

DC Linux Service

Руководство программиста

Версия 2.0.9.4

История изменений документа

Версия	Дата	Перечень изменений
1.0	18.02.2020	Первоначальная версия
1.1	15.02.2021	1. Внесены изменения в пункты: «2.2 Системные требования», «3 Реализация интеграции с использованием DC Linux Service. Прямая интеграция с DC Linux Service по HTTP», «5.1 Основные параметры».
1.2	25.03.2021	2. Внесены изменения в пункт «4 Реализация интеграции с использованием DC Linux Service. Прямая интеграция с DC Linux Service по HTTP».
1.3	25.03.2021	Внесены изменения в пункты: «4.3 Инсталляция в системе», «6.1 Основные параметры».
1.4	24.09.2021	3. Внесены изменения в пункты: «4.3 Инсталляция в системе», «5 Реализация интеграции с использованием DC Linux Service. Прямая интеграция с DC Linux Service по HTTP», «7.1 Основные параметры».

Содержание

Правовая информация и сведения о поддержке продукта	4
1. Введение	5
2. DC Linux Service	5
2.1. Назначение.....	5
2.2. Системные требования	6
2.3. Установка в системе	6
2.4. Создание файла конфигурации XML.....	6
3. Прямая интеграция с DC Linux Service по HTTP.....	7
4. Настройка параметров «DC Linux Service».....	11
4.1. Основные параметры.....	11
4.2. Взаимодействие с оператором	12
5. Приложение	Ошибка! Закладка не определена.
5.1. Свойства объекта «SAPacket»	Ошибка! Закладка не определена.

Правовая информация и сведения о поддержке продукта

DC Linux Service. Версия 2.0.7.0 Руководство программиста: М.: ООО "Лаборатория платежных решений", 2023 — 16 с.

ООО "Лаборатория платежных решений" оставляет за собой право производить незначительные изменения программного обеспечения, касающиеся функциональности и внешнего вида конфигурационных систем, без внесения изменений в настоящее Руководство без специального уведомления.

Программное обеспечение и настоящий документ не могут быть скопированы, размножены, использованы по частям для составления других текстов, переведены на другие языки, если это не оговорено в письменной форме в договоре на поставку программного обеспечения.

Программное обеспечение, описанное в настоящем Руководстве, поставляется в соответствии с договором о поставке и может использоваться или копироваться только в соответствии с условиями этого договора.

Разработчиком и правообладателем программы DC Linux Service является ООО "Лаборатория платежных решений".

DC Linux Service Версия 2.0.7.0 © ООО "Лаборатория платежных решений" 2023

Для зарегистрированных пользователей ПО DC Linux Service открыты линии телефонных и E-Mail-консультаций. На консультацию имеет право пользователь, который приобрел ПО DC Linux Service в компании ООО "Лаборатория платежных решений".

Линия телефонных консультаций работает с понедельника по четверг с 10.00 до 18.00, в пятницу с 10.00 до 17.00 часов по московскому времени, кроме выходных и праздничных дней.

На линиях консультаций работают квалифицированные специалисты, которые ответят на Ваш вопрос немедленно или, возможно, попросят сформулировать вопрос в письменном виде и отправить по E-Mail.

Адрес: 127521, Москва, Октябрьская ул., 72
Телефоны для справок: (495)721-36-21
Факс: (495)721-36-21

4. Введение

Данный документ предназначен для прочтения программистами, осуществляющими организацию взаимодействия между клиентским ПО (в дальнейшем Клиент) и терминальным ПО «UNIPOS Terminal (с функционалом Smart Sale)» или «Unipos Droid (в режиме Smart Sale)».

5. DC Linux Service

5.1. Назначение

«DC Linux Service» - сервис для интеграции кассового ПО на базе систем семейства «Linux», реализующую интерфейс обмена с терминалом по протоколу SA. Обмен данными между кассой и «DC Linux Service» выполняется с помощью «HTTP-запросов».

«DC Linux Service» предоставляет возможность терминалу, подключенному по COM/ USB или TCP/IP, работать в режиме клиента и самостоятельно инициировать сеанс связи с коннектором. Благодаря этому при отсутствии у терминала своего канала связи с внешней сетью возможно независимое от кассового ПО взаимодействие с серверами сети «TCP/ IP» через сервис, используя коммуникации кассы.

