INSTALLATION AND ADMINISTRATION GUIDE
«PalmOrder: Pre-Selling»
Version 6

2019

Content

About...

The “PalmOrder: Pre-Selling” system was created by the team of developers in 2004 in cooperation with partners and potential customers.

PalmOrder will save up to 50% of your employees' working time, let you avoid mistakes of written and oral communication, and, as a result, increase your business effectiveness and profit.

Dozens of companies in the field of trade and services are using PalmOrder. This system is very flexible in operation, so its use is rational in all business areas, where the use of mobile technology is possible.

Updates

PalmOrder is a constantly evolving application. Free updates are released regularly to enhance the functionality, performance and ergonomics of the system. The customers' wishes are the main priority in the development of new program features.

Cooperation

We are always glad to cooperate with you. If you have any requests, suggestions or questions about the software, we are ready to consider them while developing new versions of the system.

We will answer any of your questions or discuss suggestions at the earliest.

Partnership

We offer favorable conditions for companies wishing to become our partners and representatives. If you are an ordinary man but have high aspirations and ambitions, we will help you to realize them in the promotion of mobile technologies.

Abbreviations and Terms

Here are some abbreviations and terminology used in this guide:

PDA	a Pocket Personal Computer or Personal Digital Assistant. Means any device with an Android operating system, where the mobile part of PalmOrder is installed. Physically it can be a smartphone, tablet, phablet etc.
PC	MS Windows or Linux personal computer, where the server part of PalmOrder is installed
Android	an Android operating system, were the mobile part of PalmOrder is installed
MS Windows	MS Windows operating system, where the server part of PalmOrder is installed
AIS (ERP)	an office accounting system (e.g. 1C: Enterprise or other AIS or DBMS) the PalmOrder integrates with
Mobile Employee	a person that uses the mobile part of the system. Usually it is a sales representative, an agent etc.
OS	an operating system
DB	a database

Intro

This chapter describes how to install and administrate PalmOrder – the system for pre-selling, mobile commerce and merchandising (hereinafter referred to as “PalmOrder” with regard to the entire system).

This guide is intended for specialists who install, configure and administrate the system.

Hardware Requirements

Server Part

The server part of the PalmOrder system is designed to run on the IBM-compatible personal computers. A computer, where the server part of PalmOrder will be installed, must meet the following requirements:

- OS: MS Windows (XP or higher).
- Pentium II 500 MHz or higher.
- RAM 128 Mb or more.
- Hard Disc (about 30 Mb of free space).
- Internet connection to provide communication with a PDA.

Mobile Part

The mobile part of the system is designed to run on PDAs (smartphones, tablets), that should meet the following requirements:

- OS: Android 2.2 or higher.
- Wi-Fi – to provide a wireless local data exchange.
- 2G or higher Internet connection – to provide a remote data exchange.
- GPS module desired

Must-Know

Required skills for a PalmOrder System Administrator:

- knowledge of the OS, where the server part of PalmOrder is installed (MS Windows or Linux);
- installation and maintenance of the operating system efficiency;
- setting up MS Windows operating system using the Control Panel;
- the ability to use the Windows command line (cmd.exe);
- the knowledge of the basic principles of the TCP/IP protocol;
- the knowledge of an XML format;
- the knowledge of how to install a program on a PDA.

If you are not familiar with all the concepts and skills listed above, make sure to refer to the relevant documentation.

Installing

PalmOrder consists of application parts for MS Windows and Android, which are intended to conduct pre-selling and mobile data collection.

PalmOrder consists of two parts:

- The server part — is installed on an office PC. The structure of the server part includes:
  program modules that allow you to perform data exchange with a PDA;
  utility for converting XML files into DB format (for installation on a PDA);
- The mobile part — is installed on a PDA of a mobile employee:
  “PalmOrder: Pre-Selling” for Android.

Description and Location of Installation Packages

PalmOrder installation packages are delivered over the Internet, and can be downloaded from the www.palmorder.com website. The structure description of web pages and installation packages:

- Pre-selling – this page contains all necessary parts for the PalmOrder installation.
- PalmOrder setup – the server part installation package.
- APK-file – the mobile part for installing on PDAs. We recommend to install from Google Play Market to get all the advantages of Android OS: update notifications, automatic installation of updates from the Internet.
- Updates – actual updates for both parts.
- Examples of integration into AIS or ERP – here are the examples of how to integrate PalmOrder into AIS (1C:Enterprise).
- Utilities / Drivers – various useful programs for PalmOrder.

General Order of Installation

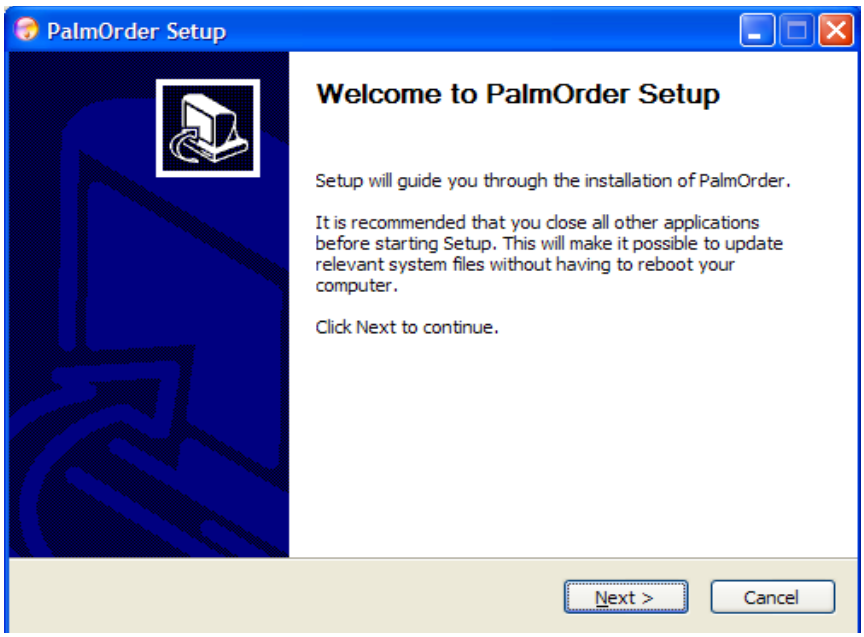
After you perform all actions mentioned below, the system is ready for use:

- Server part installation:
  - installing PalmOrderSetup;
  - installing an XML-parser;
  - installing a server license;
  - exchange settings.
- Mobile part installation:
  - installing PalmOrder apk-file;
  - downloading a license;
  - downloading exchange.xml.
- Downloading data from AIS (ERP) to server.
- Downloading data from server to a PDA.

The Server Part Installation

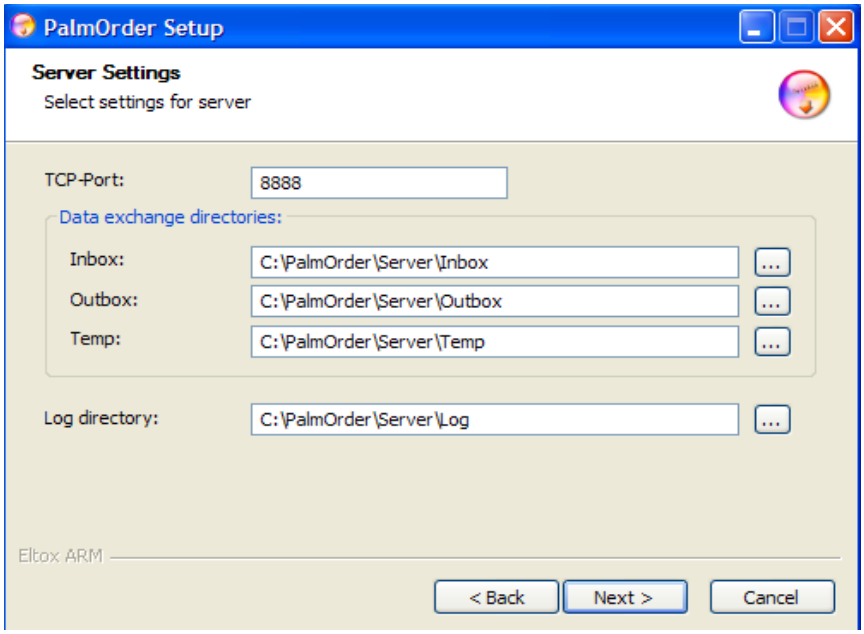
Installing PalmOrderSetup

Download the installation package to a PC you will use as a server, and execute the file. Choose a language and follow the instructions.



pic. 4.1 Server part installation

At step 6 you will be prompted to select server settings (pic. 4.2).



pic. 4.2 Selection of server settings

Server settings include:

- **TCP-port** – the TCP/IP port that will be reserved for data exchange between a PDA and a server PC.
- **Incoming** – the data received from a mobile device will be saved in this folder by default.
- **Outgoing** – the folder by default to keep the data for PDAs.
- **Temporary** – the folder to keep server temporary files.
- **Log folder** – the folder to keep server logs.

On the basis of the entered data, there will be created a file `palmserv.ini` with the

corresponding settings. A detailed description of this file is given below. Pay attention that a user, a server will run on behalf of, must have permissions to read/write files from/into these folders.

After installation, the icon “PalmOrder” will appear in the menu Start → Programs. Carefully read its content. Also to the Autorun of the operating system there will be added a shortcut to start the server.

Installing an XML-parser

As an XML-parser PalmOrder uses MSXML parser 4. Normally this parser is already pre-installed in MS Windows. If not, you can download it from our website in the section Utilities/Drivers: MS XMLParser v.4.0.

Installing a server license

After the server part is installed, when you start `palm_server.exe`, you will be prompted that the program runs in a demo mode. You will also see an activation code in the window that appears. You need to send this code to our email. In response you will receive a `palm_server.license` file.

Be careful when requesting a server license. It is directly tied to a computer the server is running on!

Copy the received file `palm_server.license` to a folder, where `palm_server.exe` is located (normally it is `<Installation_folder>\Server`) and restart `palm_server.exe`. If everything is correct, the server will start with no warnings.

Exchange settings

See chapter *Exchange Settings* for details. The `exchange.xml` file contains default exchange settings. It is located in the outgoing folder (normally it is `<Installation_folder>\Server\Outbox`).

Installation of the Mobile Part

Installing the PalmOrder apk-file

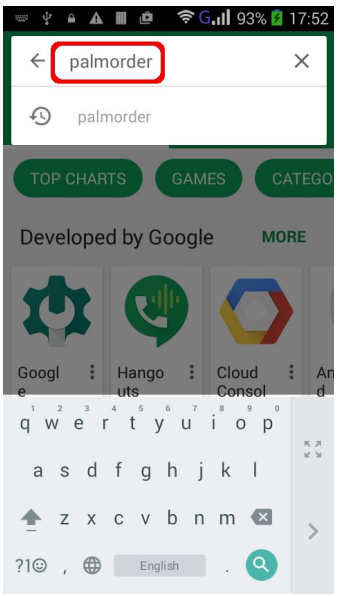
The application is the executable file of Android OS – PalmOrder.apk. You can use two sources to install it:

- Google Play Market – the main storage for Google Android applications;
- our website.

Both are identical. We recommend using the first source to experience all the advantages of Google Play Market (for example, update notifications). We will also be very grateful for your positive comment :-)

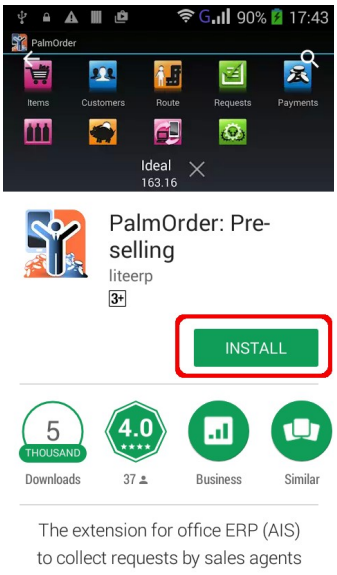
To install the application from Google Play:

- run Play Market;
- search for PalmOrder (use Search pic. 4.3);



pic. 4.3 «Play Market»

- tap the “Install” button (pic. 4.4) and confirm your choice in the next step by tapping “Accept”.



pic. 4.4 «PalmOrder in Play Market»

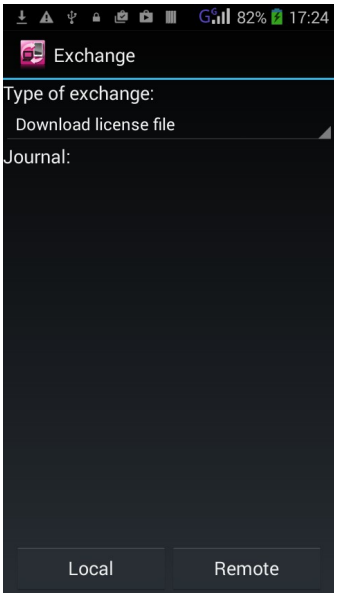
Installing a license

The license is an encrypted file that contains information about enabling application features. The license corresponds to a device name, which is used in synchronization with a PC. Each license is accompanied by a text file containing the above mentioned information.

