



# RemoteQuery

For iPhone and iPod Touch

## User's guide

version 1.0 – August 2009

# Introduction

---

**RemoteQuery** is a simple-to-use tool for browsing the content of your company's databases. To do that, it communicates with a server-side counterpart, that needs to be installed in one or more web servers connected to the databases you need to access.

The server side components are open source, free for any use, and easy to install and configure; they are available at the address

<http://www.logicainformatica.it/RemoteQuery.htm> together with a ReadMe.doc that describes how to install and configure them.

**RemoteQuery** can work with any relational database supported by an OLE DB or JDBC driver, that is to say with any relational DBMS.

# Welcome

---



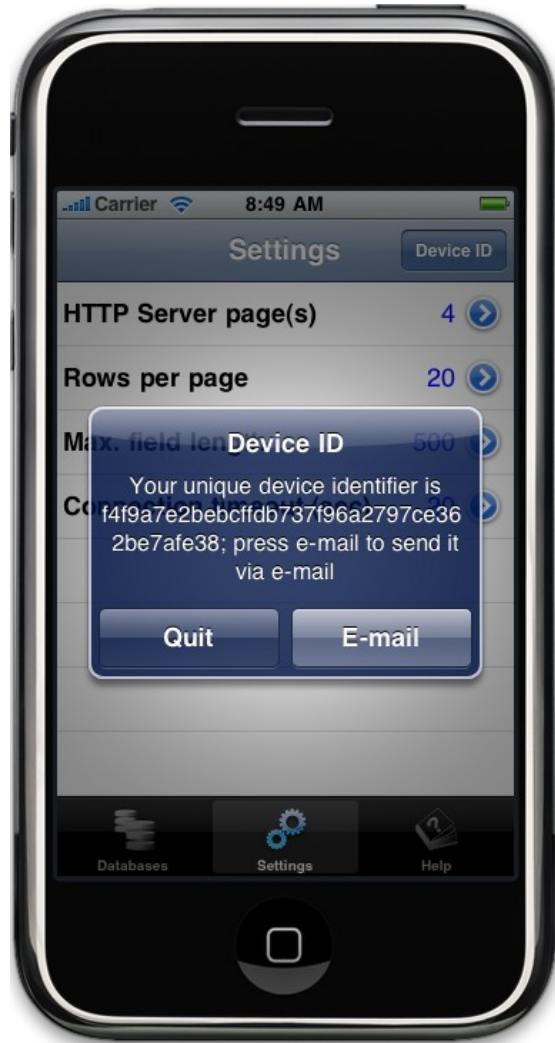
Upon installation, the first page shows a list of three **databases**, pre-configured to let you play with the application before setting up your own environment.

The three names listed corresponds to three small demo databases set up in a Windows server owned by Logica Informatica.

The first thing you have to do then is to replace those demo definitions with your own; that means to go through three steps:

1. have your system/db administrator install the **server-side components**
2. communicate your **device ID** to your system/db administrator
3. configure the **connection** to server

# Getting your device ID



Each iPhone and iPod touch has a unique **device identifier**, that is used to selectively authorize users to access databases, in the server-side components of Remote Query.

To allow your system administrator to authorize your device, you must then give him/her your device ID. This can be easily done by going to the **Settings** page (middle button of the main page button bar) and tapping on the **“Device Id”** button visible at the top right corner of the screen.

This will produce a message box showing the device ID: you can then press the E-mail button to open the Mail application and sending the ID to a mail address.