Кассовый сервис «DC Linux Service» способен обрабатывать запросы от терминала.

Функции кассового сервиса:

- Обработка входящих запросов от кассового ПО и терминала;
- Открытие и закрытие TCP-соединений с хостами;
- Управление приоритизацией соединений терминала;
- Трансляция сообщений между TCP-соединениями и интерфейсом RS232. Трансляция сообщений между RS232 и другими программными модулями;
- Инициация служебных и тестовых операций с ККМ с помощью меню кассира («Проверка соединения», «Выгрузка логов» и т.п.).

На рисунке ниже (Рисунок 1. Организация взаимодействия с DC Linux Service) представлена схема взаимодействия участников процесса обмена данными.

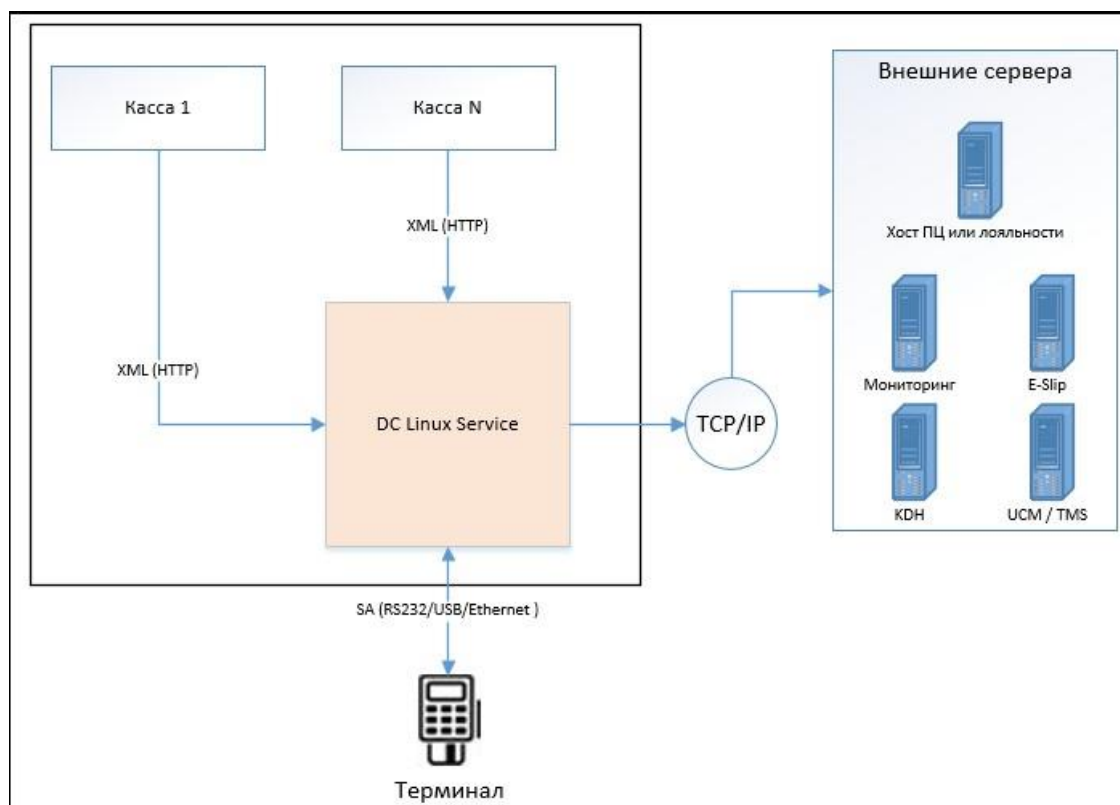


Рисунок 1. Организация взаимодействия с DC Linux Service

- **DC Linux Service** – отвечает за следующую логику работы:
 - обрабатывает вызовы от терминала, т.е. непрерывно «слушает» COM-порт и Ethernet, обеспечивает терминал коммуникациями по его инициативе. При установке терминалом соединения с внешним хостом через «**DC Linux Service**», сообщения транслируются на указанный адрес сети TCP/IP без модификации.
 - управляет приоритетом соединений: в определенных случаях разрывает поток обмена терминала с хостом для доставки запроса кассы на терминал.