Before installing the license, you must configure the exchange settings (see chapter *Configuring Exchange*). After that complete the following steps:

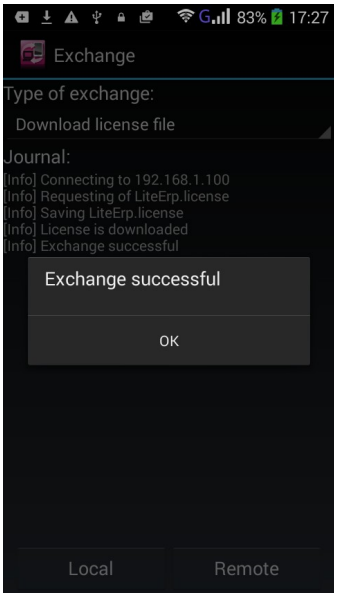
- Copy the file `LiteERP.license` to `<Installation_folder>\Server\Outbox` folder.

- Run PalmOrder on your PDA, tap “Exchange”.
- Choose “Download license file” in a drop-down list (pic. 4.5).



pic. 4.5 «Preparing for a license download»

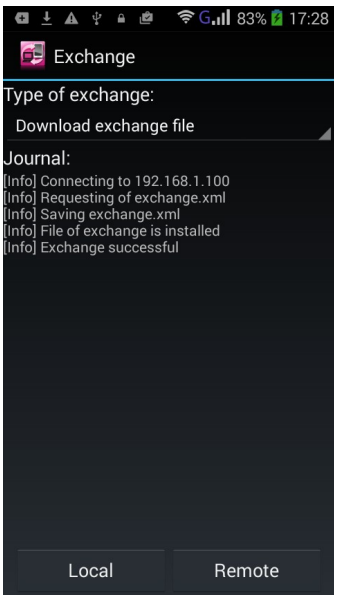
- Tap “Local” or “Remote” depending on the kind of exchange you are to perform.
- If everything is correct, you will see the confirmation message on the screen (pic. 4.6).



pic. 4.6 «The license was successfully downloaded»

Downloading exchange.xml

The file `Exchange.xml` is downloaded in a similar way. The only difference is that you need to choose “Download exchange file” in the drop-down list of the types of exchange (pic. 4.7).



pic. 4.7 «Downloading exchange.xml»

There is a standard file `exchange.xml` in `<Installation_folder>\Server\Outbox` folder.

Configuring Exchange

General Information

You have to configure the exchange modes to ensure a proper work of PalmOrder. Configuring exchange provides a proper data exchange between a server and a mobile device.

Exchange modes:

You can configure two exchange modes: local and remote. Their technical characteristics are identical; the division is simply for convenience.

Local connection (exchange) – means the exchange is executed through a local office wireless network. Connecting a PDA to a network is usually performed via Wi-Fi with direct connection to an AP office network.

Advantages:

there is no need in intermediary networks like the Internet.

Limitations:

you have to be located in your office to perform exchange.

Remote connection TCP/IP – includes exchange over the Internet. Connection to the Internet is possible in different ways: through a wireless network of a mobile operator or an available Wi-Fi connection. Naturally, your server PC must be connected to the Internet and have a permanent IP-address or a constant domain name.

Advantages:

no need to be present in the office.

Limitations:

you need a permanent Internet connection.

Remote connection via Dropbox – uses file hosting service that offers cloud storage of data (Dropbox) for exchange.

Advantages:

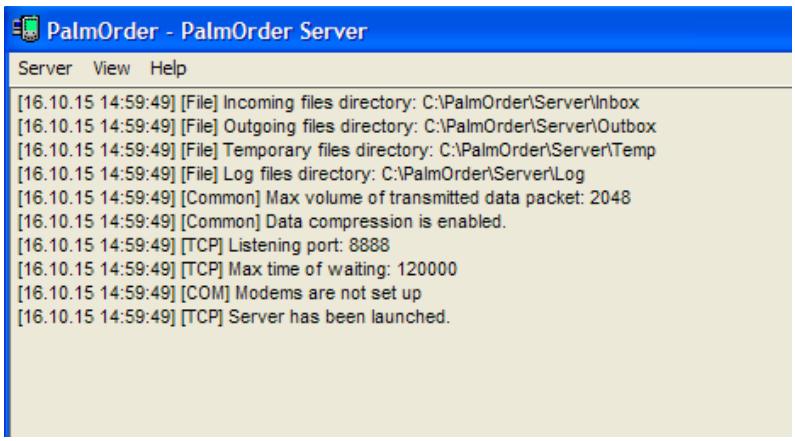
no need to run palm_server.exe for data exchange.

Limitations:

you need a permanent Internet connection.

PalmServer Settings

We use PalmServer to connect your server PC and the mobile part of PalmOrder on a PDA. PalmServer is installed on an office PC. References, lists, documents from AIS (ERP) are transmitted to a PDA. And data (documents) created by mobile employees can be sent in an opposite direction.



pic. 5.1 «PalmServer application»

PalmServer settings

PalmServer is configured by setting different parameters in the `palmserv.ini` file, which is located in `<Installation_folder>\Server`.

- Section `[Server]`
  - `tcpport` – the number of a TCP-port, which is used by PalmServer;
  - `tcptimeout` – the time of waiting for the response of the PalmOrder mobile part on the TCP/IP protocol stack in milliseconds (recommended value range 60000 - 180000);
  - `chunksize` – maximum data packet in bytes (recommended value range 1024 - 3072 bytes);
  - `compress` – the indicator of archiving data, that will be transferred to a PDA (database and references updates). If the value is set as `true`, the archiving is executed, in other cases – it's not. Value by default is `true`.
- Section `[Inbox]`
  - `path` – the path to a folder with data downloaded from a PDA.
- Section `[Outbox]`
  - `path` – the path to a folder with data to be uploaded to a PDA.
- Section `[Temp]`
  - `path` – the path to a folder with temporary files.
- Section `[Log]`
  - `screenlevel` – the details level of the server messages displayed on the screen. 0 means regular level, 1 is detailed (recommended value is 0);
    `filelevel` – the details level of the server messages displayed in a log-file. 0 means regular level, 1 is detailed (recommended value is 1);
  - `daily` – if the value is set as `true`, a new log-file is created in the folder `directory`. In case of any other value, only one log-file is created (named `file`);
  - `directory` – the folder for log-files;
  - `file` – the name of a log-file.
- Section `[PDASinchronizationName]`
  - `inbox` – the path to a folder with data downloaded from a device `PDASinchronizationName`;
  - `outbox` – the path to a folder with data to be uploaded to a device

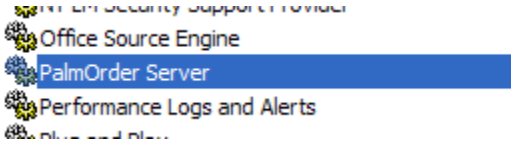
`PDASinchronizationName`.

Running PalmServer as a Windows service

To run PalmServer as a Windows service, you need to do the following:

- Delete the autorun of PalmServer from *Start → All programs → Autorun*;
- Register the service in the command line (*Start → All programs → Standard → Command line* → go to the folder where PalmServer is located) by specifying `palm_server.exe -install` (or `palm_server.exe -i`);

You can find the registered service in the Windows Service Manager (*Control panel → Administration → Services*).



pic. 5.2 «PalmOrder Data Exchange Service»

To delete the service, run `palm_server.exe -delete` (or `palm_server.exe -d`).

Configuring the Exchange of the PalmOrder Mobile Part

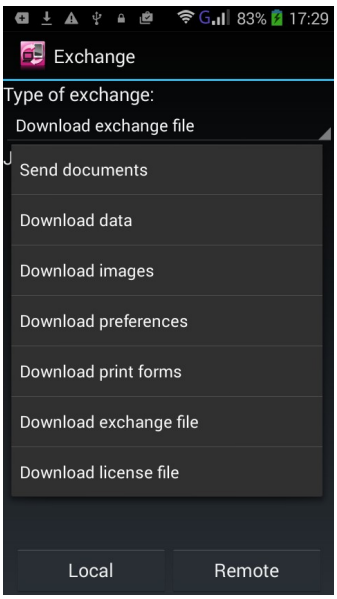
To perform the exchange, run PalmOrder on your PDA and tap “Exchange” (pic.5.3). The exchange is:

- transferring the collected data (documents, files etc.) to a server and
- receiving the new data from AIS (or ERP) (stock, debts, files etc).

The structure of the exchange data (what exactly should be transferred / received) is set up using the exchange configuration file. Each PDA can have its own exchange configuration file.

Only a PDA may initiate exchange.

You can set up different exchange rules (profiles) and choose a required one before initiation. For example, you can create a full exchange profile, or receive only actual data for references, or transfer created documents and files to server etc. Some of them are service profiles and appear depending on other settings or a current state of the system: “Download license”, “Download exchange file”, “Send DB and Log”.



pic. 5.3 «Exchange application of the mobile part»

Connection settings

The PalmOrder connection parameters (pic. 5.4) can be configured in the “Exchange menu” dialog window. Choose “Settings” icon in PalmOrder's main window to enter Exchange menu. Local and Remote TCP/IP types of exchange are configured identically:

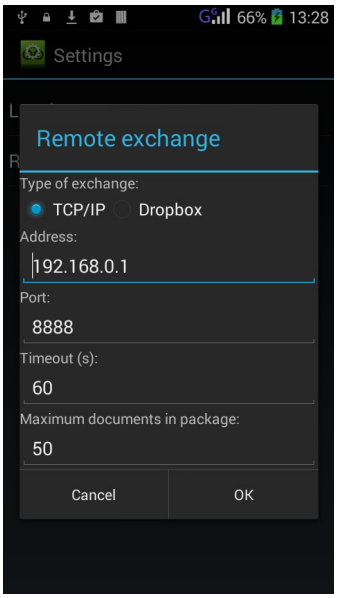
`Address (for local exchange)` – an IP-address or domain name of a server in a local Wi-Fi or other wireless network. Normally the address of a PalmServer PC in a local network is used here.

`Address (for remote TCP/IP exchange)` – an IP-address or domain name of a server PC for a remote exchange. Usually it's the IP-address (domain name) of a PalmServer PC in the Internet.

`Port` – a TCP/IP port used by PalmServer.

`Timeout (s)` – the time of waiting for a server response in seconds.

`Maximum documents in package` – documents are transmitted to a server in packages. This parameter sets the number of documents in each package.



pic. 5.4 «Configuring connection for exchange»

Usually setting the addresses and port is enough. Other settings can be left at default values. You need to change them only if transmission problems occur.

You can also configure your exchange via Dropbox service. See FAQ for details.

Configuring the connection to the Internet

To connect your PDA to the Internet, follow the instructions of your cell network operator.

Configuring the Exchange via Dropbox

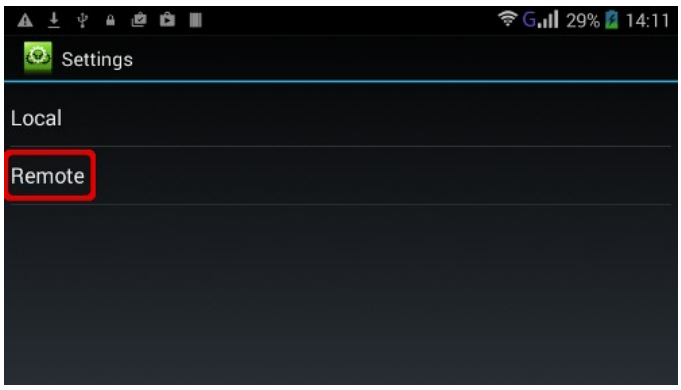
In some countries your Internet provider can impose some restrictions on the operation of your server. As a result, you may experience the difficulties with receiving the data. In this case, you can place the server part of PalmOrder on Dropbox.

First, install the server part of PalmOrder on your computer as a DOS application, not as a Windows service (see chapter *Installing*, paragraph *The Server Part Installation*). Then perform the following steps:

1. Get the e-mail addresses for each mobile device (license) and one more for a server (note, that you can use the existing e-mail addresses). You can employ the same e-mail address for all devices as well.
2. Use these addresses to register the Dropbox accounts (also for each mobile device (if you're using different addresses for your devices) and one for a server).
3. Install dropbox.exe on your server PC. Create the synchronization folder in the root of C: drive, for example, `C:\Dropbox`.
4. Copy all the files and folders from `C:\PalmOrder\Server` to `C:\Dropbox\Server`.
5. Sign into your Dropbox server account and share the folder named Server with other accounts, that are used by mobile devices. Accept the invitation in every account (enter every account sequentially and click the Apply button that will appear).
6. Correct the paths in `palmserv.ini`: they must be relative. It means, that if you share the folder named Server from the previous step, and folders Inbox and Outbox are inside it, your paths should look like this:
   ```
   inbox=.\Inbox
   outbox=.\Outbox
   ```
   If every device (license) has its own Inbox and Outbox folders, for example, the device SA001 downloads data from `C:\Dropbox\Server\Outbox\SA001`, and uploads data to `C:\Dropbox\Server\Inbox\SA001`, then in the section [SA001] of `palmserv.ini` you should enter:
   ```
   outbox=.\Outbox\SA001
   inbox=.\Inbox\SA001
   ```
   Do not change the paths of the Temp, Downloaded and Error folders, leave them as they are. There is also no need to place them in the Dropbox folder.