# Configuring your connections (1)

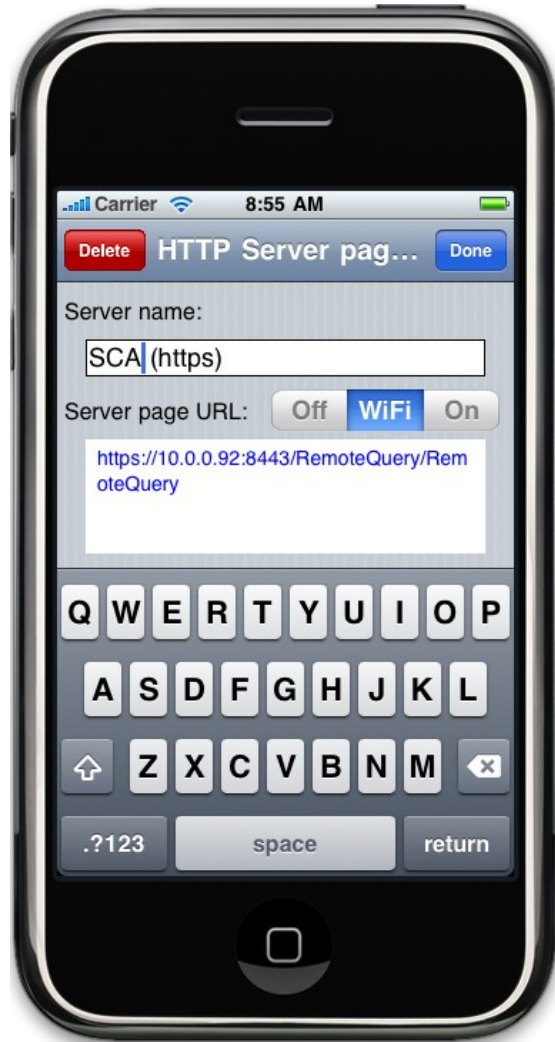


RemoteQuery can connect to one or more servers; each connection is defined as an **URL** to the server-side component, that is an ASP page (in IIS web servers) or a Java servlet.

Your system administrator, after installing the server-side components on a server must tell you the appropriate URL to the server. You must then go to the **HTTP servers page** (see picture) and add a new connection, as described in the next page.

In the list, connections shown in gray color are temporarily disabled (from your client-side configuration, see next page).

# Configuring your connections (2)



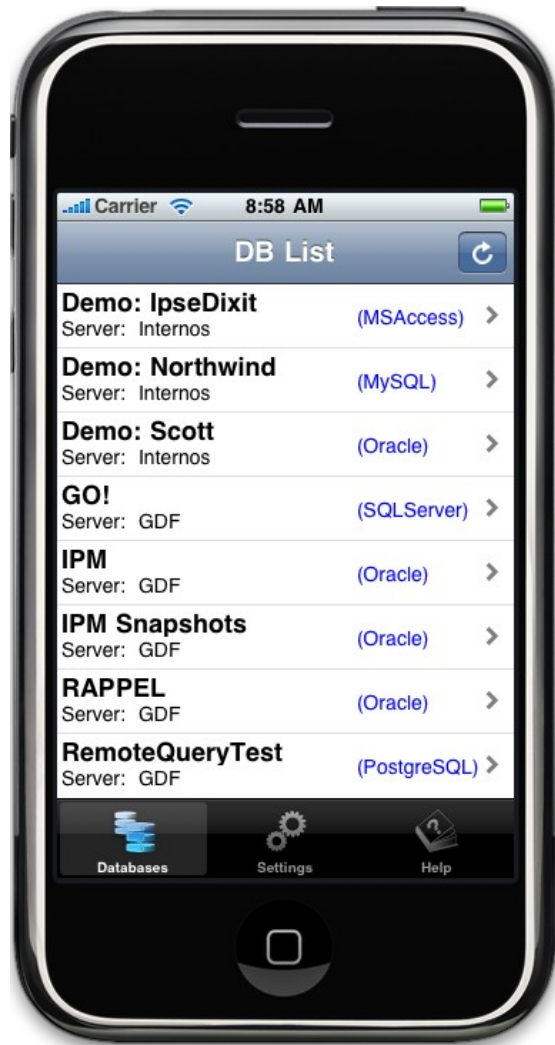
To define a connection, simply give it a mnemonic name and enter the complete connection string, that must be an URL beginning with **http://** or **https://** (see the Installation and Configuration Guide for further explanations).

The segmented button allows you to choose three operating modes:

- Off** the connection is not enabled
- WiFi** the connection must be attempted only when the device is connected to a WiFi network
- On** the connection is always enabled

After any change in the connections, you should go to the database list (first button in the button bar) and press the **refresh** button for getting an up-to-date listing.