5.2. Системные требования

- Версия ОС Linux 'Ubuntu' и 'CentOS';
- Разрядность x32 или x64;
- Java7, 8, 11;
- Права доступа «Администратор».

Внимание!

На кассе должна быть установлена Java версии 7 (или более), путь до файла 'java' должен быть указан в системной переменной «**PATH**» для всех пользователей (для системных, из-под которых запускаются службы).

5.3. Установка в системе

Установка «**DC Linux Service**» в системе осуществляется с помощью установочного скрипта. Сервис «**Dual Connector**» регистрируется в системе с помощью «**init.d**» или «**system.d**».

Путь к файлу «**connector.xml**» - «**/etc**».

Путь к файлу «**dcservice.jar**» - «**/usr/local/service/dcservice**».

Путь к файлу «**dcconsole.jar**» - «**/usr/local/bin/**».

Путь к файлу «**dcconsole.sh**» - «**/usr/local/bin/**».

Путь к файлу логов по умолчанию – «**/var/log/dualconnector**».

Примечание.

При старте «**DC Linux Service**» первоначально записывает файл лога в папку «**/var/log/dualconnector**», после анализа файла «**Connector.xml**» лог файлы начинают записываться в папку по умолчанию, если в файле не указан параметр «**LOG_PATH**» или для данного параметра не заданного значения (путь к папке). В противном случае файлы логов сохраняются по указанному в параметре «**LOG_PATH**» пути.

Лицензия отсутствует.

5.4. Создание файла конфигурации XML

Настройка файла конфигурации сервиса «**DC Linux Service**» выполняется в папке «**/etc/Connector.xml**».

6. Реализация интеграции с использованием DC Linux Service.

6.1. Прямая интеграция с DC Linux Service по HTTP

Обмен между кассовым ПО и DC Service осуществляется по HTTP протоколу.

В HTTP запросе может быть использован один из двух методов «**POST**» или «**GET**». При использовании иных методов возможно появление ошибки «**3**».

Пример HTTP заголовка с использование метода POST:

```
POST / HTTP/1.1
Host: 10.35.91.20:9015
User-Agent: curl/7.53.1
Accept: */*
Content-Type: text/html
Content-Length: 305
```

Пример HTTP заголовка с использование метода GET:

```
GET / HTTP/1.1
Host: 10.35.91.20:9015
User-Agent: curl/7.53.1
Accept: */*
Content-Type: text/xml
Content-Length: 305
```

Примечание. Если используется метод «**GET**», то в ответе возможно дополнительно получить статус работы DC с устройством. Например, DC вернет код ответа HTTP «**200**», если в данный момент DC обменивается данными с терминалом или код ответа HTTP «**404**», если DC ожидает команды.

Для корректной обработки данных в http-заголовке необходимо указываться кодировку, в которой передаются данные в HTTP запросе. Описание кодировки указывается в заголовке «**Content-Type**», например, «**Content-Type: text/xml; charset=windows-1251**». Данные будут интерпретированы в указанной кодировке. Если кодировка не указана, то данные будут интерпретированы в кодировке по умолчанию – «**ASCII**».

В заголовке «**Accept**» регулируется ответ на полученный запрос от кассы. При ошибке, в ответе формируется и передается файл «**XML**» или «**HTML**».

Если заголовок «**Accept**» имеет значение «**text/xml**», то при возникновении ошибки, передается ответ в виде «**XML**» документа. Код ответа HTTP в этом случае «**200**».

Пример ответа при возникновении ошибки при значении параметра Accept: text/xml:

```
<?xml version="1.0" encoding="windows-1251" standalone="no"?>
<response>
  <errorcode>4</errorcode>
  <errordescription>REQUEST_ERROR</errordescription>
</response>
```

При этом, возможные значения «**errorcode**»:

- **0** – DC не смог передать ответ от терминала. Возможная причина, внутренняя ошибка DC;
- **1** – истекло время исполнения операции. Устанавливает поле «**timeout**» в запросе;
- **3** - общая ошибка, ошибка параметров DC, не поддерживаемый метод HTTP, неизвестная ошибка, ошибка во время работы DC;
- **4** - ошибка поля или ошибка запроса, как между кассой и DC, так и между DC и терминалом;

- **13** - ошибка обмена данными между DC и терминалом. Возможная причина в отсутствии устройства или ответ не по протоколу SA;
- **15** – обмен данными прерван, например, выключение DC, мониторинг подключения устройства или отменен другим обменом;
- **16** – устройство занято, например, занято другим обменом.