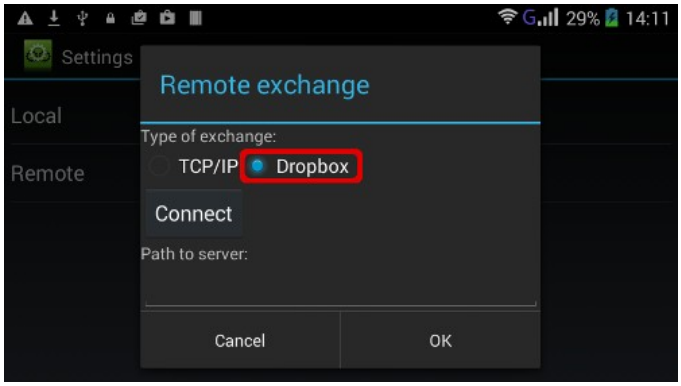
On every mobile device:

1. Launch “PalmOrder”. Tap “Settings”, choose *Exchange menu → Remote*.



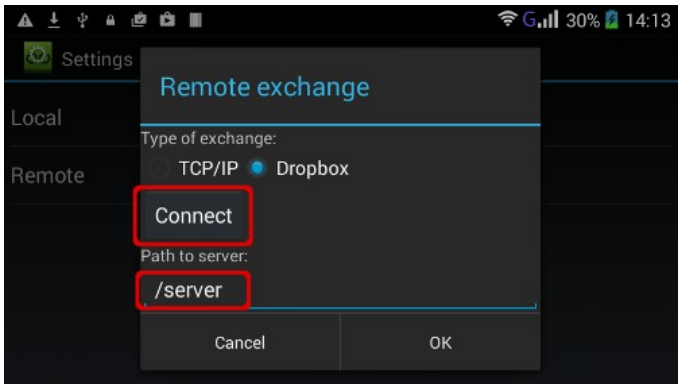
pic. 5.5 «Exchange menu window»

2. Select *Dropbox*.



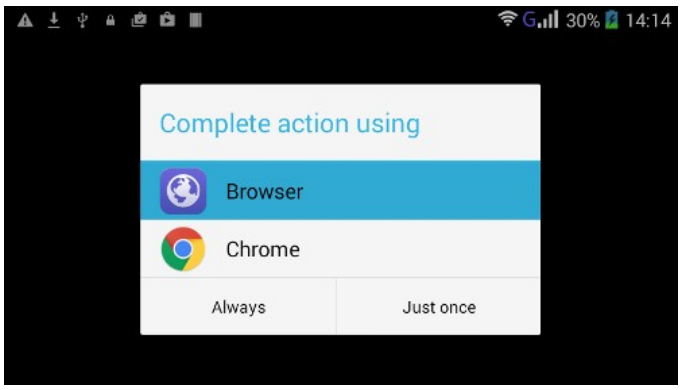
pic. 5.6 «Remote exchange window»

3. Enter **/Server** in the window “Path to server”
   Tap “Connect” (“Reconnect”, if you were connected earlier).



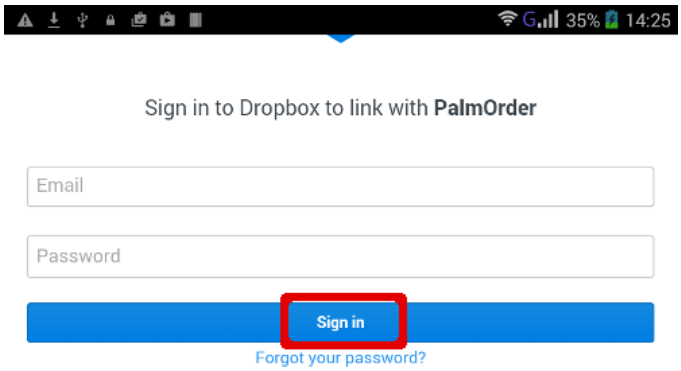
pic. 5.7 «Configuring connection for Dropbox exchange»

4. When the window “Complete action using” appears, choose the option that you prefer — as for us, we use Browser.



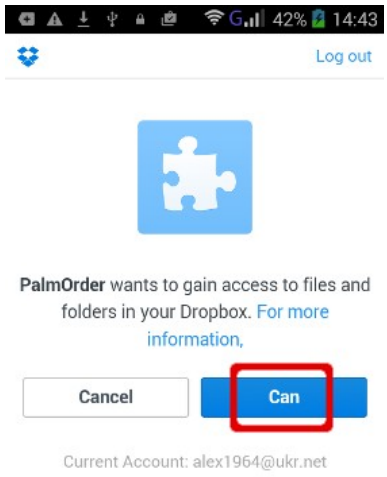
pic. 5.8 «"Complete action using" window»

5. Then there will be the window of connecting to Dropbox. Enter the e-mail address and password **that you use on Dropbox for this device**. Tap "Sign in".



pic. 5.9 «The window of connecting to Dropbox»

6. Allow PalmOrder to access the Dropbox shared folders by tapping "Can".



pic. 5.10

Now there is no need to start `palm_server.exe` to perform a remote exchange. But

if you need to carry out a local exchange, it's necessary to start `palm_serv.exe`. Make sure that you launch `c:\Dropbox\Server\palm_server.exe`, and not `c:\PalmOrder\Server\palm_serv.exe`. You better delete `c:\PalmOrder\Server\palm_serv.exe` to be certain. Nota bene: to install a license on a mobile device, you need to perform only a **local** exchange (or remote **TCP/IP)** exchange, so don't forget to start `c:\Dropbox\Server\palm_server.exe`. And be sure to exit `palm_server.exe` after the license is installed (or another local exchange is performed). **When palm_server.exe is executed locally to install the new licenses on mobile devices, the exchange via Dropbox does not function.**

If you use more than one device (license), you can provide the access to a Dropbox account in two ways:

1. If you had a Dropbox account set up for every device, just undo the sharing of the Server folder in your Dropbox account for the account of the mobile device you want to deny access to.

2. You can also use the same account for all devices (see step 5 above). In this case, the PalmOrder server checks the validity of a license in `palmserv.ini`, the way the data exchange is performed according to the TCP/IP protocol. You can ban a device by excluding the section for this device from `palmserv.ini`, or by installing a license on another (“new”) device. The exchange for the “old” device will be banned.

Exchange File

To perform data transferring between the mobile part and server, your PDA requires the exchange file – `exchange.xml`. A standard exchange file is included into a standard installation package.

The exchange rules are described in an XML-format. You can configure different exchange rules and choose a required one before initiating exchange. For example, you can create an exchange file containing three profiles:

- sending documents and files;
- receiving references updates;
- full exchange (sending and receiving in one session).

After that a created XML file is uploaded to your PDA (see “Downloading exchange.xml” for details).

The structure of an XML file to define the rules of exchange

```
<?xml version="1.0" encoding="windows-1251"?>
<palm-client>
   <exchange name="exchange name" view="representation of exchange">
      <post>
         <documents>
            <document name="document name" view="representation of document"
          tableName="name of main table of document">
                description of document
            </document>
            <document name="..." view="..." tableName="...">
               description of document
            </document>
         </documents>
         <files>
            <file name="file name" type="file type" delete="removal mark"/>
            <file name="..."/>
         </files>
      </post>
```

```
      <get>
         <dbs>
            <db name="db-file name"/>
            <db name="..."/>
         </dbs>
         <files>
            <file name="file name"/>
         </files>
      </get>
   </exchange>
   <exchange name="exchange name" view="representation of exchange">
      ........
   </exchange>
</palm-client>
```

Formation rules

The XML document of exchange parameters consists of blocks:

```
<exchange name="exchange name" view="representation of exchange">
         ...
</exchange>
```

which describe the rules of an exchange profile. Each block consists of:

- `<post>...</post>` – a description block of data sent from the mobile part to the server side of the system. It can contain such blocks: `<documents>...</documents>`, `<files>...</files>`.
- `<get>...</get>` – a description block of data received by the mobile part from a server. It can contain such blocks: `<dbs>…</dbs>`, `<files>...</files>`.

Transferred documents (`<document>`) are enclosed in tags `<documents>...</documents>`. Sent and received files (`<file>`) are enclosed in tags `<files>...</files>`. Received tables (`<db>`) of database, which make up the references, lists, documents etc., are enclosed in tags `<dbs>...</dbs>`.

The example of an exchange file can be found in `<InstallationFolder>\DataExamples\Exchange`.

The description of a transferred document

```
<document name="document name" view="representation of document"
          tableName="name of main table of document">
   <syncFields>
      <field name="field name" value="field value" set="field set"/>
   </syncFields>
   <docHead>
      <field name="field name"/>
      <field name="..."/>
   </docHead>
   <docTable tableName="table name of document tabular section">
      <field name="field name" />
      <field name="..." />
   </docTable>
</document>
```

The description of each document is enclosed in tags `<document name="document name" view="representation of document" tableName="name of main table of document">` and `</document>`.

The description of each document may consist of three blocks:

`<syncFields>...</syncFields>`	mandatory block
`<docHead>...</docHead>`	

`<docTable>...</docTable>`	optional blocks

In the block `<syncFields>…</syncFields>` we describe document flags (field values), which must be checked before sending to decide if this document should be sent.

```
<field name="field name" value="field value" set="field set"/>
```

where `name` – is the name of a field; `value` – is the value of a field that will get a document sent; `set` – is the value a field will take after a successful sending to avoid resending.

In the block `<docHead>...</docHead>` we describe transferred fields of document's Details. Each field is described by the element `<field name="field name"/>`.

In the block `<docTable>...</docTable>` we describe transferred fields of a document tabular section. It can be omitted if a document does not contain a tabular section. Each field is described by the element `<field name="field name"/>`.

The description of a transferred file

```
<file name="file name" type="file type" delete="removal mark" />
```

The name of a transferred file is specified in the attribute `name`, the type of file – in the attribute `type`. The server part will take the necessary actions with a file depending on its type. For example, if an `XML` type is specified, it means that this file must be processed in ERP or AIS. If a `JPG` type is specified, then no further action is required. If the attribute `delete` has a `true` value, a file will be deleted after successful sending.

The description of a database file

```
<db name="db-file name"/>
```

The name of a database file, which is requested from the server part, is specified in the attribute `name`.

The description of a received file

```
<file name="<name of the file>" view="<presentation>" files="<files>"
          destinationFolder="<saving folder>" storage="<storage>"
          delete="<deleting attribute>"/>
```

The attribute `name` indicates the name of a received file. The `view` attribute specifies the name that is displayed on your screen. The `files` attribute indicates which files to receive. It can be set by a mask, such as *. *. Next — `destinationFolder` — means a subfolder in the `Inbox` folder. If this attribute is not set, the files are saved to the `Inbox` folder. The `storage` attribute designates the storing place of the received files: `internal` — on the internal ROM, `external` — on the additional memory card. The value by default is `internal`. If you set the value `true` for the `delete` attribute, the original file will be deleted from your device after it is successfully received by a server.

The example of receiving photos:

```
<post>
 <files>
  <file name="photos" view="Photograph" files="photos/*.*"
          destinationFolder="" storage="external" delete="true" />
 </files>
</post>
```

The File of Remote Preferences for the Mobile Part

The file of preferences for the PalmOrder mobile part can be downloaded remotely. Just add this line to `exchange.xml`

```
<file name="preferences" files="preferences.xml" deleteOnServer="true" />
```

following the rules described in the section *Exchange File* (see above).