# Main page: database list



On start, RemoteQuery will show the list of defined **databases**, the same you obtain by pressing the first button of the main button bar.

The database list is built by concatenating all the lists supplied by the various servers you may be connected to. For better performances, the list is refreshed only when you press the **refresh** button (top right corner): do not forget to press it if the system administrator has changed anything in the server configuration or if you have changed anything in the HTTP servers settings.

For each DB in the list, you see its **mnemonic name**, its **server name** and the database **type**. Duplicate database names are allowed, as far as they stay on different servers.

By selecting a row, you can go to the following level, the list of tables / views / queries.

# Login page

---

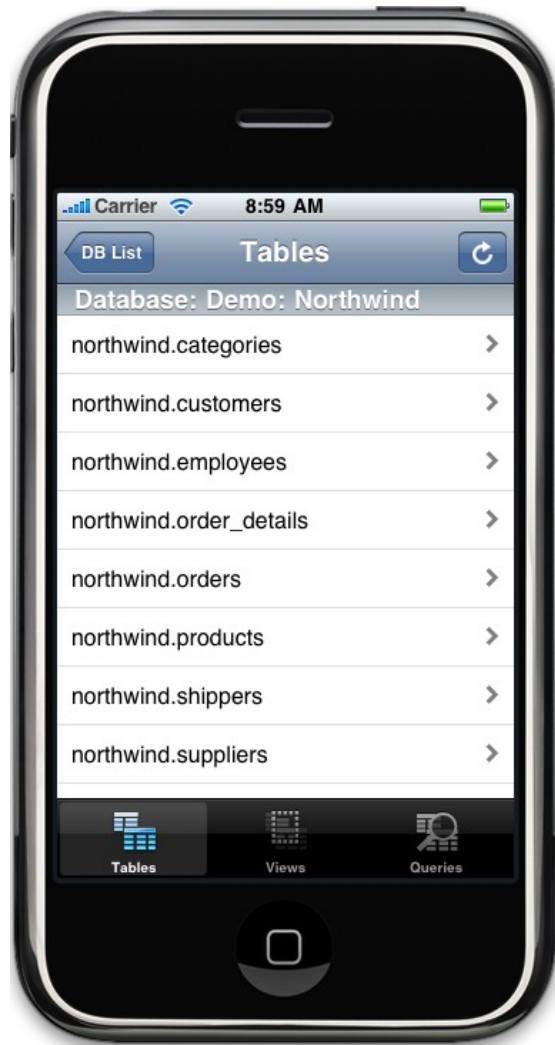


The system administrator may configure RemoteQuery so that an **explicit** user login is requested before gaining access to a database. In such a case, when selecting the corresponding row in the database list, you are presented with a login screen (see picture).

The user and password may be case-sensitive or case-insensitive, according to the database type.

After entering the correct user and password, RemoteQuery will remember them for the duration of the session (i.e. until you exit the program).

# List of tables



When tapping on a database in the db list, you get to the secondary page, subdivided in three (look at the button bar): **tables**, **views** and **queries**.

The “tables” page, visible in the picture aside, list all tables included in the given database. Depending on the way the system administrator configures the logical database, they can be the full set of database tables, or the subset of tables contained in a specific **schema** or an even smaller subset. The list is only refreshed when you tap the **refresh** button (top right corner).

By selecting a row in the list you go to the **data display**, that will be explained later.

By using the buttons in the bottom bar, you can go to the list of **views** or to the list of **queries**, explained in the next two pages.

