

**Support Guide**

**Order Tracking**

Prepared by: Andromeda Trading Ltd

Date:

Version:

INDEX

[INDEX 2](#_Toc253742806)

[1 Revision Control 3](#_Toc253742807)

[1.1 Document Approval 3](#_Toc253742808)

[2 Introduction 4](#_Toc253742809)

[2.1 Purpose 4](#_Toc253742810)

[2.2 Scope 4](#_Toc253742811)

[2.3 Definitions and Abbreviations 4](#_Toc253742812)

[2.4 References 4](#_Toc253742813)

[2.5 Overview 4](#_Toc253742814)

[2.6 Contact Details 4](#_Toc253742815)

[3 Terminology 5](#_Toc253742816)

[4 Order tracking overview 6](#_Toc253742817)

[4.1 Order tracking, Rameses and the central Andromeda server 6](#_Toc253742818)

[4.2 The GPS tracking map 6](#_Toc253742819)

[5 The central Andromeda server 7](#_Toc253742820)

[5.1 Overview 7](#_Toc253742821)

[5.1.1 Administration 7](#_Toc253742822)

[5.1.2 Geo-coding 7](#_Toc253742823)

[5.1.3 Proximity delivery 7](#_Toc253742824)

[5.1.4 Client Services 7](#_Toc253742825)

[5.2 Communications overview 8](#_Toc253742826)

[6 Rameses Integration 9](#_Toc253742827)

[7 Logs 10](#_Toc253742828)

[7.1 The log viewer application 10](#_Toc253742829)

[7.2 Event types 11](#_Toc253742830)

[7.3 Configuring Logging 11](#_Toc253742831)

[7.3.1 Enabling and disabling DEBUG events 12](#_Toc253742832)

[7.3.2 Order tracking events 12](#_Toc253742833)

[8 Comms 13](#_Toc253742834)

[8.1 Overview 13](#_Toc253742835)

[8.2 SQLite 13](#_Toc253742836)

[8.3 The comms database 13](#_Toc253742837)

[8.3.2 Messages to the central Andromeda server 14](#_Toc253742838)

[8.3.3 Data retrieved from the central Andromeda server 17](#_Toc253742839)

[8.3.4 Troubleshooting Comms 18](#_Toc253742840)

[9 Andro Admin Area 20](#_Toc253742841)

[9.1 Overview 20](#_Toc253742842)

[9.2 Edit a Store 21](#_Toc253742843)

[9.2.1 Store Details 21](#_Toc253742844)

[9.2.2 Current Details 22](#_Toc253742845)

[9.2.3 Account Details 24](#_Toc253742846)

[9.2.4 Site Logs 24](#_Toc253742847)

[9.3 Client Tracking Map 24](#_Toc253742848)

[9.4 Future Development 27](#_Toc253742849)

# Revision Control

| **Revision** | **Description of Changes** | **Revised By** | **Signed Off** | **Date** |
| --- | --- | --- | --- | --- |
| 1.0 |  | AuthorName | No |  |
|  |  |  |  |  |
|  |  |  |  |  |

## Document Approval

|  |  |  |  |
| --- | --- | --- | --- |
| **Signatory** | **Designation** | **Signature** | **Date** |
| Ben Cole | CIO |  |  |
| Ben Portsmouth | CEO |  |  |

# Introduction

## Purpose

The purpose of this documentation is to guide users through Order Tracking and to be used for training purposes for clients

The following details that will be included in this document:

* Process flows for the system
* Explanation of processes
* Screen Explanations
  + Detailing each field
  + Detailing buttons/commands that can be run from screen
  + Definitions of Data Headings

## Scope

This document will cover the available Order Tracking functionality in version X.X.

## Definitions and Abbreviations

|  |  |
| --- | --- |
| **Abbreviation** | **Description** |
| Order Tracking |  |
| Rameses | Andromeda’s POS system |
|  |  |

|  |  |
| --- | --- |
| **Concept** | **Definition** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

## References

|  |  |  |
| --- | --- | --- |
| **Document Name** | **Version** | **Date** |
|  |  |  |
|  |  |  |

## Overview

This document is designed to be used for user reference and training

## Contact Details

For further information please contact Andromeda Trading Ltd on

+44 (0)870 118 8010

or visit

<http://www.androtech.com>

# Terminology

**Geo location, GPS location, GPS co-ordinates, geo co-ordinates, lat / lon, latitude / longitude**

GPS co-ordinates are a pair of numbers that point to a physical location anywhere on the planet, much like the grid reference on a map. The latitude (lat) and longitude (lon) are the names of the two numbers that make up a geo-location. The numbers are fractions and are usually shown with a comma in between. For example, the geo-coordinates of Buckingham Palace are 51.49939, -0.14398.

The physical location on the ground that the co-ordinates point to is known as a geo-location or GPS location. The co-ordinates themselves are known as GPS co-ordinates, geo co-ordinates or lat/lon pair. The first number (51.49939) is the latitude and the second (-0.14398) is the longitude.

**Geo-code, geo-coding**

This is the process of taking an address and finding a geo-location from it, determining the physical location of the address. This is usually referred to as geo-coding an address. The result of geo-coding an address is a set of GPS co-ordinates (a latitude and longitude). The GPS co-ordinates can be used to plot the position of the address on a map. For example, geo-coding Buckingham Palace produces the GPS co-ordinates: 51.49939, -0.14398.

**GPS tracking device, GPS dongle, GPS tracker**

This is a small hand held device roughly the size of a mobile phone that is able to receive radio signals from GPS satellites orbiting the earth. The device uses theses signals to determine its physical location (its geo location). The device then sends its location over the mobile phone network to a central server. It is essentially the same as an in car sat nav system but without the screen and with the addition of an internal mobile phone used to notify a central server of where it is.

# Order tracking overview

The order tracking system is almost entirely external to Rameses. There are two main parts to the system. One part runs on the store computers and the other part runs on a central server. This document focuses on the parts of the order tracking system that run at the store. The store side of the order tracking system was developed using the Microsoft .NET Framework 2.0. The .NET Framework 2.0 must be installed for the system to run.

