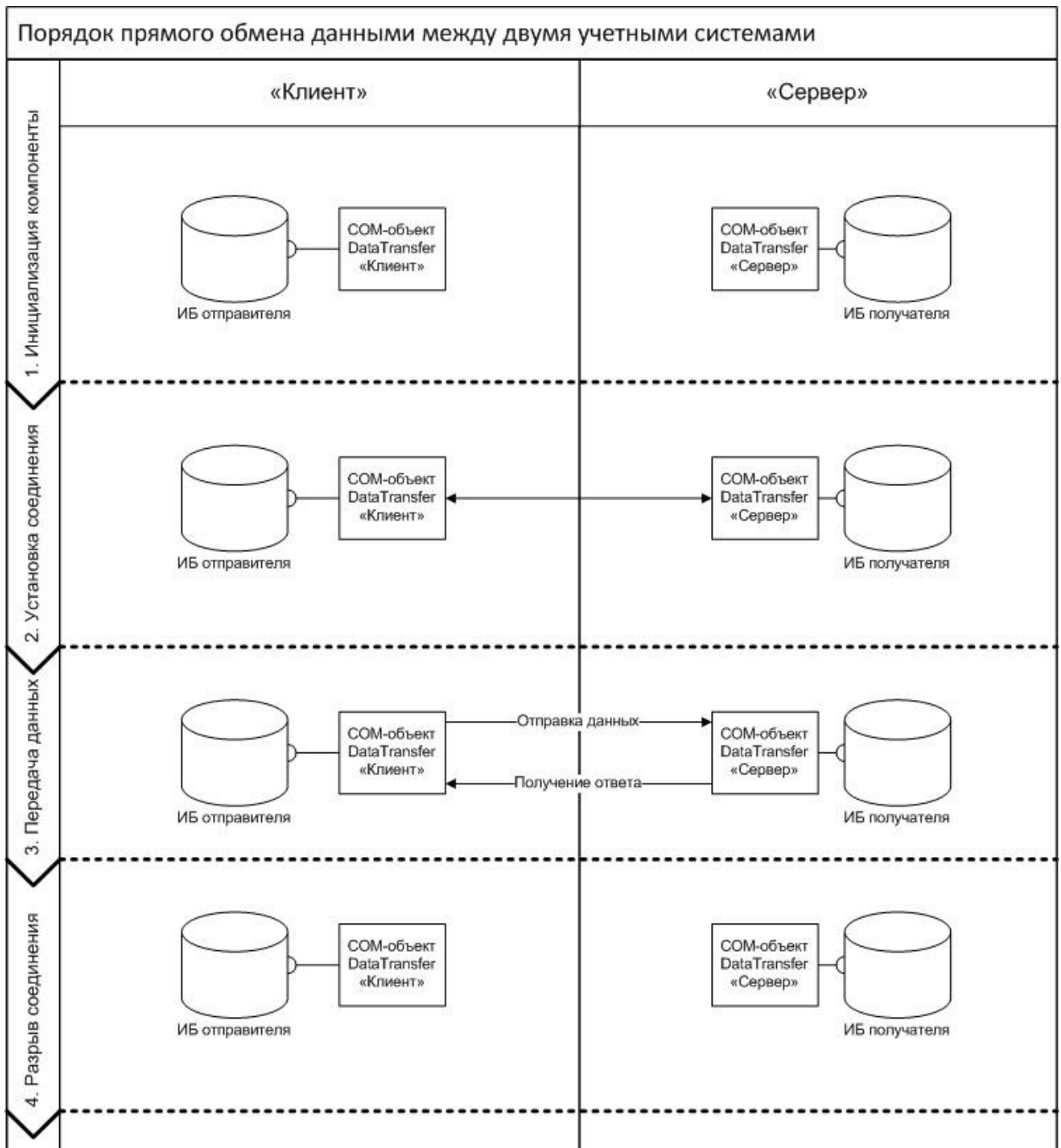


Описание программного продукта DataTransfer

Описание порядка работы

Программный продукт (далее - ПП) предназначен для прямого обмена данными между двумя учетными системами посредством «запрос-ответ» с гарантированной доставкой по протоколу TCP/IP: на одной стороне система выступает в качестве «клиента» (отправляет данные), на другой – в качестве «сервера» (ожидает данные). Учетные системы взаимодействуют с ПП посредством COM-соединения. Перед началом использования ПП необходимо скопировать файл «DataTransfer.dll» и зарегистрировать на сторонах обмена с помощью команды операционной системы «regsvr32».