Если заголовок **«Accept»** имеет значение отличное от **«text/html»**, то при возникновении ошибки, передается ответ в виде **«HTML»** документа с указанием результата и описание ошибки. Код ответа HTTP в этом случае **«400»**, **«404»**.

Пример ответа при возникновении ошибки при значении параметра Accept: text/html:

```
HTTP/1.1 404 Not Found
Content-Type: text/html; charset=UTF-8
Date: Fri, 14 Apr 2023 13:33:47 GMT
connection: close
Content-Length: 79
```

```
<h1>DEVICE_ERROR</h1><h>Can't open device /dev/ttyS0. Serial port /dev/ttyS0 not found</h>
```

При этом, возможные значения **«H1»**:

- OK - Нет ошибок;
- OK_PARTIAL - Нет ошибок, часть пакета;
- IO_PENDING - Недостаточно данных;
- ERROR - Общая ошибка;
- FIELD_ERROR - Неверное значение поля;
- DATA_ERROR - Ошибка данных;
- DATA_LEN_ERROR - Ошибка длины данных;
- ACK_ERROR - Ошибка отправки/получения ACK;
- CRC_ERROR - Ошибка контрольной суммы;
- NAK_ERROR - Ошибка. превышено количество негативных подтверждений;
- PACKET_ERROR - Ошибка. разбора пакета;
- ERR_EOT - Получен сигнал завершения соединения;
- DEVICE_ERROR - Ошибка устройства;
- TIMEOUT - истекло время ожидания;
- CANCEL – Отменено;
- DEVICE_BUSY - Устройство занято;
- RESPONSE_ERROR - Устройство занято;
- EXCHANGE_TIMEOUT - истекло время обмена данными;
- EXCHANGE_END_WITH_TERMINAL - сессия обмена с терминалом завершена.

Непосредственно запрос оформляется в формате **«XML»** и состоит из заглавного тега XML и контейнера с перечнем полей.

В случае запроса, перечень полей перечисляется в контейнере **«<request>»**, в случае ответа кассе, аналогичный перечень полей перечисляется в контейнере **«<response>»**. Каждое из полей описывается тэгом **«Field»**, номер поля указывается атрибутом **«id»**.

Атрибут **«hex»** указывает на формат данных HEX строки, которая требует переконвертации в бинарные (в примере ниже выделено красным шрифтом).

Например.

```
<?xml version="1.0" encoding="UTF-8"?>
<request>
  <field id="00">100</field>
```



```

<field id="04">643</field>
<field id="21">20150729121815</field>
<field id="25">1</field>
<field id="26"> </field>
<field id="27">00199991</field>
<field id="90">0102030405060708090a0b0c0d0e0f10</field>
<field hex="true"
id="86">EE38D10430303530D22839353437384134374241444238444333333731313233383034333341
364436384438424145424534D306343038313131</field>
</request>

```

Где:

№ поля в XML/ Поле протокола SA	Описание
0	Сумма операции, выраженная в минимальных единицах валюты
4	Код валюты операции
21	Оригинальная дата и время операции YYYYMMDDHHMMSS на внешнем устройстве
25	Код операции
26	Уникальный номер транзакции на стороне внешнего устройства
27	Идентификатор внешнего устройства (Terminal ID)
86	Дополнительные данные
90	Данные для печати на чеке

Примечание. Перечень полей, которые необходимо передавать для конкретного типа операции, указан в отдельном документе «Кассовые решение на базе SmartSale.»

Формат запроса предусматривает возможность указать напрямую адрес терминала, подключенного по «TCP/IP» или «COM/USB» и максимальное время выполнения операции. (При этом данные для соединения, указанные в файле настройки «Connector.xml» (см. п.п. 4.) будут игнорироваться). Данная возможность позволит маршрутизировать запросы на тот или иной терминал, когда к кассовой сети подключено большое количество терминалов, и регулировать максимальное время выполнения операции.