Order tracking at the outlet is split into two parts. One part sits between Rameses and the central Andromeda server and allows the two to communicate. This part of the system integrates with another system that we developed called comms which allows messages to be queued to be sent to the central server. Comms was designed provide a service that could be used by any part of the system and is currently also used by Loyalty.

The second part of the store order tracking system is the GPS tracking map, a separate .NET Windows Forms application that allows the store to track their drivers and orders on a map in near real time.

## Order tracking, Rameses and the central Andromeda server

When various events occur within Rameses it notifies the order tracking system. The system queues a comms message to tell the central Andromeda server about the event. The comms system then sends the message over the internet to the central server. The following events are currently raised:

|  |  |
| --- | --- |
| **Rameses event** | **Description** |
| Rameses starts | Order tracking gets a list of trackers from the central server and tells the server to clear the list of orders and drivers for that outlet |
| Tracker assigned to driver | Lets the central server know |
| Tracker removed from driver | Lets the central server know |
| Order added | New delivery order is sent to the central server. GPS co-ordinates of order delivery address may be returned |
| Order assigned to driver | Lets the central server know |
| Order status changes | Sends the new order status to the central server |

Rameses also periodically polls the order tracking system to get proximity delivery times (explained later).

## The GPS tracking map

There are two versions of the application. One that is distributed as part of the Rameses system and one that is stand alone and that does not require Rameses at all. At the time of writing only the version that is part of the Rameses system is being used. The map application is distributed with Rameses but it is completely separate, using only settings generated by Rameses. It gets a list of orders and driver positions over the internet from the central Andromeda server every few seconds and plots them on the map.

# The central Andromeda server

There is a separate document that provides more detailed information about the central Andromeda server.

## Overview

The central Andromeda server hosts a database that keeps track of orders and drivers. It provides the following services.

### Administration

The central Andromeda server provides an administration website that allows you to:

* Add a new outlet
* Edit an outlet – change the address and GPS co-ordinates, enable/disable GPS
* View an outlets current orders and trackers
* View the server error logs

### Geo-coding

The central Andromeda server is responsible for geo-coding the delivery address for Rameses orders.

### Proximity delivery

For each order being delivered, the central Andromeda server continuously compares the physical location of the driver with the physical delivery location of the order. When the driver gets to within a preset distance (100 meters) from the order delivery location, it marks the order as proximity delivered.

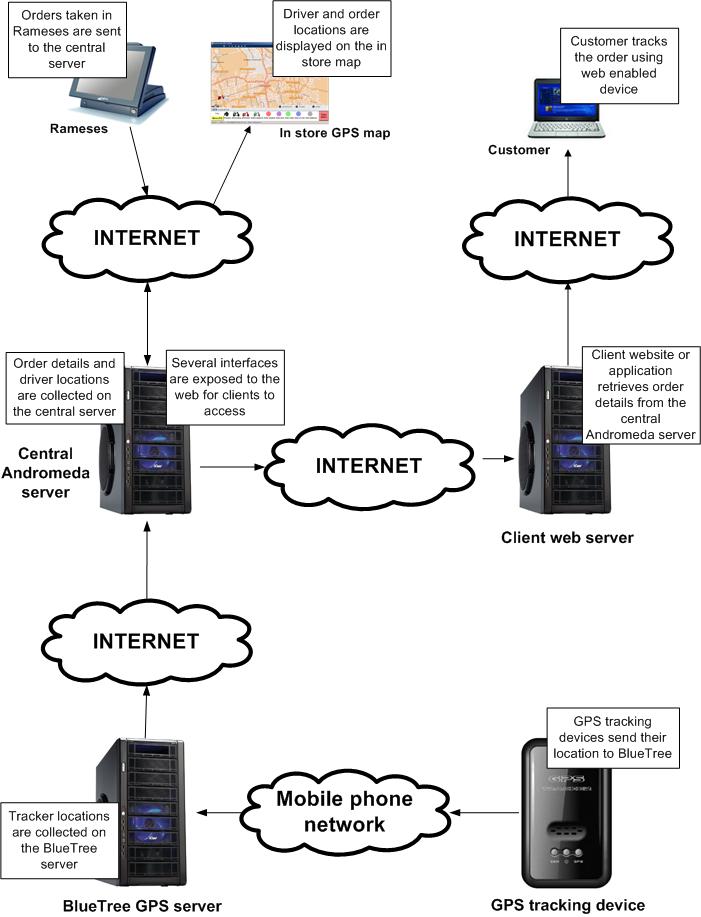
### Client Services

The central Andromeda server exposes a set of services that can be used by external clients over the internet. The services are designed to allow external clients to build their own web sites and applications using order and driver data.

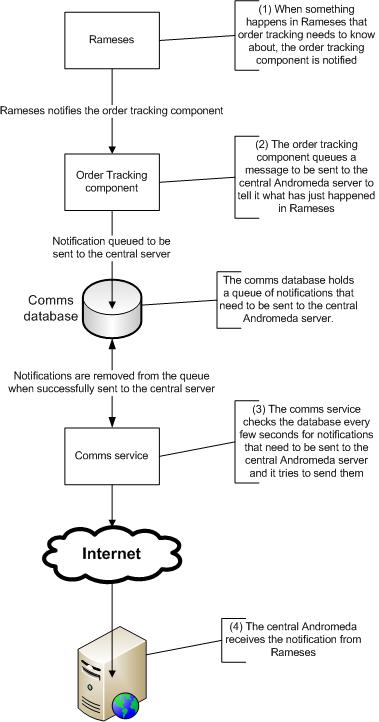
There is a template website available to clients to use for free that uses these services to allow the customer to view their order status and the location of the driver as the order is delivered.

The GPS map application uses one of these services to retrieve details of an outlets orders and drivers to be displayed on the map. The GPS tracking map calls the server every few seconds to get the current driver locations.

## Communications overview



# Rameses Integration



# Logs

All logs can be found in “C:\Rameses\Components\Logging”. The default log files can be found in this folder. There is usually one log file per day. Log files are created even when there are no messages logged so there will probably be empty log files.

There are also several sub-folders for different parts of the system which also contain log files. Again, there is usually one log file per day and there will be empty log files if there are no messages.