# List of views

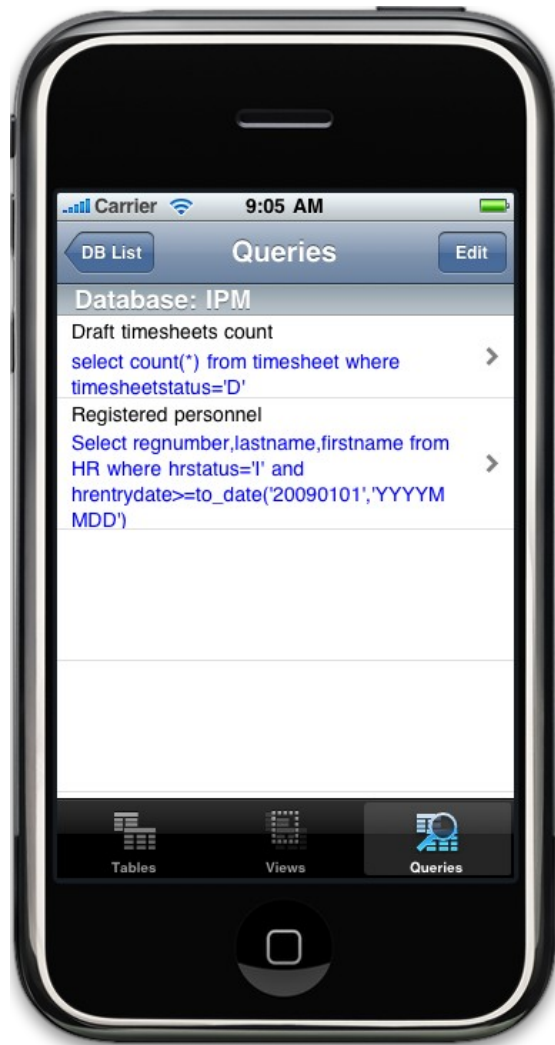


When tapping on the **Views** button in the table list or in the query list, you get the list of views for the database you are browsing. This list is structurally identical to the table list just explained.

This is a list of all **views** (called “queries” in an Access DB) included in the given database. Depending on the way the system administrator configures the logical database, they can be the full set of database views, or the subset of views contained in a specific **schema** or an even smaller subset. The list is only refreshed when you tap the **refresh** button (top right corner).

By selecting a row in the list you go to the **data display**, that will be explained later. By using the buttons in the bottom bar, you can go to the list of **tables** or to the list of **queries**, explained in the next page.

# List of queries



When tapping on the **Queries** button in the table list or in the view list, you get the list of **saved queries** for the database you are browsing.

This is a list of all **SQL** statements that you have created and saved in the tool itself; pay attention: saved queries are **not** objects stored in the target database.

By selecting a row in the list you go to the **data display**, that will be explained later. By pressing the **edit** button you enter the edit mode, that allows you to create, modify and delete queries (see next page).

By using the buttons in the bottom bar, you can go to the list of **tables** or **views**.

# Editing a query

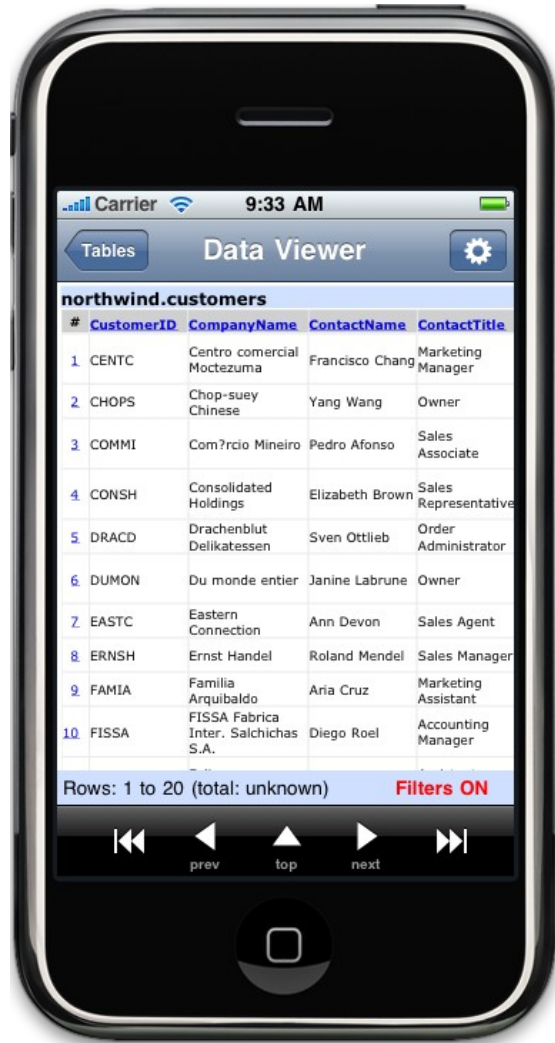


To start editing a query you must first press the **Edit** button in the query list, then you can select the chosen query from the list or you can press the **+** button to create a new query.