The structure of preferences.xml

```
<?xml version="1.0" encoding="utf-8"?>
<preferences>
   <section_1>
      <variable_1.1>value of variable_1.1</variable_1.1>
      <variable_1.2>value of variable_1.2</variable_1.2>
      ..........
      <variable_1.N>value of variable_1.N</variable_1.N>
   </section_1>
   .........
   <section_N>
      <variable_N.1>value of variable_N.1</variable_N.1>
      <variable_N.2>value of variable_N.2</variable_N.2>
      ..........
      <variable_N.N>value of variable_N.N</variable_N.N>
   </section_N>
</preferences>
```

The list of sections and the description of variables

№	Section/ Variable	Valid Values	Info
1	**`global`**		
1.1	`debug-mode`	`true, false`	Value `true` sets the application in a debug mode, `false` sets a regular mode
1.2	`theme`	`dark, light`	Value `dark` sets the Android Holo Dark theme, value `light` sets the Android Holo Light theme respectively
1.3	`table-font-size`	`small, normal, large, extra-large`	These values set a font size in table forms (references, journals, document's tabular sections)
2	**`references`**		
2.1	`back-hierarchy-up`	`true, false`	If `true` is set and hierarchy is enabled , you will move one level up through a hierarchy when pressing "Back." If `false` is set, you will leave a reference when pressing "Back"
2.2	`show-parents-in-details`	`true, false`	If `true` is set, in Details you will see item parent groups
3	**`product-reference-images`**		
3.1	`path`	String	The path to the folder with item images
3.2	`extensions`	String	The list of valid extensions of item image files. Values are separated by a semicolon: jpg; jpeg; gif; png; bmp
4	**`document-order`**		

4.1	calculate-rest	true, false	If `true` is set, the `Rest` field value decreases automatically, when `Qty` increases. And vice versa
4.2	set-custom-price	true, false	If `true` is set, the correction of price is possible
4.3	products-in-package	true, false	If `true` is set, you will enter the number of packages in the `Qty` field, if `false` is set — you will enter units
4.4	customer-own-prices-only	true, false	If `true` is set, only the category of prices, specified in the `Customers` reference, is allowed for a customer
4.5	forbid-new-document-with-expired-debt	true, false	If `true` is set, you are not allowed to take requests from a customer with an overdue debt
4.6	check-photo	true, false	If `true` is set, you cannot create a "Request" document until a photo is saved
5	**product-count-dialog-info-fields**		
5.1	name	true, false	If `true` is set, the field will be displayed in the "Additional data" section of the Quantity dialog box
5.2	code	true, false	If `true` is set, the field will be displayed in the "Additional data" section of the Quantity dialog box
5.3	<other field from the Items reference>	true, false	If `true` is set, the field will be displayed in the "Additional data" section of the Quantity dialog box
6	**document-payment**		
6.1	forbid-edit-saved	true, false	If `true` is set, you are denied to correct saved documents
6.2	forbid-delete-not-sent	true, false	If `true` is set, you are denied to delete unsent documents
7.1	**locale-exchange**		
7.1.1	host	String	An IP-address or domain name used in the "Local" exchange
7.1.2	port	1 - 65535	This field value must match the value tcpport of palmserv.ini
7.1.3	time-out	Integer value > 0	The time of waiting for the response of the PalmOrder server part in seconds
7.2	**remote-exchange**		
7.2.1	type	tcpip, dropbox	Sets the type of the remote exchange
7.2.2	host	String	An IP-address or domain name used in the "Remote" TCP/IP exchange
7.2.3	dropbox-server-path	String	The path to a folder that you shared in Dropbox settings (usually /Server) in case of using the exchange via Dropbox
7.2.4	port	1 - 65535	This field value must match the value tcpport of palmserv.ini

7.2.5	time-out	Integer value > 0	The time of waiting for the response of the PalmOrder server part in seconds
8	**gps-tracking**		
8.1	enabled	true, false	The value true turns GPS-online on
8.2	required	true, false	If true is set, the application requires a GPS module to be turned on
8.3	interval	Integer value>0	The interval (in seconds) of fixing coordinates
8.4	on	Time in hh.mm	The time when GPS-online starts
8.5	off	Time in hh.mm	The time when GPS-online stops
8.6	send-to-server	true, false	Starts the mode "Send points to GPS-server"
8.7	login	String	Login on gps.palmorder.com
8.8	password	String	Password on gps.palmorder.com
8.9	send-interval	Integer value>0	The interval of sending coordinates to a GPS-server. The lesser the interval, the higher the battery consumption of a device
8.10	service-check-interval	Integer value>0	Sets the interval (in minutes) of checking if a GPS module is turned on

The example of preferences.xml with default settings:

```
<?xml version="1.0" encoding="utf-8"?>
<preferences>
   <global>
      <debug-mode>false</debug-mode>
      <theme>dark</theme>
      <table-font-size>normal</table-font-size>
   </global>
   <references>
      <back-hierarchy-up>false</back-hierarchy-up>
      <show-parents-in-details>false</show-parents-in-details>
   </references>
   <product-reference-images>
      <path>/mnt/sdcard/Android/data/com.trukom.erp/files/images/products</pa
         th>
      <extensions>jpg; jpeg; gif; png; bmp</extensions>
   </product-reference-images>
   <document-order>
      <calculate-rest>false</calculate-rest>
      <set-custom-price>false </set-custom-price>
      <products-in-package>false</products-in-package>
      <customer-own-prices-only>false</customer-own-prices-only>
      <forbid-new-document-with-expired-debt>false</forbid-new-document-with-
          expired-debt>
      <check-photo>false</check-photo>
   </document-order>
   <product-count-dialog-info-fields>
      <name>false</name>
      <code>false</code>
      <color>true</color>
      <rest>false</rest>
```

```
      <packaging>false</packaging>
      <ID>false</ID>
   </product-count-dialog-info-fields>
   <document-payment>
      <forbid-edit-saved>false</forbid-edit-saved>
      <forbid-delete-not-sent>false</forbid-delete-not-sent>
   </document-payment>
   <locale-exchange>
      <host>192.168.0.1</host>
      <port>8888</port>
      <time-out>60</time-out>
   </locale-exchange>
   <remote-exchange>
      <type>dropbox</type>
      <dropbox-server-path>/Server</dropbox-server-path>
      <time-out>60</time-out>
   </remote-exchange>
   <gps-tracking>
      <enabled>false</enabled>
      <required>false</required>
      <interval>60</interval>
      <on>00:00</on>
      <off>00:00</off>
      <send-to-server>true</send-to-server>
      <login>yourlogin</login>
      <password>yourpassword</password>
      <send-interval>60</send-interval>
      <service-check-interval>60</service-check-interval>
   </gps-tracking>
</preferences>
```

File with the Remote Settings for the Table Columns

The settings for all tables, such as references, journals, tabular sections of the "Request", "Store-check" and "Debt" documents, can also be downloaded and installed with a remote exchange. To do this, you need to create the file `tables_columns.xml` and copy it to the folder `Outbox`.

Add the string:

```
<file name="tables_columns" files="tables_columns.xml" />
```

to the `exchange.xml` following the rules, described in the *Exchange File* chapter (see above). For the Item Selection Table you need to assign the value `multiselect` to the `layout` attribute. The value in the `position` field is set automatically. Number 1 is attributed to the column next to the tag `<table name="table_nameN">`, number 2 – to the next column and so on. Be sure to keep this in mind when creating `tables_columns.xml.`

After downloading the file on your mobile device, execute a "Tabular section reset" (in our case – for Items reference) to initialize the new settings: tap "Settings", choose "Tabular section reset", enable the "Items reference" check-box and tap "OK".

The structure of tables_columns.xml

```
<?xml version="1.0" encoding="utf-8"?>
<tables-columns>
   <table name="tabular_section1">
      <column name="column1">
         description of column1
      </column>
      ........
```

```
      <column name="columnN">
         description of columnN
      </column>
   </table>
   <table name="tabular_sectionN">
      <column name="column1">
         description of column1
      </column>
      .........
      <column name="columnN">
         description of columnN
      </column>
   </table>
</tables-columns>
```

For Item Selection Table you need to assign `multiselect` value to `layout` attribute, the syntax is the same:

```
   <table name="ref_products" layout="multiselect">
      <column name="column1">
         description of column1
      </column>
      ........
      <column name="columnN">
         description of columnN
      </column>
   </table>
```

List of tables and table columns:

Table / Column	Comment
ref_products (Items reference)	
_id	Item identifier (primary key)
code	Item code
name	Item name
rest	Units left in stock
price_Retail	Value of the retail price
price_Trade	Value of the trade price
price_Wholesale	Value of the wholesale price
packaging	Item packaging (units in a package).
parent	The code of a parent group.
group	Indicates whether an element is a group.
color	The color of a reference element.
measure_unit	
ref_customers (Customers reference)	
_id	Customer identifier (primary key)
code	Customer code

name	Customer name
catprice	The category of price set for a customer by default.
rest	A total debt of a customer
address	Post address of the customer
phone	Phone number of the customer
parent	The code of a parent group.
group	Indicates whether an element is a group.
color	The color of a reference element.
expired_debt	The overdue debt of the customer
routes	
route_id	Identifier of the route
_id	Identifier of the customer
code	Customer code
name	Customer name
catprice	Category of price
rest	Total debt
address	Customer's post address
phone	Customer's phone number
parent	The code of a parent group.
group	Indicates whether an element is a group.
color	The color of a reference element.
expired_debt	The overdue debt
doc_exists	The sign indicates that any document was created for this customer
trade_point_value	
trade_point	
doc_orders (Request Journal)	
_id	Identifier of the document
date_time	Date and time of the creation of the document
number	Number of the document
save_date_time	Date and time of saving
customer_name	Customer name
customer_code	Customer code
trade_point	Point of sale of the customer
delivery	Date of delivery

sum	Amount
send_status	Status of sending
pay_type	Type of payment
longitude	GPS longitude of place of creation
latitude	GPS latitude of place of creation
info	Notes
discount	Discounted amount
delivery_type	Type of delivery
sum_no_discount	Amount before discounting
cat_price	Category of price set in the document
doc_invoices (Payments Journal)	
_id	Identifier of the document
date_time	Date and time of the creation of the document
number	Number of the document
save_date_time	Date and time of saving
customer_name	Customer name
customer_code	Customer code
sum	Amount
send_status	Status of sending
longitude	GPS longitude of place of creation
latitude	GPS latitude of place of creation
info	Notes
base_doc	Base document
doc_storechecks (Storechecks Journal)	
_id	Identifier of the document
date_time	Date and time of the creation of the document
number	Number of the document
save_date_time	Date and time of saving
customer_name	Customer name
customer_code	Customer code
trade_point	Point of sale of the customer
send_status	Status of sending
longitude	GPS longitude of place of creation
latitude	GPS latitude of place of creation

info	Notes
doc_debts (Debts Journal)	
_id	Identifier of the document
date_time	Date and time of the creation of the document
number	Number of the document
save_date_time	Date and time of saving
customer_name	Customer name
customer_code	Customer code
sum	Amount
info	Notes
trade_point	Point of sale of the customer
cat_price	Category of price set in the document
pay_type	Type of payment
delivery	Date of delivery
delivery_type	Type of delivery
doc_orders_products (Request document tabular section)	
_id	Item identifier (primary key)
doc_id	Document identifier
sum_no_discount	Amount before discounting
sum	Discounted amount
price_discount	Discounted price
price	Price before discounting
num	Line number
name	Item name
koef	Multiplier (units in package)
discount	Amount of discount
count	Quantity
code	Item code
ref_products (layout="multiselect") (item Selection Table for Request document)	
_id	Item identifier (primary key)
code	Item code
name	Item name
rest	Units left in stock
price_Retail	Retail price

price_Trade	Trade price
price_Wholesale	Wholesale price
packaging	Packaging (units in package)
parent	The code of a parent group. If a parent group is not present, the value is "" (an empty string)
group	Indicates whether an element is a group. If value is 0, an element is not a group; otherwise it is a group
color	The color of a reference element.
doc_orders_products_count	Current number of units
doc_orders_products_koef	Multiplier (units in package)
doc_orders_products_price	Current price
doc_orders_products_id	Request document identifier
cpd_price	Special price of the item for the customer
cpd_preorder_count	Number of units of the item in previous Request
doc_storechecks_products (Store-check tabular section)	
_id	Item identifier (primary key)
doc_id	Document identifier
price	Price
num	Number of row
name	Item name
count	Number of units
code	Item code
doc_debts_products (Debt tabular section)	
_id	Item identifier (primary key)
doc_id	Document identifier
num	Number of row
code	Item code
name	Item name
price	Price
count	Quantity
koef	Multiplier (units in package)
discount	Amount of discount
sum	Amount

Every column is described by eight blocks:

Variable	Type	Description
<display_name>...</display_name>	TEXT	Column name displayed
<align>...</align>	INTEGER	Alignment: 0-left, 1-center, 2-right
<visible>...</visible>	BOOLEAN	Hide/show
<width>...</width>	INTEGER	Width
<stretchable>...</stretchable>	BOOLEAN	Stretchable/not
<wrap_text>...</wrap_text>	BOOLEAN	Wrap text/not
<from_new_line>...</from_new_line>	BOOLEAN	Move this value to the new line
<type>...</type>	INTEGER	Data type: 0-text, 1-integer, 2-real, 3-date
<decimal_scale>...</decimal_scale>	INTEGER	Additional block: the number of decimals for the «Real» type fields

The example of `tables_columns.xml` (settings for the `name` column of `ref_products.xml` (Items reference)):

```
<?xml version="1.0" encoding="utf-8"?>
<tables-columns>
   <table name="ref_products">
      <column name="name">
         <display_name>Name</display_name>
         <!-- 0-left, 1-center, 2-right -->
         <align>0</align>
         <visible>true</visible>
         <width>200</width>
         <stretchable>true</stretchable>
         <wrap_text>false</wrap_text>
         <from_new_line>false</from_new_line>
         <!-- 0-text, 1-integer, 2-real, 3-date -->
         <type>0</type>
      </column>
   <table>
</tables-columns>
```

The value of `position` field is set automatically. The number 1 is set to the column next to tag `<table name="table_nameN">`, the number 2 – to the next column and so on. Keep in mind this, when create `tables_columns.xml`.