Most of the log files are XML which means that they are not easy to read in notepad.

## The log viewer application

There is a utility that can be used to browse the log files which can be found here: “C:\Rameses\Components\SharedGACComponents\Andromeda.Logging.LogViewer.exe”

The log viewer application needs to know where the logs are. The logs are stored in a folder called “Components\Logging” under the main Rameses install folder. You will need to tell the log viewer where Rameses is installed. It defaults to “C:\Rameses”. If you ever need to put the logs somewhere else then ensure that they are placed in a sub-folder named “Components\Logging” under whichever folder you select.

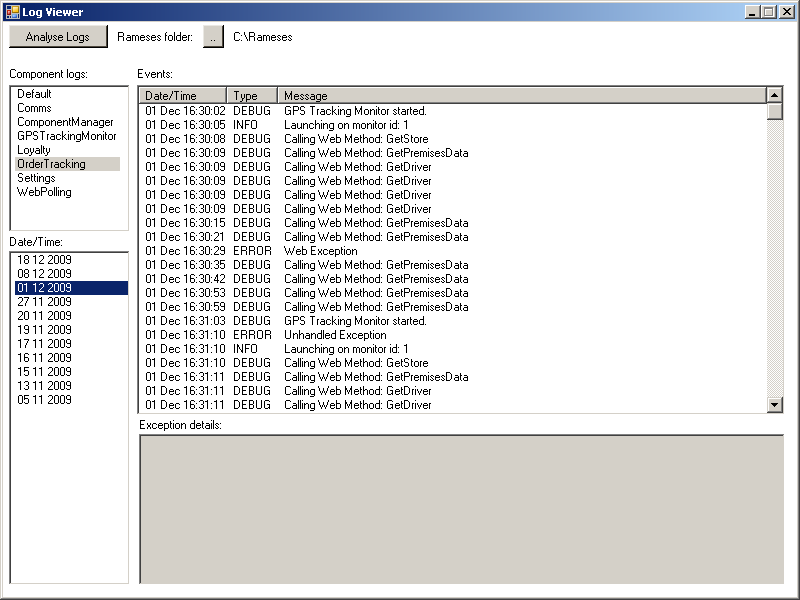


Click this button when you are happy that the correct Rameses install folder is selected.

Click this button to select a different Rameses install folder.

This is the name of the selected Rameses install folder.

(1) There are several different logs. There is a default log and there are logs for many parts of the system. In general, ignore the default log. Select the log you wish to view.



(3) This is a list of all of the events (messages and errors) for the selected log and day.

The date and time that the event was logged.

The type of event. This can be DEBUG, INFO, WARN or ERROR.

The event details.

(4) Sometimes ERROR events contain additional error information. When you click on an ERROR event in the list, this box may show additional error details useful for developers.

(2) The logs are split by day (midnight to midnight). The most recent are shown at the top. Click on the date to view.

## Event types

DEBUG events are useful for developers when trying to diagnose problems. These events will be numerous and should ordinarily be disabled until a developer is trying to debug a system problem. INFO events are relatively insignificant events such as the comms service starting and stopping. WARN events occur when something odd has happened but is not causing a problem. ERROR events are when a serious problem has occurred. There is sometimes additional error information attached to these events.

## Configuring Logging

Open the logging configuration file in notepad. Make sure word wrap is switched off:

“C:\Rameses\Components\Logging\Andromeda.Logging.dll.config”

This is the Log4Net XML configuration file that controls logging. You can change many aspects of the logging using this file.

Be very careful editing this file! If you break it, logging will fail and it may not be immediately obvious what has happened or why.

You will probably need to restart Rameses for any changes to this file to take effect.

### Enabling and disabling DEBUG events

Scroll just over half way down the file contents looking for the following text:

<!-- ########## LOGGERS - One per component ########## -->

Below this line of text is a block of five lines of text for each component. For example:

<!-- Order Tracking logger -->

<logger name="Andromeda.OrderTracking">

<level value="ALL" />

<appender-ref ref="OrderTrackingLog" />

</logger>

In the following line, the text in the quotes controls what kind of events are logged for this particular component:

<level value="ALL" />

Allowable values are:

ALL, DEBUG, INFO, WARN, ERROR

It logs events of the type specified and any higher priority events. The order of priority from low to high is DEBUG, INFO, WARN and then ERROR. So if you specified:

<level value="WARN" />

Only WARN and ERROR events would be logged. Under normal operation it should be set to INFO:

<level value="INFO" />

This will result in INFO, WARN and ERROR events being logged and not DEBUG events.

### Order tracking events

If order tracking is enabled in Rameses then the “Comms” and “OrderTracking” logs may contain useful information. If GPS tracking is also enabled then the “GPSTrackingMonitor” log may contain useful information relating to the GPS tracking map application.

In the logging config file, these are marked as follows:

<!-- Order Tracking logger -->

<!-- Comms logger -->

<!-- Order Tracking GPS Tracking Monitor logger -->

# Comms

## Overview

Rameses notifies the order tracking system when certain events occur, for example when a new delivery order is taken. The order tracking system then queues this information to be sent over the internet to the central Andromeda server.

## SQLite

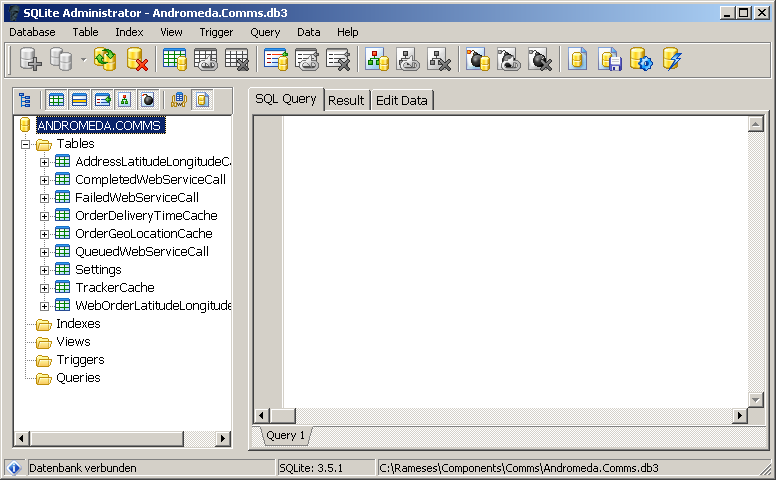
Messages that need to be sent to the central server are stored in a SQLite database. SQLite is a database system not unlike SQL Server but smaller and simpler. It is file based and does not run as a service. There is no install for SQLite as it is embedded inside the application that is using it.