Для терминалов, подключенных по COM/USB:

- **ncom** – путь до подключенного устройства;
- **baudrate** – скорость порта (опциональный параметр).

Для терминалов, подключенных по TCP/IP:

- **ipaddr** – IP-адрес и порт подключения терминала, формат: 192.168.0.2:9000;

Максимальное время операции (для всех типов коммуникаций):

- **timeout** – предоставляемое время на выполнение операции в секундах. Данный таймаут защищает вызываемое приложение от «вечного» зависания в случае нештатной ситуации. Должен рассчитываться из принципа разумной необходимости и немного превосходить суммарное расчётное время на ввод карты клиента, ввод пинкода клиентом, обмен данными. Рекомендуемое значение 180 (3 минуты). Если таймаут не указан, то значение равно 0, бесконечное ожидание. Параметр обрабатывается, только для команд от кассы.

Пример запроса с указанием ip-адреса терминала:

```

<request>
<field id="00">100</field>
<field id="04">643</field>
<field id="21">20150729121815</field>
<field id="25">1</field>
<field id="26"> </field>
<field id="27">00199991</field>

```

```
<field id="90">0102030405060708090a0b0c0d0e0f10</field>  
<ipaddr>192.168.0.2:9000</ipaddr>  
<timeout>180</timeout>  
</request>
```

Пример запроса с указанием COM-порта, к которому подключен терминал:

```
<request>  
  <field id="00">100</field>  
  <field id="04">643</field>  
  <field id="21">20150729121815</field>  
  <field id="25">1</field>  
  <field id="26"> </field>  
  <field id="27">00199991</field>  
  <field id="90">0102030405060708090a0b0c0d0e0f10</field>  
  <ipaddr>192.168.0.2:9000</ipaddr>  
  <timeout>180</timeout>  
  <ncom>/dev/ttyS0</ncom>  
  <baudrate>115200</baudrate>  
</request>
```

В случае необходимости прерывания операции с терминалом до получения результата выполнения операции на стороне терминального ПО, со стороны кассового ПО необходимо отправлять пакет RST.

6.2. Реализация работы с DC Service через компоненты DC Console Linux Service

Для реализации работы с DC Service можно использовать компоненту «DC Console Linux Service». Компонента «DC Console Linux Service» предназначена для организации взаимодействия кассового ПО и ПО терминала с использованием параметров командной строки. Подробнее работа с компонентой DC Console описана в документе «DCConsole Linux Service.pdf».

7. Настройка параметров «DC Linux Service»

7.1. Основные параметры

Основной файл параметров должен находиться в директории «**/etc/Connector.xml**» и называться «**Connector.xml**».

Содержит данные в следующей структуре xml:

```
<ROOT>
  <SERVERPORT>9015</SERVERPORT>
  <LOG_TYPE>DEBUG</TYPE>
  <LOG_PATH>/var/logs</PATH>
  <LOG_CLEARTIME>30</LOG_CLEARTIME>
  <CONFIRM_OPERATION>ON</CONFIRM_OPERATION>
  <TRIPLEACK>ON</TRIPLEACK>
  <DEVICES_TYPE>TERMINAL</DEVICES_TYPE>
  <CONNECTION_TYPE>COM</CONNECTION_TYPE>
  <CONNECTION_PORT>/dev/ttyS0</CONNECTION_PORT>
  <CONNECTION_BAUDRATE>115200</CONNECTION_BAUDRATE>
  <IPADDRESS>10.35.1.40:1006</IPADDRESS>
  <IPADDRESSGUI>127.0.0.1:6000</IPADDRESSGUI>
  <WAITACK>6</WAITACK>
  <WAITPACKET>45</WAITPACKET>
  <CONNECT_TIMEOUT>20</CONNECT_TIMEOUT>
  <EXCHANGE_TIMEOUT>180</EXCHANGE_TIMEOUT>
  <RECONNECTION_DELAY>6</RECONNECTION_DELAY>
  <HEX_STRING_FORMAT>ON</HEX_STRING_FORMAT>
</ROOT>
```