If the field values have the «Real» type, the additional block `<decimal_scale>...</decimal_scale>` is available. Here you can set the number of decimals.

After downloading the file on your mobile device, execute “Tabular section reset” (in our case – for Items reference) to initialize new settings: tap “Settings”, choose “Tabular section reset”, enable “Items reference” check-box and tap “OK”.

PalmOrder Database

The PalmOrder mobile part uses a relational database. Its structure is defined by the set of tables, lists and documents. When creating the XML documents for data exchange between the server and mobile parts of the system, you must adhere to the rules of name formation for references, lists, documents and names of fields, i.e. for a correct identification of a specific reference, document, field etc., its tabular section must have a specific name with a specific set of fields.

The documents and lists of the PalmOrder mobile part have a clearly defined set of fields, while the references may have additional optional fields formed during the download from AIS (ERP).

Below you will find the information about the PalmOrder mobile part references, lists, documents, fields of tabular sections, as well as their names and the rules of formation.

Fields of Tabular Sections

You should strictly adhere to the following rules of forming names and types of tabular section fields.

A tabular section must contain a set of obligatory fields and may contain optional information fields. Obligatory fields are used in the business logic of the system, and their presence is checked during the download of a tabular section. Optional information fields are commonly used to provide additional information for a mobile employee, for example, customer's address, phone etc.

Field names

Each field of a tabular section has its name. Every field name of the same tabular section of a reference or document must be unique, but may be repeated in tabular sections of different documents. For example, the field `name` can be present in every reference and document for an item identification.

For obligatory fields you should use predefined names. Other fields can have any name containing letters (Latin and Cyrillic) and digits. Name must begin with a letter, not a digit.

Field types

You can use these types of fields in the PalmOrder tabular sections:

№	Type	Meaning	Size(bytes)	Info
1	`INTEGER`	Integer digit	1-8	Is used to store integer values. Size depends on value
2	`INTEGER`	Date and time	4	Is used to store date and time. When downloading from / uploading to AIS (ERP), date is represented as the number of milliseconds starting from 00:00 January 01, 1970
4	`REAL`	Real digit	8	Is used to store real values
5	`TEXT`	String	-	A variable-length string
6	`BLOB`	binary	-	A variable-length binary field

Any field can be indexed. To create an index, you have to add "KEY" at the end of a field type value, e.g. `type="INTEGER KEY"`.

All tables contain the obligatory field `_id` (the synonym is `id`), which is a primary table

key. When creating a table, you can omit this field, because it is created automatically by `Xml2Eldb.exe`. However, when creating a document tabular section, you should use this field, since it binds the Details and a tabular section of a document.

Lists

List is a set of values used in the PalmOrder mobile part for choosing a desired value from a drop-down list. In its structure a list is similar to a reference, but unlike it a list contains only three fields – a primary key, value and performance. When forming the XML files to download data from AIS or ERP, use the following names for lists and its fields:

Name	Description
`vl_pay_types`	The list of payment types. Is used to select the type of payment in the “Request” and “Debt” documents
`vl_cat_prices`	The list of price categories. Is used to select the category of price in the “Request” and “Debt” documents
`vl_delivery_types`	The list of delivery types. Is used to select the type of delivery (carriage) in the “Request” and “Debt” documents
`vl_trade_points`	The list of points of sale. Is used to select a point of sale in the “Request” and “Debt” documents. This list contains one more obligatory field `customer_code`, which shows the code of a customer who owns a point of sale

The description of list fields:

Field	Type	Description
`_id`	`INTEGER`	A primary key of list. A valid value is >0, is sorted in an ascending order, is created automatically
`value`	`TEXT`	The value of a list record. Must be unique. Is stored in the document Details
`presentation`	`TEXT`	The presentation of a list record. It shows how value is displayed on the PDA's screen

The example of an XML file:

```
<?xml version="1.0" encoding="windows-1251"?>
<!-- List of price categories -->
<table name="vl_cat_prices">
   <schema>
      <field name="value" type="TEXT" />
      <field name="presentation" type="TEXT" />
   </schema>
   <data>
      <record>
         <f>Standard</f>
         <f>Standard price</f>
      </record>
      <record>
         <f>Dealer</f>
         <f>Price for Dealers</f>
      </record>
      <record>
         <f>Distributor</f>
         <f>Price for Distributors</f>
      </record>
   </data>
</table>
```

References

When creating the XML files to download data from / upload to AIS or ERP, use the following references:

Reference Name	Reference Description
ref_products	Item reference
ref_customers	Customer reference
customer_product_data	The table of special data on a specific item for a definite customer, for example, price, last requested quantity etc.
routes	The table of routes

There is a set of fields that must be present in all references, and there are fields unique for each reference.

The description of fields for ref_customers:

Field	Type	KEY	Oblig-atory	Description
`_id`	`INTEGER`	+	+	A primary key of reference. A valid value is >0, is sorted in an ascending order, the first field in a table
`code`	`TEXT`	+	+	Corresponds to the field `code` in AIS (ERP). The value must be unique
`name`	`TEXT`	-	+	The name of a reference item
`cat_price`	`TEXT`	-	+	The price categories set for a customer by default. This field value must be equal to one of the values of `vl_cat_prices`. (`ref_customers.cat_price=vl_cat_prices.value`)
`rest`	`REAL`	-	+	A total debt of a customer
`group`	`INTEGER`	-	-	Indicates whether an element is a group. If value is 0, an element is not a group; otherwise it is a group
`parent`	`TEXT`	+	-	The code of a parent group. If a parent group is not present, the value is "" (an empty string)
`color`	`INTEGER`	-	-	The color of a reference element. Colors are downloaded in an RGB format, for example, red is `0xFF0000`, blue is `0x0000FF` and so on. You can use different colors to indicate different features of reference elements. For example: heavily indebted customers can be highlighted with red and so on

The example of a table in an XML file of Customers reference:

```
<?xml version="1.0" encoding="windows-1251"?>
<!-- Customers reference -->
<table name="ref_customers">
   <schema>
      <field name="code" type="TEXT KEY" />
      <field name="name" type="TEXT" />
      <field name="cat_price" type="TEXT" />
      <field name="rest" type="REAL" />
      <field name="license_date" type="INTEGER" />
      <field name="contract_date" type="INTEGER" />
```

```
      <field name="parent" type="TEXT KEY" />
      <field name="group" type="INTEGER KEY" />
   </schema>
</table>
```

The description of fields in ref_products:

Field	Type	KEY	Obligatory	Description
_id	INTEGER	+	+	A primary key of reference. A valid value is >0, is sorted in an ascending order, the first field in a table
code	TEXT	+	+	Corresponds to the field code in AIS (ERP). The value must be unique
name	TEXT	-	+	The name of a reference item
rest	REAL	-	+	Units left in stock
price_cat1, price_cat2, ...	REAL	-	+	The price categories. cat1, cat2 are field values from the list vl_cat_prices. Example: if there's the record value="Standard" in vl_cat_prices, then there must be the field price_Standard in ref_products
packaging	REAL	-	+	Item packaging (units in a package). You can choose packaging to make it easier and faster to set quantity in different documents
group	INTEGER	-	-	Indicates whether an element is a group. If value is 0, an element is not a group; otherwise it is a group
parent	TEXT	+	-	The code of a parent group. If a parent group is not present, the value is "" (an empty string)
color	INTEGER	-	-	The color of a reference element. Colors are downloaded in an RGB format, for example, red is 0xFF0000, blue is 0x0000FF and so on. You can use different colors to indicate different features of reference elements. For example: promotional items can be highlighted with blue etc.

The example of a table in an XML file of Items reference:

```
<?xml version="1.0" encoding="windows-1251"?>
<!-- Items reference -->
<table name="ref_products">
   <schema>
      <field name="code" type="TEXT KEY" />
      <field name="name" type="TEXT" />
      <field name="rest" type="INTEGER" />
      <field name="price_Standard" type="REAL" />
      <field name="price_Dealer" type="REAL" />
      <field name="price_Distributor" type="REAL" />
      <field name="unit" type="TEXT" />
      <field name="parent" type="TEXT KEY" />
      <field name="group" type="INTEGER KEY" />
      <field name="type" type="INTEGER" />
      <field name="min_count" type="INTEGER" />
   </schema>
</table>
```

The description of fields in customer_product_data (individual prices, previous requested quantity etc):

Field	Type	KEY	Obligatory	Description
_id	INTEGER	+	+	A primary key of reference. A valid value is >0, is sorted in an ascending order, the first field in a table
customer_code	TEXT	+	+	The code of a customer. customer_product_data.customer_code = ref_customers.code
product_code	TEXT	+	+	The code of an item. customer_product_data.product_code = ref_products.code
price	REAL	-	-	An individual customer's price for a specific item

The example of a table in an XML file of the reference:

```
<?xml version="1.0" encoding="windows-1251"?>
<!-- Individual customers price for specific item -->
<table name="customer_product_data">
   <schema>
      <field name="customer_code" type="TEXT KEY" />
      <field name="product_code" type="TEXT KEY" />
      <field name="price" type="REAL" />
      <field name="preorder_count" type="REAL" />
   </schema>
</table>
```

The description of fields in routes:

Field	Type	KEY	Obligatory	Description
_id	INTEGER	+	+	A primary key of reference. A valid value is >0, is sorted in an ascending order, the first field in a table
customer_code	TEXT	-	+	The code of a customer. routes.customer_code = ref_customers.code
trade_point_value	TEXT	-	-	The code of a point of sale. routes.trade_point_value= vl_trade_points.value
week_day	INTEGER	+	-	A week day. Week days for routes: 1 is for Monday, 2 is for Tuesday and so on. If there is the field date, a route for week days is ignored
date	INTEGER	+	-	The date of a route in the format YYYY-MM-DD

The example of a table in an XML file:

```
<?xml version="1.0" encoding="windows-1251"?>
<!-- Routes -->
<table name="routes">
   <schema>
      <field name="customer_code" type="TEXT" />
      <field name="trade_point_value" type="TEXT" />
      <field name="week_day" type="INTEGER KEY" />
   </schema>
</table>
```

Optional fields

In addition to these obligatory fields in references, you can download any number of optional fields of any type. For example, you can download a customer's address and phone number to `ref_customers`.

Documents

We use the following documents in the mobile part of PalmOrder: Pre-Selling:

System Name	Document Name
order	Document "Request"
invoice	Document "Payment"
debt	Document "Debt"
storecheck	Document "Store-Check"
Custom table	The table, created by the user

The documents “Request”, “Payment” and “Store-Check” are formed by mobile employees on their PDAs. “Debt” and “Custom table” are downloaded from AIS (ERP).

All documents, except “Payment” and “Custom table”, consist of two tabs. A tabular section is absent in “Payment”.

The rules of forming the tables of a document:

doc_<DocumentName>s	section Details of a document
doc_<DocumentName>s_products	A tabular section of a document that contains items

`<DocumentName>` is the name of a specific document. For example, Details and tabular section (Items) of the document `order` are called `doc_orders` and `doc_orders_products` respectively. A tabular section of the document is tied to Details section by the fields `doc_<DocumentName>._id = doc_<DocumentName>s_products.doc_id`.