Messages are stored in a database because messages should get through to the central server even if the outlets broadband connection goes down for a while or Rameses is restarted. The database is used as a temporary queue to hold the messages until they can be sent.

## The comms database

The comms database can be found here: “C:\Rameses\Components\Comms\ Andromeda.Comms.db3”. Unlike MS Access and SQL Server, SQLite does not come with management tools to view or design databases. We are currently using a free tool called “SQLite Administrator” which can be downloaded from <http://sqliteadmin.orbmu2k.de/>.

Run the application. Click the “Database” dropdown menu and select “Open”. Browse to the “C:\Rameses\Components\Comms\” folder, select the file “Andromeda.Comms.db3” and click “Open”.

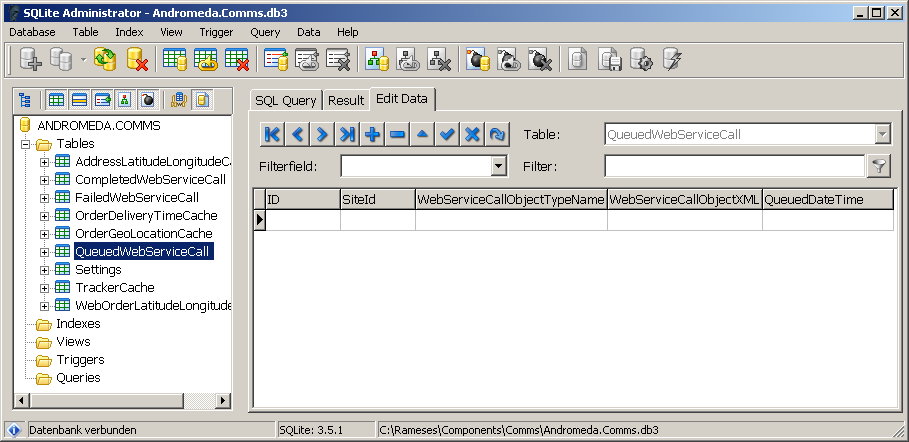


Once opened, the database tables are listed here.

Use the “Database” menu to open the comms database.

SQLite administrator has a slightly unintuitive way of viewing the contents of a table. Click on the “Edit Data” tab first to open the table viewer. Then select the table to view from the list.

(1) Click on the “Edit Data” tab first



(2) Then click on a table to view

#### Tables

There are two groups of tables in the Comms database. Those that are used to send messages up to the central Andromeda server and those that are used to hold (cache) data retrieved from the central server that will be used by Rameses.

### Messages to the central Andromeda server

The three tables that are used to send messages to the central server are:

|  |  |
| --- | --- |
| **QueuedWebServiceCall** | A queue of messages waiting to be sent to the central server |
| **CompletedWebServiceCall** | The messages that have been successfully sent to the central server |
| **FailedWebServiceCall** | The messages that could not be sent to the central server |

These three tables provide a useful audit log of what information has been sent to the server. The three tables are almost identical.

In normal circumstances the QueuedWebServicesCall table should be empty. The order tracking system adds rows to this table when various events occur within Rameses. The comms Windows service runs continuously in the background, checking this table every few seconds to see if there are any rows. If there are, it takes them one by one, starting with the oldest and attempts to send the information over the internet to the central server. If the information is successfully sent to the central server then the row is moved to the CompletedWebServiceCall table. If there is an error sending the information then the row is moved to the FailedWebServiceCall table. The row is only left in the QueuedWebServiceCall table if the broadband connection or the central server is down. The comms Windows service will keep retrying every few seconds until the broadband connection or central server are back up.

#### The FailedWebServiceCall table

The FailedWebServiceCall table holds messages that could not be sent to the central server for some reason. Messages are initially queued in the QueuedWebServicesCall table when Rameses requests a message be sent to the central server. But the messages are moved to the FailedWebServiceCall when a problem is encountered. Note that messages are not moved to this table when the internet connection goes down.

The table has several interesting columns. The columns and data are not necessarily very human readable as they were created to be used directly by the application but can be viewed to diagnose problems.

##### WebServiceCallObjectTypeName

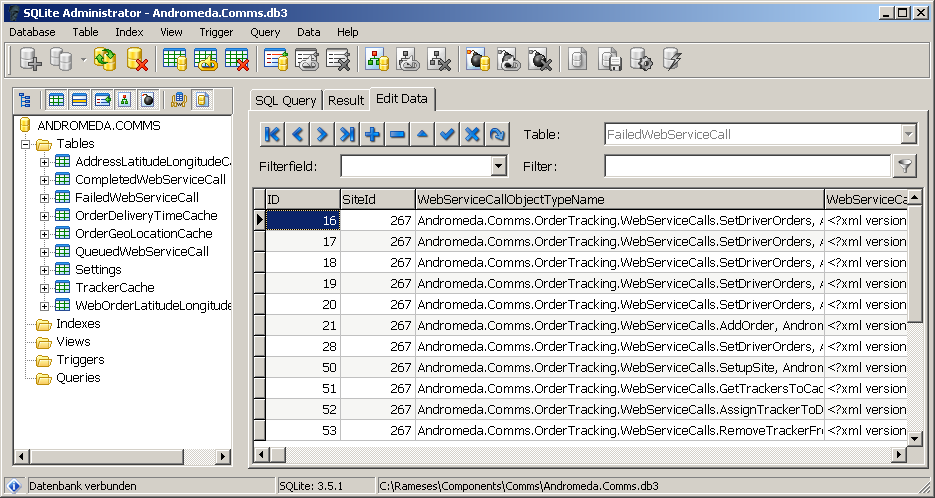
This is the action that Rameses has requested be sent to the central Andromeda server such as a add a new order or assign an order to a driver

You can double click on the cell to open it in a dialog. The data is split in to three parts separated by a comma. Most of the information can be ignored. We are only interested in the first part, for example: “Andromeda.Comms.OrderTracking.WebServiceCalls.**SetDriverOrders**,”

The action requested by Rameses is at the end. In this case SetDriverOrders.