1. **ROOT** – корневая область. Наличие обязательно.
2. **SERVERPORT** – порт, по которому доступны запросы к сервису. Наличие обязательно.
3. **LOG_TYPE** – тип детализации информации в файле лога. Допустимые значения в порядке увеличения выводимой информации: «**OFF**», «**SYSTEM**», «**ADVANCED**», «**DEBUG**», «**VERBOSE**». Наличие необязательно, по умолчанию «**ADVANCED**».
4. **LOG_PATH** – путь сохранения файлов лога. Если в параметре не указан путь, куда сохранять файлы логов или вообще отсутствует данный параметр, то файлы логов сохраняются в директорию по умолчанию «**/var/log/dualconnector**».
5. **LOG_CLEARTIME** – время хранения логов (в днях). Диапазон возможных значений от 1 до 365 дней. Если параметр не задан, используется значение по умолчанию 30 дней.
6. **CONFIRM_OPERATION** – секция настройки включения автоматического подтверждения операции на стороне внешнего устройства. Наличие не обязательно. По умолчанию, выключено.
7. **TRIPLEACK** – секция настройки отправки 3 символов подтверждения (ACK) по завершении операции на терминал. Наличие не обязательно. По умолчанию выключено.
8. **DEVICES_TYPE** – тип терминала. Допустимые значения «**TERMINAL**», «**PINPAD**». Наличие необязательно. По умолчанию «**TERMINAL**». Отличие типов используется для определения наличия принтера.
9. **CONNECTION_TYPE** – тип подключения к терминалу. Допустимые значения «**COM**», «**IP**». Наличие обязательно.
10. **CONNECTION_PORT** – путь до подключенного устройства, например, «**/dev/ttyS0**».
11. **CONNECTION_BAUDRATE** – скорость обмена. Наличие необязательно. По умолчанию 115200.
12. **IPADDRESS** – IP адрес терминала. Наличие обязательно при соединении по IP.
13. **IPADDRESSGUI** – если касса обрабатывает команды, то указывает IP-адрес и порт кассы.
14. **WAITACK** – время ожидания сигнала подтверждения получения пакета в секундах. Наличие необязательно, по умолчанию 5.

15. **WAITPACKET** – время ожидания ответного пакета в секундах. Наличие необязательно, по умолчанию 45.
16. **CONNECT_TIMEOUT** – механизм прерывания установки соединения по истечению времени. Указывается время ожидания соединения с сервером в секундах. Значение по умолчанию – 30 секунд.
17. **EXCHANGE_TIMEOUT** – устанавливает максимальное время выполнения операции в секундах.
18. **RECONNECTION_DELAY** – устанавливает задержку в секундах на повторное подключение/соединение к серверу после отключения в секундах.
19. **HEX_STRING_FORMAT** – указывает формат поля в ответе (response) в XML-файле. При необходимости перекодировывает поля в данные для передачи в XML документе и добавлен атрибут «hex». Если данный параметр не задан или имеет значение «ON», то будет проведена конвертация, при необходимости. Если данный установлен и имеет значение «OFF» или любое другое значение, отличное от «ON», то будет проведена нормализация данных к XML формату. Если данные пришли с атрибутом «hex», то данный будут конвертированы из HEX строки в бинарные данные вне зависимости от параметра.

8. Взаимодействие с оператором

«DualConnector 2.0» имеет возможность транслирования запросов терминала для отображения сообщений кассиру. Взаимодействие организовано альтернативным способом.

Альтернативный способ вывода окон необходим, когда производитель кассового ПО решает использовать свою реализацию диалоговых окон. В данном случае на стороне кассового ПО необходимо реализовать сервер вывода окон. Адрес сервера для приема терминальных окон настраивается в файле конфигурации Connector.xml, параметр IPADDRESSGUI.

Информационное сообщение

Предназначено для информирования кассира о событии. Ответ кассира не требуется.