Note that, when the list is in edit mode, the row button (**>**) opens the edit page, while when the list is in normal (non-edit) mode, the row button opens the data display (i.e. it executes the query).

The query edit page allows you to create a SQL statement and to save it with a name. The only allowed statement type is **SELECT** (in this version); it must be a valid SELECT statement in the syntax of the specific DBMS; RemoteQuery will pass the statement unchanged to the server (unless filters are defined, see below).

# Data display (1)



When selecting a table, a view, or a query in non-edit mode, data are retrieved from the server and the **first rows** of the table / view or query are displayed on the screen, in the **data display** page.

Each page contains a maximum number of rows (defined in the Settings, initially equal to **20**). With the **navigation buttons** in the bottom bar, you can go through pages. A **status bar** at the bottom indicates what is the current first and last row, what is the total number of rows (if known) and whether a **filter** is applied. Each row is prefixed by its **row number** (relative to the current data set) that works as a hyperlink to get to the row detail display (see page 18).

The **column headers** show the column names as they are defined in the table, view or query; the list can be **sorted** on one or more columns by clicking on the corresponding column header. Sorting is not allowed if the data set is the result of a user defined query containing an ORDER BY clause.

# Data display (2)



Some notes about how data are displayed:

-all texts are truncated to a maximum length defined in the Settings (initially **500** characters)

-null values are displayed as **(null)**

-binary long values are replaced by an indication of their length; for example: **(2109929 bytes)**

The data display can be manipulated as a normal web view, for instance you can:

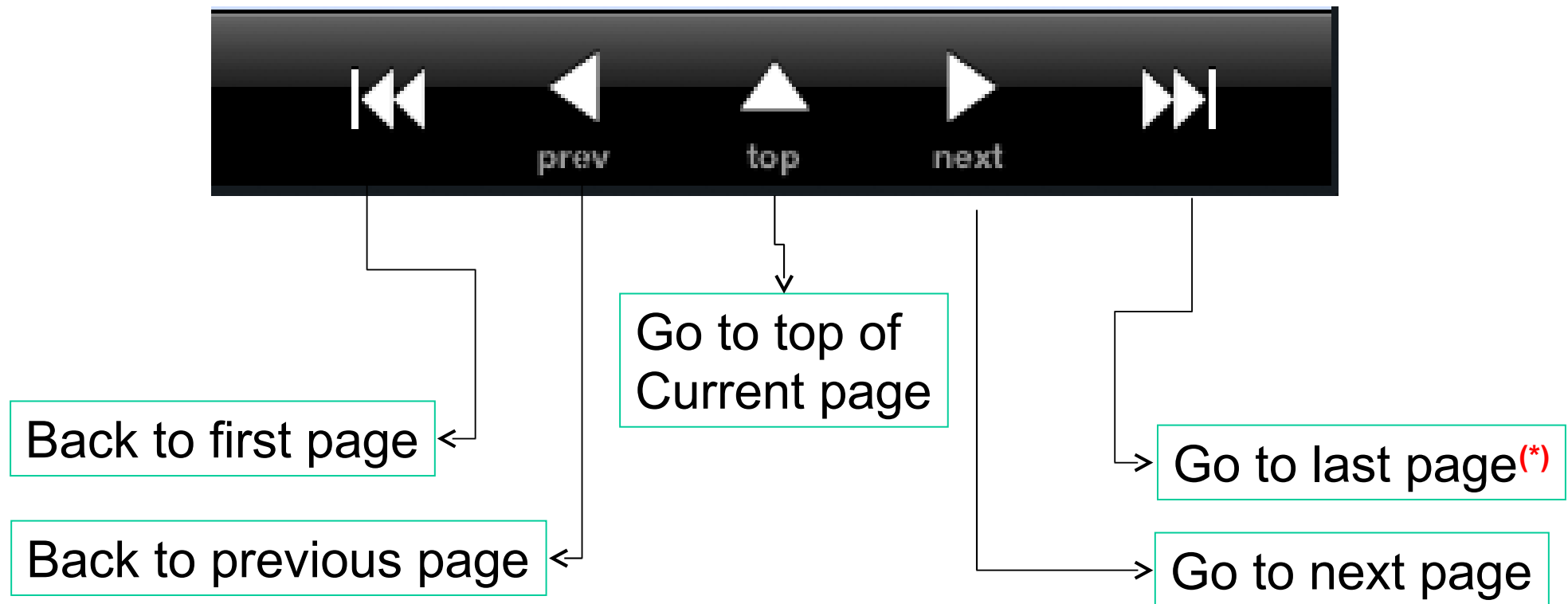
- **scroll** it vertically and horizontally;