If you adjust the columns in SQLite administrator you should be able to see the actions in the list:

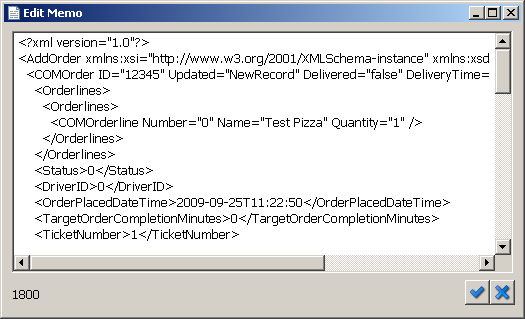
You can see the type of message in the WebServiceCallObjectTypeName column



You can see the data that was to be sent with the message by double clicking on this cell

##### WebServiceCallObjectXML

This is the data that was to be included in the message to the central server. You can double click on the cell to view the data. For example, for AddOrder, the data includes the customer’s name and address. Please note that for some message types (such as AddOrder) only some of the data held in the cell is actually sent to the central server.



The message data is XML but it should be fairly easy to understand.

##### QueuedDateTime

This is the date and time that Rameses initially requested that the message be sent to the central server.

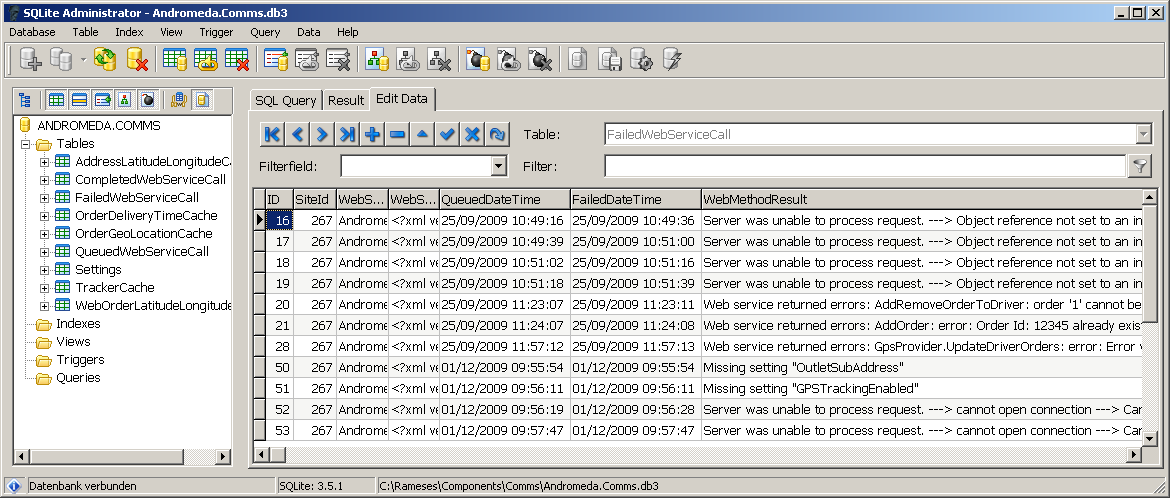
##### FailedDateTime

This is the date and time that the comms Windows service tried to send the message to the central server but encountered the problem.

This is when the error occurred.

This is the error that occurred.

This is when the message was queued.



##### WebMethodResult

This column contains text that describes the error that caused the message to be moved to the FailedWebServiceCall table. If an error was returned by the central server then the text will be prefixed with either “Server was unable to process request. --->” or “Web service returned errors: ”.

### Data retrieved from the central Andromeda server

The tables that are used to cache data retrieved from the central Andromeda server are:

|  |  |
| --- | --- |
| **AddressLatitudeLongitudeCache** | *Obsolete – please ignore* |
| **OrderDeliveryTimeCache** | The time that orders were proximity delivered |
| **OrderGeoLocationCache** | The GPS co-ordinates of order delivery addresses |
| **TrackerCache** | A list of trackers assigned to the outlet |
| **WebOrderLatitudeLongitudeCache** | *Obsolete – please ignore* |

##### OrderDeliveryTimeCache table

When an order is cashed off, Rameses asks order tracking to retrieve the proximity delivery time for that order from the central Andromeda server. Rameses does not wait for the result. Order tracking goes away and retrieves the proximity delivery time in the background. Rameses then periodically asks the order tracking system whether it has retrieved the delivery time yet.

Order tracking gets the proximity delivery time from the central Andromeda server and temporarily stores it in the OrderDeliveryTimeCache table until Rameses asks for it. At which point the time is returned to Rameses and is deleted from the table.

Proximity delivery is specific to GPS tracking. It occurs automatically when the driver gets to within a set distance of the delivery location. The central Andromeda server stores this time for Rameses to pick up later. It is a reasonably accurate time for when the order was actually delivered to the customer as opposed to when the driver returned to the outlet and cashed the order off.

##### OrderGeoLocationCache

Rameses stores GPS co-ordinates for addresses in its database. When a new order is taken in Rameses, it asks order tracking to send the order delivery address to the central Andromeda server. If it already has GPS co-ordinates for the address, Rameses provides them with the order delivery address.

Rameses does not wait for the result. Order tracking sends the new order to the central Andromeda server in the background.

If Rameses does not provide GPS co-ordinates for the address then the central server will geo-code the address to get GPS co-ordinates itself. These are then returned to order tracking. Order tracking temporarily places them in the OrderGeoLocationCache table.

If Rameses didn’t provide GPS co-ordinates, it periodically asks the order tracking system whether it has got the GPS co-ordinates yet. When it does, the GPS co-ordinates are returned to Rameses and are deleted from the table.

**UPDATE**: Currently the webservice geo-codes all customer addresses, the reason for this is that Rameses tends to send the wrong coordinates, or the coordinates are not accurate enough.