Формат запроса:

```
<request>
  <type>1</type>
  <data>4^^Заголовок^Сообщение</data>
  <timeout>10</timeout>
</request>
```

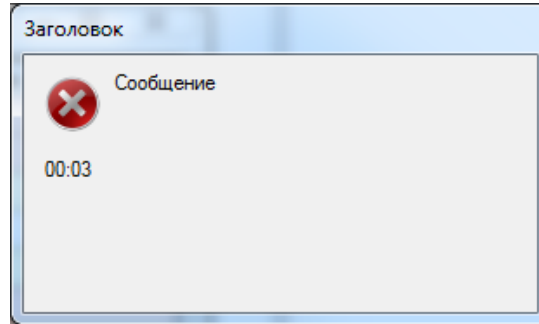
Формат ответа:

```
<response>
  <type>1</type>
  <data>0</data>
</response>
```

Где:

- type - 1 – информационное сообщение;
- data (в запросе) - данные для отображения в соответствии с форматом данных для отображения (см. ниже);
- timeout - время отображения окна в секундах;
- data (в ответе) - ответ кассира (в данном случае 0, кассир ничего не нажимал и не должен);

Пример экранной формы:



Т.к. данное сообщение не требует ответа кассира, ответ должен поступить без задержки.

Сообщение подтверждения

Предназначено для запроса у кассира определённого ответа.

Формат запроса:

```
<request>
  <type>2</type>
  <data>3^5^ Заголовок^Сообщение </data>
  <timeout>30</timeout>
</request>
```

Формат ответа:

```
<response>
  <type>2</type>
  <data>32</data>
</response>
```

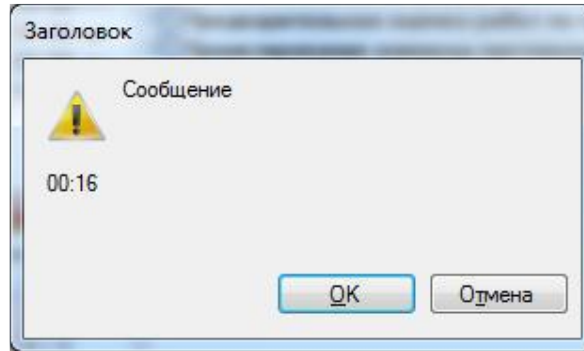
Где:

- type - 2 – сообщение подтверждения;
- data (в запросе) - данные для отображения в соответствии с форматом данных для отображения (см. ниже);
- timeout - время отображения окна в секундах;
- data (в ответе) - ответ кассира.

Возможные значения поля data в ответе кассира:

- 0 – кассир ничего не нажимал
- 1 – нажал ОК;
- 2 – нажал ответ ДА (Yes);
- 4 – нажал ответ ОТМЕНА (Cancel);
- 8 – нажал ответ НЕТ (No);
- 16 – вышло время диалога (timeout);
- 32 – кассир нажал Escape(закрыл форму без выбора варианта ответа);
- 64 – переданы ошибочные параметры, диалог не отображён.

Пример экранной формы



Сообщение выбора из списка

Предлагает кассиру выбрать вариант из списка.

Формат запроса:

```
<request>
  <type>3</type>
  <data>2^1^ Заголовок^Сообщение </data>
  <adata>RUB;USD;EUR</adata>
  <timeout>30</timeout>
</request>
```

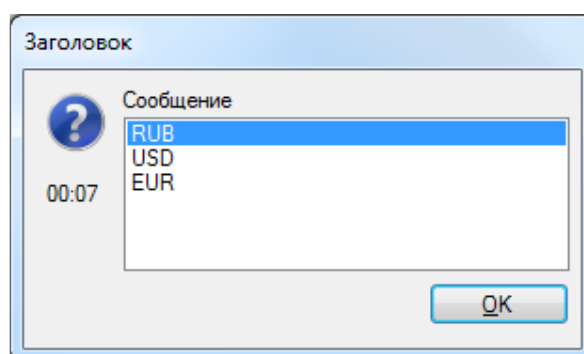
Формат ответа:

```
<response>
  <type>3</type>
  <data>1</data>
  <adata>4</adata>
</response>
```

Где:

- type - 3 – сообщение выбора;
- data (в запросе) - данные для отображения в соответствии с форматом данных для отображения (см. ниже);
- adata (в запросе) – (additional) список вариантов, разделённых символом '\n' или ';' ;
- timeout - время отображения окна в секундах;
- data (в ответе) - ответ кассира, варианты описаны ранее.
- adata (в ответе) – вариант выбора кассира в представлении N=2m.
 - где m – индекс строки, начиная с 0. Т.е. первая строка будет 1, вторая - 2, третья – 4, четвёртая – 8 и т.д.