Document “Request”

The fields of Details section (doc_orders):

Field	Type and KEY	Description
_id	INTEGER	A primary table key of the Details. A valid value is >0, is sorted in an ascending order, the first field in a table
date_time	INTEGER	The date and time when a document was created. Is assigned by the system automatically when a document is created
save_date_time	INTEGER	The date and time when a document was saved. Is assigned by the system automatically when a document is saved / re-saved
number	TEXT	A serial number of a document. Is assigned by the system automatically in the format Rq-xxxx
customer_code	TEXT KEY	The code of a customer from Customers reference. doc_orders.customer_code = ref_customers.code
customer_name	TEXT	The name of a customer from Customers reference doc_orders.customer_name= ref_customers.name
info	TEXT	A comment as a text of variable length

send_status	INTEGER	The status of sending to AIS (ERP). Status equals 0 when a document is created, and 1 when it is successfully sent
sum	REAL	A total amount of a document. Is equal to a total sum of all requested items. Discount is included
sum_no_discount	REAL	A total sum without discount
discount	REAL	The amount of discount
trade_point	TEXT	Is equal to the value of a selected point of sale. doc_orders.trade_point = vl_trade_points.value
cat_price	TEXT	Is equal to a selected price category. doc_orders.cat_price = vl_cat_price.value
pay_type	TEXT	Is equal to a selected type of payment. doc_orders.pay_type = vl_pay_types.value
delivery	INTEGER	The date of an expected delivery to a customer
delivery_type	TEXT	Is equal to a selected type of delivery. doc_orders.delivery_type = vl_delivery_types.value

A tabular section of “Request” document (doc_orders_products):

Field	Type and KEY	Description
_id	INTEGER	A primary table key of document's tabular section. A valid value is >0, is sorted in an ascending order, the first field in a table
doc_id	INTEGERKEY	The field tied to Details section. doc_orders_products.doc_id = doc_orders._id
num	INTEGER	A serial number of a position in a tabular section. Must be unique for a document
code	TEXT	The code of an item in Items reference. doc_orders_products.code = ref_products.code
name	TEXT	The name of an item in Items reference. doc_orders_products.name = ref_products.name
price	REAL	An item price. Depends on a price category selected in Details section of a document
price_discount	REAL	An item price with discount included
count	REAL	The number of items or packages, if “Of packages” is enebled
koef	REAL	If “Of packages” is enabled, this field contains the number of units in a package, and is equal to the value of the field packaging for this item in Items reference. In other cases the value is 1
discount	REAL	Discount percentage for an item
sum_no_discount	REAL	A total amount without discount
sum	REAL	A total amount for an item. Is equal to price multiplied by quantity multiplied by coefficient (if "Of packages" is enabled) minus discount

Document "Payment"

The fields of “Payment” document (doc_invoices):

Field	Type and KEY	Description
_id	INTEGER	A primary table key of the Details. A valid value is >0, is sorted in an ascending order, the first field in a table
date_time	INTEGER	The date and time when a document was created. Is assigned by the system automatically when a document is created
save_date_time	INTEGER	The date and time when a document was saved. Is assigned by the system automatically when a document is saved / re-saved
number	TEXT	A serial number of a document. Is assigned by the system automatically in the format Pt-xxxx
customer_code	TEXT KEY	The code of a customer from Customers reference. doc_invoices.customer_code = ref_customers.code
customer_name	TEXT	The name of a customer from Customers reference. doc_invoices.customer_name= ref_customers.name
sum	REAL	A total amount of a document
send_status	INTEGER	The status of sending to AIS (ERP). Status equals 0 when a document is created, and 1 when it is successfully sent
info	TEXT	A comment as a text of variable length
base_doc	TEXT	The number of a document which is a basis for this document. doc_invoices.base_doc = doc_<DocumentName>s.number

Document "Store-Check"

The fields of Details section of a document (doc_storechecks):

Field	Type and KEY	Description
_id	INTEGER	A primary table key of the Details section. A valid value is >0, is sorted in an ascending order, the first field in a table
date_time	INTEGER	The date and time when a document was created. Is assigned by the system automatically when a document is created
save_date_time	INTEGER	The date and time when a document was saved. Is assigned by the system automatically when a document is saved / re-saved
number	TEXT	A serial number of a document. Is assigned by the system automatically in the format Sc-xxxx
customer_code	TEXT KEY	The code of a customer in Customers reference. doc_storechecks.customer_code = ref_customers.code
customer_name	TEXT	The name of a customer in Customers reference. doc_storechecks.customer_name= ref_customers.name
info	TEXT	A comment as a text of a variable length
send_status	INTEGER	The status of sending to AIS (ERP). Status equals 0 when a document is created, and 1 when it is successfully sent

trade_point	TEXT	Is equal to a selected point of sale. doc_storechecks.trade_point = vl_trade_points.value

A tabular section of a "Store-Check" document (doc_storechecks_products):

Field	Type and KEY	Description
_id	INTEGER	A primary key of a tabular section. A valid value is >0, is sorted in an ascending order, the first field in a table
doc_id	INTEGERKEY	The field tied to Details section. doc_storechecks_products.doc_id = doc_storechecks._id
num	INTEGER	A serial number of a position in a tabular section. Must be unique for a document
code	TEXT	The code of an item in Items reference. doc_storechecks_products.code = ref_products.code
name	TEXT	The name of an item in Items reference. doc_storechecks_products.name = ref_products.name
price	REAL	An item price. Depends on a price category selected in Details section of a document
count	REAL	The number of items or packages, if “Of packages” is enabled

Document “Debt”

The fields of a requisites section of a document (doc_debts):

Field	Type and KEY	Description
_id	INTEGER	A primary key of the Details section. A valid value is >0, is sorted in an ascending order, the first field in a table
date_time	INTEGER	The date and time when a document was created. Is assigned by the system automatically
number	TEXT	A serial number of a document. Is assigned by the system automatically in the format Inv-xxxx
customer_code	TEXT KEY	The code of a customer in Customers reference. doc_debts.customer_code = ref_customers.code
customer_name	TEXT	The name of a customer in Customers reference. doc_debts.customer_name = ref_customers.name
sum	REAL	A total sum of a document
info	TEXT	A comment as a text of variable length
trade_point	TEXT	Is equal to a selected point of sale. doc_debts.trade_point = vl_trade_points.value
cat_price	TEXT	Is equal to a selected price category. doc_debts.cat_price = vl_cat_price.value
pay_type	TEXT	Is equal to a selected type of payment. doc_debts.pay_type = vl_pay_types.value
delivery_type	TEXT	Is equal to a selected type of delivery. doc_debts.delivery_type=

		vl_delivery_types.value

A tabular section of “Debt” document:

Field	Type and KEY	Description
_id	INTEGER	A primary key of a tabular section. A valid value is >0, is sorted in an ascending order, the first field in a table
doc_id	INTEGER KEY	The field tied to Details section. doc_debts_products.doc_id = doc_debts._id
num	INTEGER	A serial number of a position in a tabular section. Must be unique for a document
code	TEXT	The code of an item in Items reference. doc_debts_products.code = ref_products.code
name	TEXT	The name of an item in Items reference. doc_debts_products.name = ref_products.name
price	REAL	An item price. Depends on a price category selected in Details section of a document
count	REAL	The number of items or packages, if “Of packages” is enabled
koef	REAL	If “Of packages” is enabled, this field contains the number of units in a package, and is equal to the value of the field packaging for this item in Items reference. In other cases the value is 1
discount	REAL	Discount percent for an item
sum	REAL	A total amount for an item

The example of a table scheme for "Debts" documents

doc_debts

```
<?xml version="1.0" encoding="windows-1251"?>
<!-- Debts details -->
<table name="doc_debts">
   <schema>
      <field name="_id" type="INTEGER" />
      <field name="date_time" type="INTEGER" />
      <field name="number" type="TEXT" />
      <field name="customer_code" type="TEXT KEY" />
      <field name="customer_name" type="TEXT" />
      <field name="sum" type="REAL" />
      <field name="info" type="TEXT" />
      <field name="trade_point" type="TEXT" />
      <field name="cat_price" type="TEXT" />
      <field name="pay_type" type="TEXT" />
      <field name="delivery_type" type="TEXT" />
   </schema>
</table>
```

doc_debts_products

```
<?xml version="1.0" encoding="windows-1251"?>
<table name="doc_debts_products">
   <schema>
      <field name="_id" type="INTEGER KEY" />
      <field name="doc_id" type="INTEGER KEY" />
      <field name="num" type="INTEGER" />
```

```
      <field name="code" type="TEXT KEY" />
      <field name="name" type="TEXT" />
      <field name="price" type="REAL" />
      <field name="count" type="REAL" />
      <field name="koef" type="REAL" />
      <field name="discount" type="REAL" />
      <field name="sum" type="REAL" />
   </schema>
</table>
```

Custom Tables

PalmOrder allows you to create and download the tables for arbitrary data on your mobile device. To use this feature, perform the following steps:

- create a table in an XML format and convert it into a DB format;
- create a `custom_tables.xml` file;
- edit your `exchange.xml` file to be able to download custom tables

The rules of forming a custom table

The structure of a custom table is formed according to the rules described in chapter *The Format of Data Downloaded from AIS (ERP) (References, Lists, Documents)*.

You can create an unlimited number of custom tables.

The conversion of custom tables is performed using the Xml2Eldb.exe utility.

A service name must be unique and cannot be repeated. You also need to create a file with the column settings for a tabular section as described in chapter *File with the Remote Settings for the Table Columns*.

The custom_tables.xml file

In this file you describe the names of custom tables that are displayed on the screen. The tags `<name>...</name>` contain a service name of a table, and `<label> ...</label>` — the name that is displayed on the screen.

To download custom tables on your mobile device, you need to edit your `exchange.xml:` to the subsection `<dbs>` of the section `<get>` add the string

```
<db name="ct_tableN"/>
```

for each custom table.

You also need to add an instruction for downloading the `custom_tables.xml` file. The subsection `<files>` of the section `<get>` must contain the string

```
<file name="custom_tables" view="Custom tables" files="custom_tables.xml"/>
```

Printing

PalmOrder makes it possible to print documents via BlueTooth. The printing settings include:

- creating a `print_forms.xml` file;
- creating a print forms;
- editing `exchange.xml` to download print forms and `print_forms.xml`;
- selecting printer and editing printing requisites (details).

print_forms.xml File

The `print_forms.xml` file is an XML-document, and is formed according to the rules described in a corresponding chapter. The containing data are enclosed in tags `<print_forms>` and `</print_forms>.`

The file consists of blocks, that describe the print forms used for a specific document. The blocks are enclosed in tags this way:

```
<document name="name of a document">
        ...
</document>
```

The print forms and their presentations are listed inside a block. They are enclosed in tags `<forms>` and `</forms>`. It looks like this:

```
<forms>
   <form>
      <presentation>Name of a printed document</presentation>
      <file>name_of_print_form.prn</file>
   </form>
</forms>
```

Here's an example of `print_forms.xml` with two print forms for “Request” and one for “Payment”:

```
<print-forms>
   <document name="order">
      <forms>
         <form>
            <presentation>Invoice</presentation>
            <file>invoice.prn</file>
         </form>
         <form>
            <presentation>Delivery note</presentation>
            <file>delivery_note.prn</file>
         </form>
      </forms>
   </document>
   <document name="payment">
      <forms>
         <form>
            <presentation>Ticket</presentation>
            <file>ticket.prn</file>
         </form>
      </forms>
   </document>
</print-forms>
```

How to Create a Print Form

A print form is a text file that contains some text and a set of printing instructions (commands) that are sent to printer.

For example, a print form of a delivery note for printing on Epson LX-300 looks like this:

```
{% escp.esc 2 %}{% escp.12 %}
Supplier:  {$we.company} Address: {$we.address}
Reg. №: {$we.registration}
Details for payment: {$we.billing}

Customer:    {$doc.customer.name}
Address:     {$doc.customer.PostAddress}
{% escp.esc E %}     DELIVERY NOTE № {$doc.number} from {$doc.date}{% escp.esc
          F%} {% escp.esc 0 %}

          +-------------------------------------------------------------------
          --------+
  |  №  |         Item             |Unit of| Units   |   Per     |   Amount
          |
  |     |                          |measure|         |   unit    |
          |

          +-------------------------------------------------------------------
          --------+
{% loop in $table as $row counter=$num %}
  | {$num|qcalc(+1)|sprintf(%-4s)}| {$row.name|left(25)|sprintf(%-25s)}|
          {$row.product.measure_unit|left(4)|sprintf(%4s)} | {$row.count|
          sprintf(%8.2f)}|{$row.price|sprintf(%11.2f)} | {$row.sum|
          sprintf(%12.2f)}|
{% endloop %}

          +-------------------------------------------------------------------
          --------+
{% escp.esc 2 %}

Total amount: {$doc.sum|sprintf(%.2f)} USD.

  Sent by_____________________      Received by _____________________

{% escp.0C %}
```

Let's review the form line by line:

- The command `{% escp.esc 2 %}` selects a 1/6-inch line spacing; the command `{% escp.12 %}` cancels condensed printing (in case if it was turned on in a previous document).
- We print the word `Supplier`, after that comes the variable `we.company`, then we print the word `Address` and the variable `we.address.`
- Next line is `Reg. №`, the variable `we.registration.`
- Then we print the phrase `Details for payment` and the corresponding variable `we.billing`. The variables `we.*` are entered by a user when setting printing parameters. Instead of the variables you can specify the desired details. For example, instead of `we.billing`, enter the text: “Acc. № 2600111222333 in JP Morgan Chase, NY”. But if your bank account details are changed, you will need to correct your print form and upload it to your PDA anew. And if your use variables, you'll just have to correct the information in your PDA. Here is the list of variables for print setup (specified in “Settings” - “Print” - “Details”):
-

Variable	Type	Description
company	TEXT	Your company name
address	TEXT	Your company address
billing	TEXT	Details for payment: Acc. №, bank name
registration	TEXT	Registration number of your company
comment	TEXT	Comment: enter any other useful information

- After an empty line we print a customer name: `Customer: {$doc.customer.name}`. Let's explain how this field is created. `doc` means that the data are the values of Details tab of a document (in our case — Details tab of “Request”). In “Request” there is no field `customer.name`, but there is `customer_name`, so you can specify `{$doc.customer_name}` instead. But if you need to print a customer address (and this field is absent in document Details), this format will not be useful. That's why we added the value `customer`, that contains subvalues corresponding to Customers reference (`ref_customers`) for this customer. From this perspective, the values of `doc.customer.name` and `doc.customer_name` are the same, although they are different fields: the first one is the value of the field `name` from the reference `ref_customers` of a customer selected in the “Request” document, and the second one is the field value of `customer_name` in Details tab of a current document.
- We print `Address:` and enter the value created according to the principles described in the previous paragraph `{$doc.customer.PostAddress}`.
- The command `{% escp.esc E %}` selects bold font. Then we print `DELIVERY NOTE №`, then the values of the fields `number` and `date` of the document that initiates printing. `{% escp.esc F %}` cancels bold font, `{% escp.esc 0 %}` selects a 1/8-inch line spacing.
- Next four lines are the header of the document and are printed as they are.