##### TrackerCache

This table is specific to GPS tracking. Rameses displays a list of GPS tracker device names on various dialogs. The central Andromeda server keeps a record of which trackers are associated with which outlet. Rather than keep asking the central server for the outlets trackers every time they need to be displayed, they are retrieved once and held in the TrackerCache table.

When Rameses starts, it asks order tracking to get a list of the trackers associated with the outlet from the central server. Order tracking does this in the background and puts the results in the TrackerCache table. Rameses then asks order tracking for the tracker names whenever it needs them.

### Troubleshooting Comms

If there are rows in the QueuedWebServicesCall table and they do not disappear after a few seconds then something is wrong. There are a few things that can be checked. Firstly, check that the broadband connection is working. Check that the comms Windows service is running (see the next section). You can also check that the central Andromeda server is up and running.

#### The Comms Windows service

The service should be automatically started when Rameses starts. Note that the service should be set to “Manual” startup type and NOT “Automatic”.

To check the service, open the services management console. In Windows XP, click Start then Run and type in “services.msc”.

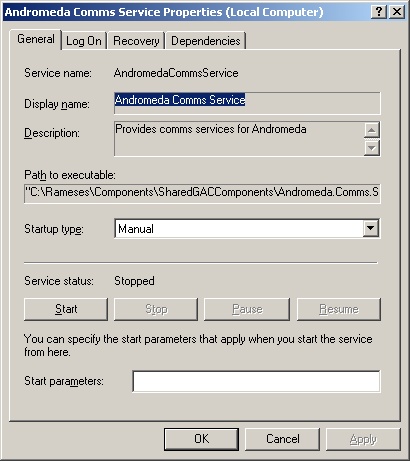
Look for a service named “Andromeda Comms Service”



The Status should be “Started”. In this case the service is **not** running.

The Startup Type should be set to “Manual”

If the service is not running then you can try and start it manually. Double click on the service name in the Services management console. On the service properties dialog click “Start”.



You can try to manually start the comms service

If the service has started successfully and assuming there are no other problems, rows in the QueuedWebServicesCall table should start disappearing. If there are a lot of rows then this might take a while.

If the service fails to start then you can check the “Comms” log and see if there is an error event.

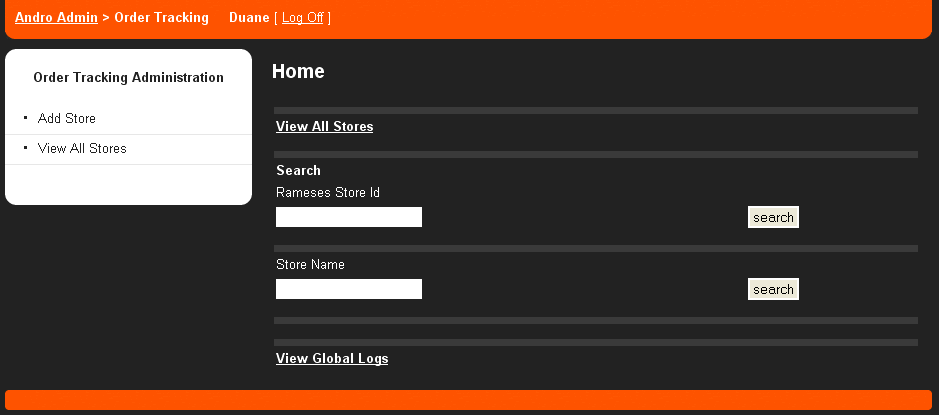
# Andro Admin Area

### Overview

The Andro Admin Area can be found at:

<http://admin.androtechnology.co.uk>

If you have been given sufficient permissions, you will be able to login, and view the Order Tracking Admin area:



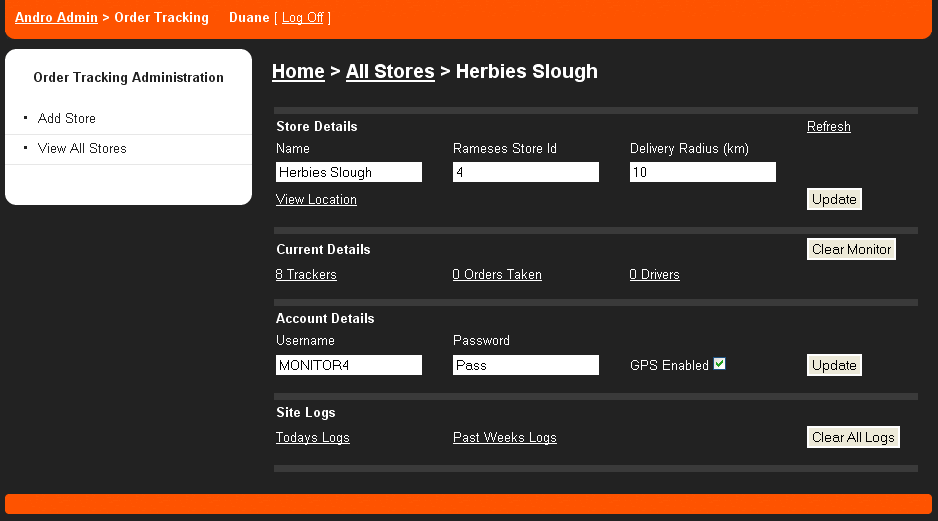
From here you will be able to view add view stores for Order Tracking.

Add Store has been described in the Order Tracking Install Guide.

View All Stores displays a list of all stores that have order tracking setup, alternatively, you can search by the Rameses Store Id.