Пример экранной формы



Сообщение ввода данных

Запрашивает у кассира символные данные.

Формат запроса

```
<request>
  <type>4</type>
  <data>1^1^ Заголовок^Сообщение </data>
  <adata>000999</adata>
  <timeout>30</timeout>
</request>
```

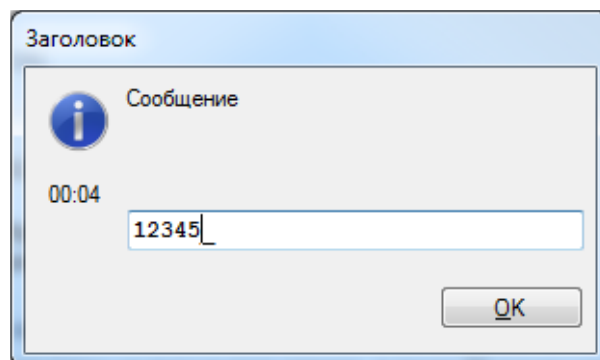
Формат ответа:

```
<response>
  <type>4</type>
  <data>1</data>
  <adata>12345</adata>
</response>
```

Где:

- type - 4 – сообщение ввода данных;
- data (в запросе) - данные для отображения в соответствии с форматом данных для отображения (см. ниже);
- adata (в запросе) – (additional) маска ввода.
- timeout - время отображения окна в секундах;
- data (в ответе) - ответ кассира, варианты описаны ранее.
- adata (в ответе) – введенные данные.

Пример экранной формы



Сообщение печати данных

Касса распечатывает данные на принтере.

Формат запроса:

```
<request>
  <type>5</type>
  <data>ДАННЫЕ ДЛЯ ПЕЧАТИ</data>
</request>
```





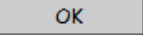
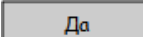
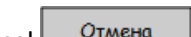
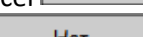
Формат ответа:

```
<response>
  <type>5</type>
  <data>0</data>
</response>
```

Где:

- type - 5 – сообщение печати данных;
- data (в запросе) - данные для печати. Строки заранее отформатированы, разделены символом '\n';
- data (в ответе) - ответ. 0 – успешная печать, 64 – произошла ошибка.

Формат данных для отображения

Элемент подполя	Описание	Атрибут	Тип поля
Уровень сообщения	<p>Определяет стиль иконки окна, выводимого на ККМ. Может принимать значения:</p> <ul style="list-style-type: none"> • 1  MB_INFORMATION - информирование кассира • 2  MB_ICONQUESTION - запрос кассиру, требующий ответа • 3  MB_ICONEXCLAMATION, MB_ICONWARNING - сообщение об ошибке или предупреждение • 4  MB_ICONSTOP критическая ошибка 	O	n1
^	Разделитель между элементами данных внутри поля.	M	
Элемент управления	<p>Определяет элементы управления (кнопки), которые должны быть отрисованы в окне, выводимом на ККМ. Поле представляет собой битовую маску:</p> <p>0x01 – Ok  MB_OK</p> <p>0x02 – Yes  MB_YES</p> <p>0x04 – Cancel  MB_CANCEL</p> <p>0x08 – No  MB_NO</p>	O	n..3
^	Разделитель между элементами данных внутри поля.	M	
Заголовок сообщения	Заголовок окна, выводимого на ККМ	O	an..40
^	Разделитель между элементами данных внутри поля.	M	
Сообщение кассиру	Строка (или набор строк, разделенных символом «\n» или «;»), содержащая текст информационного сообщения для кассира	M	an..950

M – mandatory: обязательный элемент;

O – optional: опциональный элемент, может отсутствовать.

Пример:

Все элементы присутствуют	1^1^ОПЛАТА ТОВАРА^ПОДТВЕРДИТЕ ДАННЫЕ\nНАЖМИТЕ ОК
Заголовок и элементы управления (кнопки) отсутствуют	1^^^УСТАНОВКА\nСОЕДИНЕНИЯ
Уровень сообщения отсутствует	^2^ОПЛАТА^ВВЕДИТЕ\nНОМЕР ЧЕКА