`{% loop in $table as $row counter=$num %}` created a loop to print all records of a document's tabular section `$table`; the variable `$num` is a counter.

`|{$num|qcalc(+1)|sprintf(%-4s)}| {$row.name|left(25)|sprintf(%-25s)}| {$row.product.measure_unit|left(4)|sprintf(%4s)} | {$row.count|sprintf(%8.2f)}|{$row.price|sprintf(%11.2f)} | {$row.sum|sprintf(%12.2f)}|` - first prints two spaces and a vertical bar, then increases the value of the counter `$num` by 1, and prints the document in the format: 4 characters with left aligning, then a vertical bar and a space. Then we cut the variable `$row.name` leaving only the first 25 symbols (procedure `left(n)`, which discards all characters to the right of `n`, was created by the developers and integrated into PalmOrder), print the result, also aligning it to the left, then a vertical bar and so on up to the end of the line. The rest of the lines are printed in the same way. A tabular section of the document is specified as `$table.` Its fields are marked as `$row.name`, `$row.count`, `$row.price`, `$row.sum`. The name of the field `$row.product.measure_unit` is formed according to the principles described in paragraph 5 for the value `customer`. But here we use `product` instead. Its subvalues are the fields of Items reference (`ref_products`). Thus, the variable `$row.product.measure_unit` corresponds to the value of the field `measure_unit` of the reference `ref_products` for the current record of the document's tabular section.

`{% endloop %}` means the end of the loop.

- Then we print a line starting from two spaces.

`{% escp.esc 2 %}` selects a 1/6-inch line spacing.

- After the empty line comes `Total amount:` and the value of the field `sum` of the current document (it is printed as a floating point number with two decimals).
- The last four lines are printed as they are.

`{% escp.0C %}` is form feed. Use it if you print on paper sheets.

Save this instruction in the file `delivery_note.prn` and place it into the outbox folder. Similar instructions must be created for other print forms.

To generate the print data, Chunk library is used, where you can find the detailed information on commands, structures, filters used in print forms.

We additionally developed the command `escp`. This command allows you to send the esc-sequences of characters, as well as arbitrary ASCII characters. Thus, there are two formats:

- `escp.esc <text>` sends to printer ESC symbol and the text specified in the command;
- `escp.<symbol_codes>` sends to printer ASCII symbols specified as codes in the command. The codes must be specified in a HEX format and contain two symbols each. For example, `esc.0C1B33` sends to printer symbol codes 0C, 1B, 33.

Downloading print_forms.xml and Print Forms

When `print_forms.xml` and print forms are created, download them to your PDA by adding a corresponding type of exchange. For this you need to add the following section to exchange.xml:

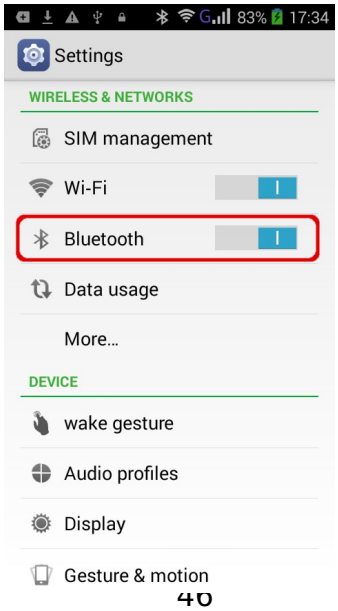
```
<exchange name="print_forms" view="Download print forms">
   <get>
      <files>
         <file name="print_forms_xml" files="print_forms.xml" />
         <file name="print_forms_prn" files="*.prn" />
      </files>
   </get>
</exchange>
```

According to this instruction, your PDA will receive `print_forms.xml` and all created print forms (`*.prn`) when the type of exchange “Download print forms” takes place.

Now let's review how to configure the printing settings.

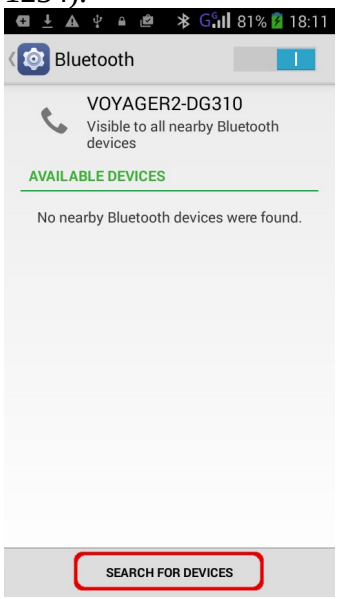
Configuring Printer and Editing Printing Details

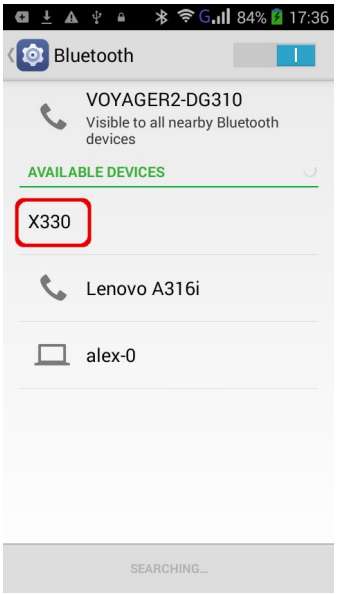
Make sure that BlueTooth printer is on, and the feature of BlueTooth is activated on your

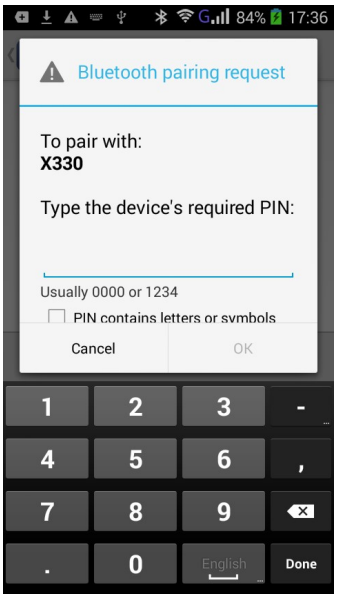


device. Add BlueTooth printer to the list of trusted devices. To do this, choose Android “Settings” — “BlueTooth”.

Select “Search new devices”. Find and select your printer. Enter PIN-code (usually 0000 or 1234).







Start PalmOrder. Choose “Settings”, select “Print” – “Printer”. Select a desired printer in the field “Print device”. In a corresponding field enter a correct code page (it can be, for example, CP866 or CP1251 for Cyrillic symbols — refer to the printer manual). The field “Delay (ms)” contains the value of delay, if printing cuts off, i. e. a print form is not printed up to the end. The value depends on the length of your document. The longer the document, the larger the value. Start with 500 milliseconds. Tap “OK” when you finish. Choose “Print Entries”. Fill in all the fields. Tap “OK”.

To print a document, you need to open it, tap “Menu” – “Print”, select a desired print form (if there are several). Printing will start immediately.

Service Tables

Different service tables are used for the PalmOrder mobile part settings. If necessary, these tables can be created using Xml2Eldb.exe, the same way as similar tables of references, documents.

Global System Settings

Some settings, that are used in the PalmOrder mobile part, are stored in the table `gl_settings`.

The structure of the table `gl_settings`

Field	Type and KEY	Description
`_id`	`INTEGER`	A primary key of a tabular section. A valid value is >0, is sorted in an ascending order, the first field in a table
`code`	`TEXT`	The code of settings
`value`	`TEXT`	The value of settings

The list of settings and their purpose description

№	Settings Code	Type of Constant	Name of Constant	Description
1	`ADMIN_USER_PASS`	TEXT	Administrator's password	This password is requested when you try to edit administrator's features. The value by default is `admin`

Interaction with AIS (ERP)

PalmOrder is compatible with the AIS or ERP systems, that can perform exchange using the XML documents, i.e. formation and analysis of the XML documents are carried out by means of AIS (ERP).

This chapter describes in details the purpose and format of the files that you need to download from or upload to AIS (ERP).

Downloaded and uploaded documents

To avoid confusion, the terms *downloaded* and *uploaded* documents in the context of this chapter shall apply in relation to AIS (ERP). Uploaded is the document that is uploaded to server and further to AIS (ERP); and downloaded is the one that is downloaded from AIS (ERP) to server and further to the PalmOrder mobile part (a PDA).

A brief description of the XML documents

PalmOrder uses three formats of documents to perform exchange.

XML-Document	Assignment	Description
Reference	Downloading a reference to a PDA	A reference file is formed in AIS (ERP) for further download to the PalmOrder mobile part (a PDA). After creation an XML-document must be converted to *.db file using the Xml2Eldb.exe utility
Document	Downloading a document to a PDA	A document file is formed in AIS (ERP) for further download to the PalmOrder mobile part (a PDA). After creation an XML-document must be converted to *.db file using the Xml2Eldb.exe utility
Document	Uploading documents to AIS (ERP)	A document is saved in the PalmOrder server part during the exchange with a PDA. A saved XML-document must be uploaded to AIS (ERP)

Formation of the XML Documents

The XML documents are formed for data exchange between PalmOrder and AIS (ERP). An XML document must be “correct,” which means that at the beginning of a document there must be the instruction of XML declaration followed by a root element. The content of an entire document is enclosed between the initial and final tags:

```
<?xml version="1.0" encoding="windows-1251"?>
<root element>
</root element>
```

You need XML version 1.0. A root element can differ depending on a document. Below the formats for all types of the PalmOrder XML documents are described. Every type of a document is intended only for downloading from or uploading to AIS (ERP). If a document is intended for downloading from AIS (ERP) to a PalmOrder PDA, it must be converted into a corresponding format.

See http://www.w3.org/TR/xml/ for more details on the XML documents.

The Format of Data Downloaded from AIS (ERP) (References, Lists, Documents)

The PalmOrder server part sends a set of tables of relational database, which are the source for creating references, lists, documents etc. on a PDA. The tables are formed from the specific XML documents and converted to a binary format for downloading to a PDA using the

Xml2Eldb.exe utility.

Below there is the example of the description of an XML document that forms a database.

```
<?xml version="1.0" encoding="windows-1251"?>
<table name="Reference Name">
   <schema>
      <field name="_id" type="INTEGER" />
      <field name="code" type="TEXT KEY" />
      <field name="name" type="TEXT" />
         ...
      <field name="Field name" type="Data Type" />
         ...
      <field name="parent" type="TEXT KEY" />
      <field name="group" type="INTEGER" />
   </schema>
   <data>
      <record>
         <f>Serial number</f>
         <f>2222</f>
         <f>Beer Lech</f>
            .....
         <f>Field value</f>
            .....
         <f>0</f>
         <f>1</f>
      </record>
         .....
      <record>
         <f>Serial number</f>
         <f>5555</f>
         <f>Beer Budwiser</f>
            .....
         <f>Field value</f>
            .....
         <f>2222</f>
         <f>0</f>
      </record>
         .....
   </data>
</table>
```

The formation rules for an XML document that contains a DB table

- A document begins with the instruction of XML declaration. Then comes a root element between tags `<table>` and `</table>`:

`<table name="Table Name">`
`</table>,`

where `name` is the name of a table.

- Then comes a table scheme enclosed in tags `<schema>` and `</schema>`.
- After scheme there are downloaded records (i.e. a reference itself or any other table) enclosed in tags `<data>` and `</data>`.
- Between tags `<schema>` and `</schema>` there is a description of all downloaded table fields:

`<field name="Field name" type="Data type"/>`, where `name` is the name of a table field in PalmOrder, `type` is a data type of a field.

- There are records between `<data>` and `</data>`. Each record is enclosed in tags `<record>` and `</record>`.
- Every pair `<record>` and `</record>` contains the field values enclosed in tags `<f>` and

</f>. The number and order of field values of reference elements must exactly match the structure of a scheme described at the beginning of an XML document between tags <schema> and </schema>.

The Format of Uploaded Documents

```
<?xml version="1.0" encoding="utf-8" ?>
<post user="User Name">
   <docs name="Document Name">
      <doc>
         Description of document
      </doc>
   </docs>
</post>
```

The formation rules for an XML file of an uploaded document

- A document begins with the instruction of XML declaration.
- Then comes a root element marked by tags <post> and </post>. Between <post> and </post> there is the whole content of a document:

```
<post user = "User Name">
         ...
</post>
```

where user is the name of a PDA the data is downloaded from.