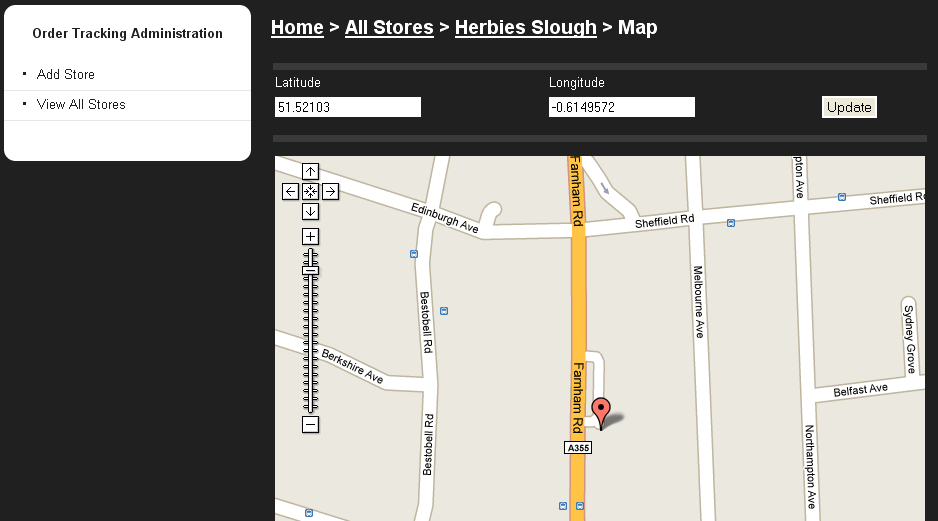
View Global Logs has a list of logs over the past 7 days displaying all ‘uncaught errors’ such as invalid username/password combinations.

### Edit a Store



In this area you are presented will the individual stores details. Most of this is self explanatory, or familiar from setting up the store, a full explanation is in the Order Tracking Install Guide.

### Store Details



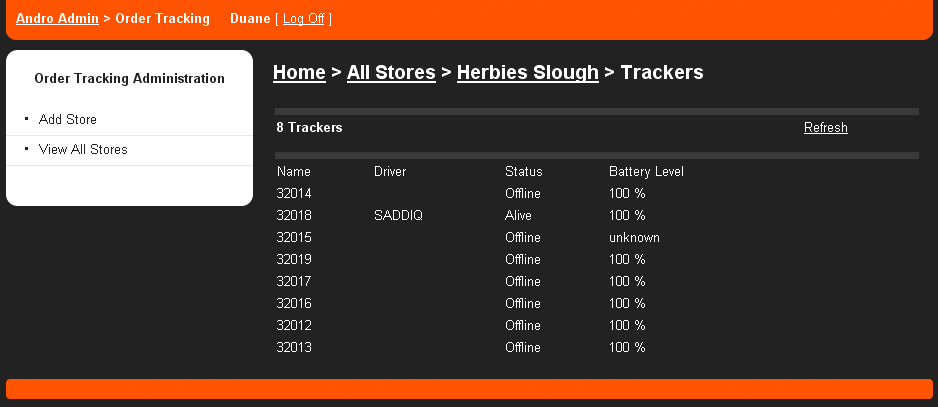
View Location displays a map with the current stores location, if needed you can move the marker to the exact location of the store, click update and the stores location will be changed. This is useful if the store appears in the wrong location on the maps.

### Current Details

#### Clear Monitor

**Please pay particular attention** to the AdminBtnClearMonitor.gif Button. By clicking this, the current data for the store is cleared for the day. If clicked, all current orders, customers, current drivers and their assigned trackers are cleared. All calls to ‘Clear Monitor’ are noted in the site logs.

#### X Trackers (If ‘GPS Disabled’ is displayed, ignore this section)



This displays all the trackers assigned to the store. When a store is first setup, the tracker names may be displayed as ‘unknown’, they will be assigned names once they get a fix.

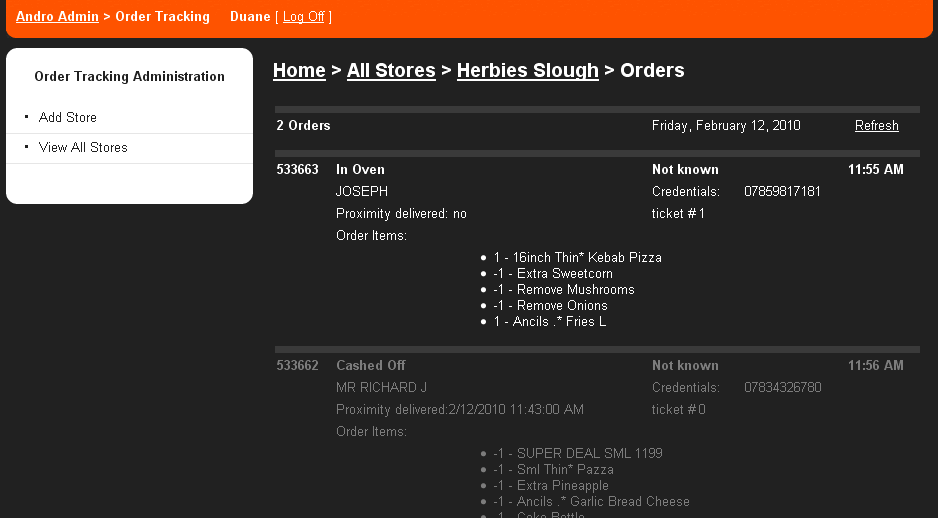
If trackers are assigned to a driver, the drivers name is displayed.

The tracker status is either ‘Alive’ or ‘Offline’. Alive if the tracker has a GPS fix, Offline if it doesn’t.

**NOTE:** Battery level is currently undergoing changes with the new Ranger IIs and currently displays incorrect data, unknown means that the tracker doesn’t have a GPRS connection.

By clicking Refresh, the tracker details are refreshed with current data.

#### X Orders Taken

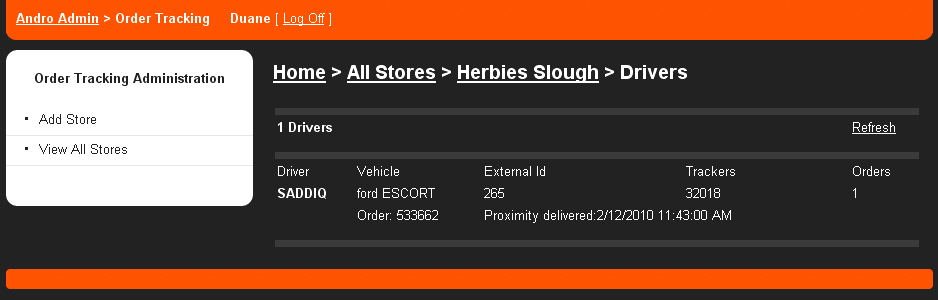


This area displays a list of current orders and all relevant details in their chronological order. Once an order has been completed it will be ‘greyed out’. The time displayed is the time of the last status update, except for ‘proximity delivered’ which will display the time it was triggered.