- **zoom** it with a click on a cell or with the two-fingers zoom gesture;

-**rotate** it in landscape mode

# Data navigation buttons

---



(\*) If the total number of rows is unknown (as happens with some DB interfaces), the “fast forward” button only go to the last among the visited pages.

# Selecting columns



The data display **configuration page** (see picture) allows you to customize two aspects of the data grid:

- **column visibility**
- **filters**

The page shows all columns in the table or view or query (in their natural order) and allows you to switch on/off their visibility on the data grid, by switching the corresponding checkbox.

Changing a column visibility from **on** to **off** or vice versa does not require any access to the server (all columns are always retrieved from the server, the visibility only changes the way they are displayed).

Editing filters, on the contrary, causes a different SQL command to be sent to the server.

# Filtering (1)



The configuration page also shows which filters (if any) are applied to each column; the **Set** button allows you to edit a filter for each column.

Filters only cover simple conditions (like “is equal to”, “begins with”, “is greater than”) that can be defined for numeric, date and text fields. You do not need to bother about how a string is delimited or a date is specified in the particular SQL; if you need more sophisticated filters, the solution is to define a whole query.

Filters are applied when you press the **Done** button and are saved for subsequent usage. When you exit and enter again RemoteQuery, you will be notified, with the “**Filter OFF**” label in the data display status bar (see picture on page 13), that some filter is defined, but not applied. You may tap the **Filter OFF / Filter ON** label to switch filters on and off.

# Filtering (2)



To edit a filter, press the **Set** button on the corresponding row: this action opens up the row showing two more fields: a pop-up list for choosing the comparison operator and a text field for entering a value (a value is not needed for the “is null” and “is not null” conditions).

After entering a condition you can close back the row by pressing Set again, or you can directly apply the new filter, by pressing Done.

Note: when defining a filter for a date field, you must enter the value in the following format:

**yyyy-mm-dd**

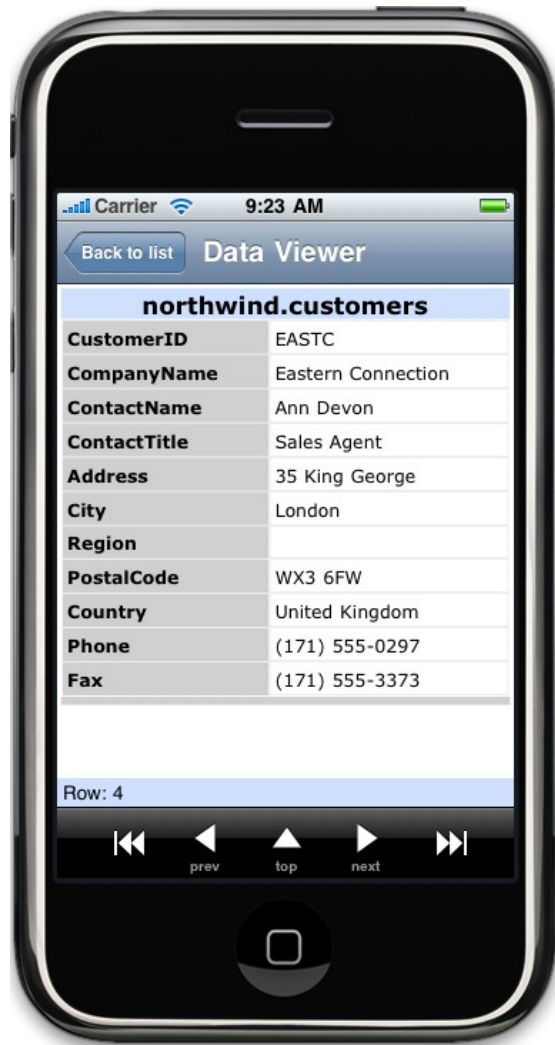
e.g.: 2009-08-05

or, including the time part:

**yyyy-mm-dd hh.mm.ss**

e.g.: 2009-08-05 15.40.00

# Row detail display



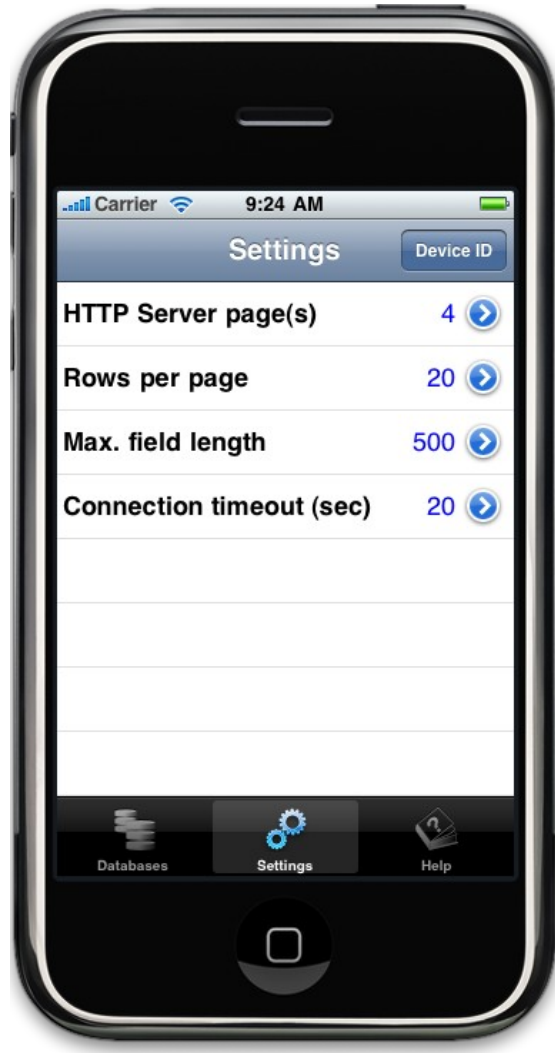
In any data display, the first column contains a relative row number, that can be used to open the **row detail page** (shown aside).

The row detail is a scrollable table of all columns defined in the table, view or query, with the value of each column in the current row.

The navigation buttons in the bottom bar allows you to move back and forth, still remaining in the detail page, so that you can easily inspect all rows with no need to go back to the list.

In the detail page the same rules apply as regards truncations, nulls and binary objects.

# Settings



In addition to the definitions of server connections, the Settings page also contains three other parameters that you can alter to best fit your needs:

## **Number of rows per page** (default=20)

defines how many rows at maximum are retrieved from the server at each call, that is equal to the maximum number of rows displayed in each page of the data display.

## **Maximum text length** (default=500)

maximum length of data read from server for any column (values are truncated to this length if longer).

## **Timeout** (default = 20 seconds)

maximum time the program waits for a response from a server.

# References and contacts

---

All messages, requests and claims can be e-mailed to the following address:

[helpdesk@logicainformatica.it](mailto:helpdesk@logicainformatica.it).

It is not an automated mailbox: a person from Logica Informatica will receive the message and dispatch it in the more convenient way.

If you want to more directly contact Logica Informatica, please refer to the contact information on the company web site:

<http://www.logicainformatica.it>

All remarks and suggestions are welcome if they help us to improve our product.

Thank you.