- Further between tags <post> and </post> there are uploaded documents. All documents are enclosed in tags:

```
<docs name = "Document Name">
         ...
</docs>
```

where name is the name of uploaded documents.

- Each document in enclosed in tags <doc> and </doc>.

The description rules for uploaded documents

```
<?xml version="1.0" encoding="utf-8" ?>
<doc>
   <head>
      <field name="Field of document Details section" value="Field value" />
        .....
   </head>
   <table>
      <row>
         <field name="Tabular section field name" value="Field value" />
           .....
      </row>
      <row>
        .....
      </row>
   </table>
</doc>
```

- In the block <head>...</head> we describe the fields of a document Details section. The format of a field description:

```
<field name="Field of document Details section" value="Field value" />
```

- In the block <table>...</table> we describe a tabular section of a document. It can be omitted if a document does not contain a tabular section.
- Between tags <table>...</table> we describe the values of a document tabular

section. Every element is enclosed in tags <row>...</row>, inside them there are field values of uploaded elements:

```
<field name="Tabular section field name" value="Field value" />
```

The Utility for Processing the XML Documents

PalmOrder uses two programs to process different types of documents. Both are useful to process the XML documents. Here is a brief description:

Program	Processed Document	Result
Palm_Server.exe	1. Receives information transferred from a PDA. Works automatically 2. When a PDA requests some db-files, Palm_Server.exe finds and transfers them	1. Forms the XML files of documents on a PDA for further processing in AIS (ERP) 2. Data downloading
Xml2Eldb.exe	An XML document with a database table (reference, list, document) downloaded from AIS (ERP)	A db-file used by Palm_Server for further uploading to a PDA

The Utility Xml2Eldb.exe

Xml2Eldb.exe is a program for converting the XML documents (references, lists etc.) that are downloaded from AIS (ERP) into db-format files for further transmission to a PDA. Xml2Eldb.exe is executed from the command line. The XML documents to be converted into db-files act as parameters. Usually an AIS (ERP) processing module is configured in such a way, that after the formation of the XML documents, Xml2Eldb.exe is started automatically to convert them into db-files. So there is no need to start it manually.

The parameters of the Xml2Eldb.exe:

When you execute Xml2Eldb.exe with no parameters, you will see the following hint on your screen:

```
Usage: Xml2ElDb [-options] [--e=<encoding>] [--id=<id>] [--
          update=<update_filename>] filename [output_directory]
Version: 6.0.1
Options:
   d - print detailed log
   a - parse all file in <filename> directory
   <encoding> - encoding of database.
       Possible values of <encoding> are: UTF-8, UTF-16, UTF-16le, UTF-16be.
       Default is UTF-8. (LiteERP uses UTF-8. PalmOrder uses UTF-16le.)
   <id> - id field name.
       Possible values of <id> are: id and _id.
       Default is _id. (LiteERP uses _id. PalmOrder uses id.)
   <update_filename> - update file name.
       File should be in specific XML format.
```

where:

Parameter	Name	Description
`-options`	Executing options	One or several options: **Option** / **Description** `-a` — Parameter "-a"("-all") is used to convert all the XML files located in a `filename` folder into a db format `-d` — Displays the details of conversion To use several options at a time, write them together, for example "-ad". Executing options are not mandatory

`--e=encoding`	Encoding of data	Is used to provide compatibility with older versions of PalmOrder. Valid values: UTF-8, UTF-16, UTF-16le, UTF-16be. UTF-8 is a default encoding. When converting, use the same encoding as in the PalmOrder mobile part. PalmOrder for Android uses UTF-8
`--id=<id>`	id field name	Is used to provide compatibility with older versions of PalmOrder. Specifies the name of a primary key. PalmOrder for Android uses `_id` by default
`filename`	File or folder name	The name of an XML document to be converted into a db-file. If the option "`-a`" is used (see above), this parameter means a folder with the XML documents
`output_directory`	Folder name	A folder name, a created db-file(s) is placed to. An optional parameter. If not specified, the created db-files are placed in a current folder

The example of use:

```
Xml2Eldb -a c:\base\xml\ c:\base\db\
```

After executing this command, all XML documents located in `c:\base\xml\` will be converted into db-files and placed into `c:\base\db\`.

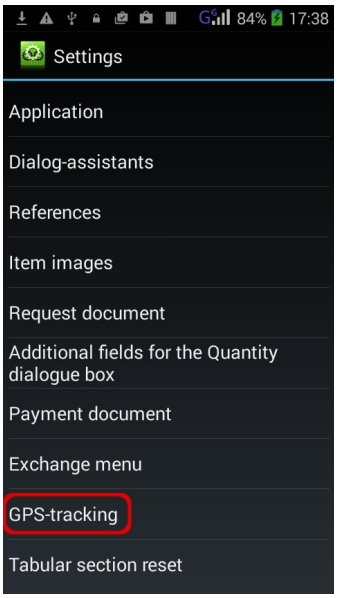
Processing with a GPS-Tracking Server

PalmOrder allows you to monitor your mobile employees' tracking by means of device positioning sensors that work through GPS, mobile network, Wi-Fi etc.

You need to configure the connection of your PDA to a GPS-tracking server. We created an additional interface of interaction between your device and a PalmOrder GPS-server. You can upload scheduled routes and compare them with real tracks. Or you can upload the coordinates of a customer and control, whether a document was created in a customer's office or outside.

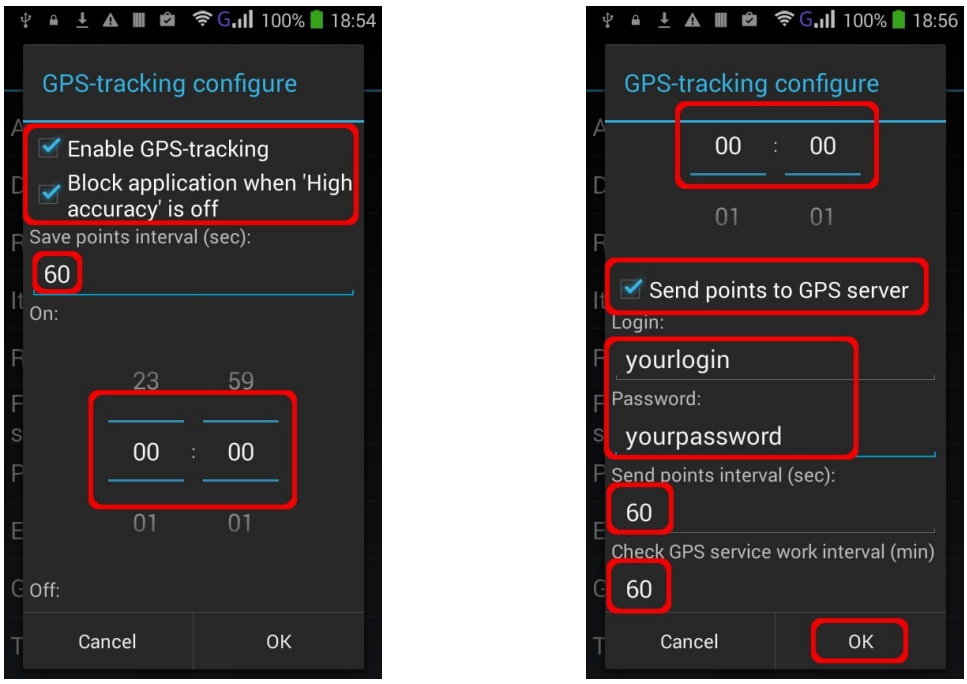
Configuring a PDA

- Launch PalmOrder on your PDA. Tap “Settings” and choose “GPS-tracking” (password by default is *admin*).



Pic. 11.1 Setting menu

- Tick off the check-boxes “Enable GPS-tracking”. If you want your mobile employees not to turn off the GPS-module on a PDA, enable the option “Block application when 'High accuracy' is off”.
- Set the interval of saving coordinates (in seconds). Value by default is 60 seconds. Besides it should be mentioned, that Android versions lower than 4.3 ignore this parameter and attempt to get coordinates every second. That is why we implemented the condition of getting new coordinates only if a PDA location changes for more than 5 meters.
- Set the time to start and stop feature activity. If start time = stop time, the feature will operate permanently.
- Enable “Send points to GPS-server”, fill in the fields of “Login” and “Password” with values you received when registering on gps.palmorder.com.
- Set the interval of sending coordinates (in seconds). Use the principle: the weaker the battery, the longer the interval. Value by default is 60 seconds.
- Set the period of checking the GPS module operation. This option is necessary because the operating system can suspend the service (for example, in case of insufficient memory). In this case the service will be re-launched, and the saved coordinates will be transferred to a server. A period is set in minutes. Value by default is 60 minutes.



Pic. 11.2, 11.3, 11.4 Configuring a PDA for GPS-tracking

- Tap “OK”.
- Make sure that a GPS module operates, and set high accuracy mode. The Android security policy does not allow enabling/disabling a GPS module with application tools. A PDA owner controls this feature via Android “Settings”. Thus, a mobile employee can disable a GPS module any time. In such case we can only limit the scope of work with PalmOrder.

Server Interaction API

A GPS-server API is based on the RESTful architecture. You can upload/download Customers reference and a scheduled route. For the authentication the server uses the HTTP-headers Login and Password, which you received when registering on gps.palmorder.com. The body of a JSON-message contains the resource(s) description.

A Customers reference

A Customers reference table on a GPS-server contains:

Field	Type	Description
`customer_id`	`TEXT`	The code of a customer
`name`	`TEXT`	The name of a customer
`latitude`	`REAL`	Latitude
`longitude`	`REAL`	Longitude

URI: customer

HTTP-method	Description
`POST`	Adding new customer(s) to a reference
`PUT`	Substituting customer(s) in a reference

GET	Getting customer(s) from a reference
DELETE	Deleting all customers from a reference

The body for the methods POST and PUT: array of object

```
{"customer_id":"customer_code",”name":"customer_name","latitude":"latitude","l
        ongitude":"longitude"}
```

The example of an HTTP-request that replaces the entire Customers reference on a server:

```
PUT /customer HTTP/1.1
Host: gps.palmorder.com
Accept: */*
Login: my_login
Password: my_password
Content-type: application/json
Content-Length: 363
[{"customer_id":"4","name":"Standard","latitude":"-
        50.3034","longitude":"28.6223"},
{"customer_id":"5","name":"Ideal","latitude":"-
        50.3034","longitude":"28.6423"},
{"customer_id":"6","name":"EnergoSintez","latitude":"-
        50.3234","longitude":"28.6623"},
{"customer_id":"7","name":"NoName","latitude":"-
        50.3434","longitude":"28.6623"}]
```

The example of a successful HTTP-answer:

```
HTTP/1.1 200 OK
Server: nginx/1.4.7
Content-Type: text/html;charset=UTF-8
Content-Length: 0
Connection: close
```

URI: customer/customer_id

where *customer_id* is the code of a customer

HTTP-method	Description
PUT	Substituting a customer in a reference
GET	Getting a customer from a reference
DELETE	Deleting a customer from a reference

The body for the method PUT: object

```
{"customer_id":"customer_code","name":"customer_name","latitude":"latitude","l
        ongitude":"longitude"}
```

The value of the field *customer_id* is ignored.

The example of an HTTP-request that returns the information about a customer with code 4.

```
GET /customer/4 HTTP/1.1
Host: gps.palmorder.com
Accept: */*
Login: my_login
Password: my_password
Content-type: application/json
Content-Length: 0
```

The example of a successful HTTP-answer:

```
HTTP/1.1 200 OK
Server: nginx/1.4.7
Content-Type: text/html;charset=UTF-8
```

```
Content-Length: 0
Connection: close
{"customer_id":"4","name":"Standard","latitude":"-50.303400","longitude":"28.6
         22300"}
```

A scheduled route

A scheduled route table on a GPS-server contains:

Field	Type	Description
agent_id	INTEGER	The code of a mobile employee. You can review the settings directly from the site
customer_id	TEXT	The code of a customer
date	TEXT	The date in the format YYYY-MM-DD

URI: plan-route/agent_id/date

where *agent_id* is the code of a mobile employee, *date* is the date of a route in the format YYYY-MM-DD.

HTTP-method	Description
POST	Adding a new route section at the end of a route
PUT	Substituting a route
GET	Getting a route
DELETE	Deleting a route

The body for the methods POST and PUT: array of customers codes

The example of an HTTP-request that creates a route for a mobile employee with code 67 for September 01, 2014:

```
PUT /plan-route/67/2014-09-01 HTTP/1.1
Host: gps.palmorder.com
Accept: */*
Login: my_login
Password: my_password
Content-type: application/json
Content-Length: 13
["5","4","7"]
```

The example of a successful HTTP-answer:

```
HTTP/1.1 200 OK
Server: nginx/1.4.7
Content-Type: text/html;charset=UTF-8
Content-Length: 0
Connection: close
```