1. Инициализация.

На стороне «клиента» (отправителя данных) создается экземпляр объекта с указанием IP-адреса или DNS-имени компьютера получателя и номером порта, на который будет происходить отправка данных.

На стороне «сервера» (получателя данных) создается экземпляр объекта с указанием номера порта, с которого ожидается получение данных, а также объект-ресурс, который будет обрабатывать данные и имя его метода-обработчика события получения данных, запускается режим ожидания соединения.

Возможно одновременное использование одного и того же экземпляра объекта как в роли «клиента», так и в роли «сервера».

2. Установка соединения.

Экземпляр объекта «клиента» устанавливает соединение с экземпляром объекта «сервера».

3. Передача данных.

На стороне «клиента» система готовит данные, трансформирует в текстовую строку и вызывает метод по отправке текстовой строки на сторону «сервера». Непосредственно перед отправкой можно убедиться в активности «сервера», а в случае неудачи (отсутствует связь с «сервером», «сервер» не отвечает и т.д.) можно перенаправить данные в резервную систему-получатель.

После успешного получения данных «сервер» обрабатывает их и в синхронном режиме отвечает «клиенту» при этом может вернуть, например, результатом обработки полученных данных может служить «уникальный идентификатор» (ИД) записанного элемента справочника или документа. Если произошел сбой при передаче данных, то «клиент» об этом узнает, получив ошибку, и может повторить отправку столько раз, сколько требуется.

4. Разрыв соединения.

Экземпляр объекта «клиента» разрывает соединение с экземпляром объекта «сервера», на стороне «сервера» освобождается ресурс-объект, а также выключается режим ожидания соединения.

Описание свойств и методов объекта *DataTransfer*

Свойства:

Connect

Описание:

Тип: Булево.

Присваивание значения **Истина** данному свойству запускает процедуру соединения с сервером, присваивание **Ложь** - отключение от сервера.

Передача данных методом **SendText** возможна только подключенному серверу.

Наличие ошибки в свойстве **LastError** после выполнения команды установки соединения сообщает о невозможности установки связи.

Host

Описание:

Тип: Строка.

Содержит адрес сервера (в форме IP-адреса или в формате доменного имени, например, ru.wikipedia.org), к которому производится подключение.

IsServer

Описание:

Тип: Булево.

Присваивание значения **Истина** данному свойству запускает «прослушку» порта **Port** данного экземпляра объекта и вызов метода обработки полученных данных имя которого указано в свойстве **ServerCallFunction**.

LastError

Описание:

Тип: Строка.

Содержит последнюю ошибку, возникающую при передачи данных.

Object1C

Описание:

Тип: Объект1С.

Содержит ссылку на объект учетной системы на стороне сервера, в котором располагается метод-обработчик получаемых данных.

Port

Описание:

Тип: Строка.

Содержит номер порта сервера, к которому производится подключение.

PortServer**Описание:**

Тип: Строка.

Содержит номер порта, который прослушивается данным экземпляром объекта.

Для прослушивания у объекта должно быть установлено свойство **IsServer**.

ServerCallFunction**Описание:**

Тип: Строка.

Содержит имя метода объекта учетной системы на стороне сервера, который обрабатывает получаемые данные.

Методы:**SendText****Синтаксис:**

SendText (<Строка>)

Параметры:

<Строка> (обязательный)

Тип: Строка.

Исходная строка.

Возвращаемое значение:

Тип: Строка.

Строка, обработанная и возвращенная сервером.

Описание:

Отправляет текстовую строку системе-получатель и ожидает ответа. Если при передаче возникла ошибка, ее значение помещается в свойство **LastError**.

SetReadTimeOut**Синтаксис:**

SetReadTimeOut(<при соединении >,< при получении клиентом >,< при получении сервером>)

Параметры:

<при соединении> (обязательный)

Тип: Строка.

Количество миллисекунд до вызова ошибки (исключения) при отсутствии связи, во время установки связи с сервером свойством **Connect = Истина**.

<при получении клиентом >(обязательный)

Тип: Строка.

Количество миллисекунд до вызова ошибки (исключения) при отсутствии связи, во время передачи данных методом **SendText**.

< при получении сервером>(обязательный)

Тип: Строка.

Количество миллисекунд до вызова ошибки (исключения) при отсутствии связи, во время приема передачи данных на сервере – установлено свойство **IsServer =**

Истина.