By clicking Refresh, the current details are updated.

#### X Drivers

Current drivers and if GPS is enabled, their assigned trackers are displayed. If a driver has orders assigned, these are displayed also along with their status.



By clicking Refresh, the current details are updated.

### Account Details

This area has been described in the Order Tracking Setup Guide.

### Site Logs

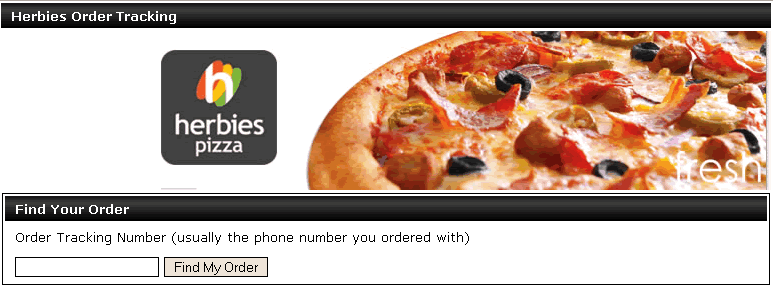
Todays Logs displays a list of warnings and errors for that day.

Past Weeks Logs displays a list of warning and errors for the past 7 days.

### Client Tracking Map

This area was created to cater for a store or chain would like to display the order tracking process to their customers.

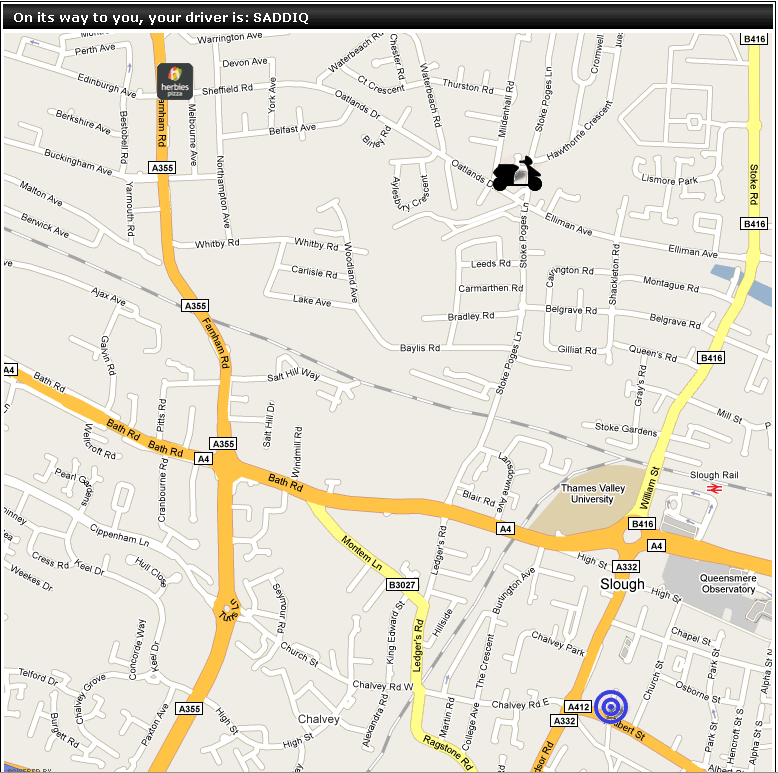
This can be completed in two ways. The first option, by using the current website hosted on androtechnology.co.uk. The store or chain would have link that would open a popup window such as:



Each area can be manually branded for each store/chain. Please refer to the ‘Future Development’ section of this document.

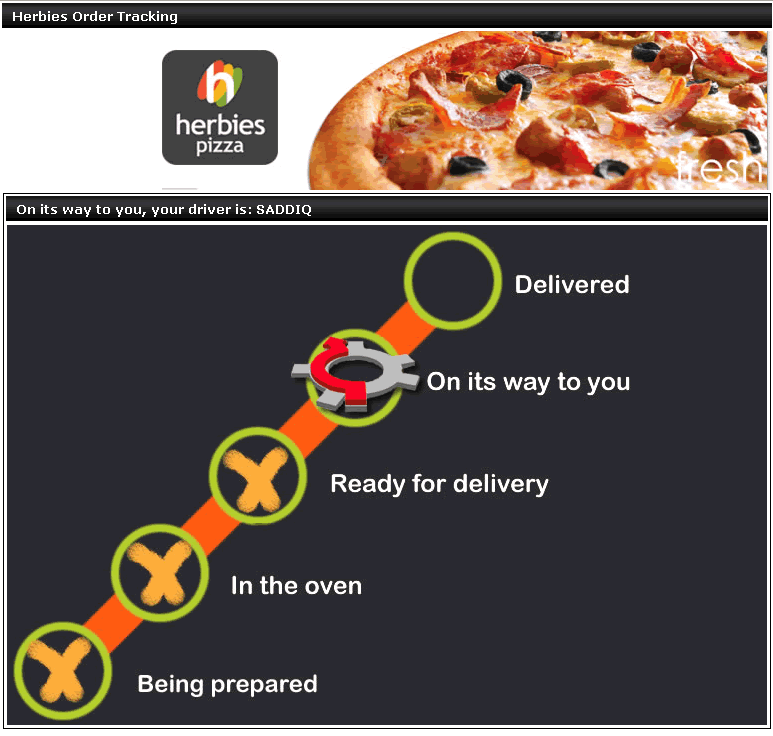
Once the client has entered the Tracking Number, they are presented with a view that contains their current tracking information. This view is updated automatically every 10 seconds. Again the graphics in this area can be modified for each store/chain.

If GPS is enabled, the stores location, the client’s location, and the driver (if the tracker is Alive) will be shown a map.



The store is displayed as a branded icon, the driver as a scooter, and the customer is the blue target.

Once a driver is within 100 metres, the order is considered to be ‘proximity delivered’ and the client is displayed a view such as:



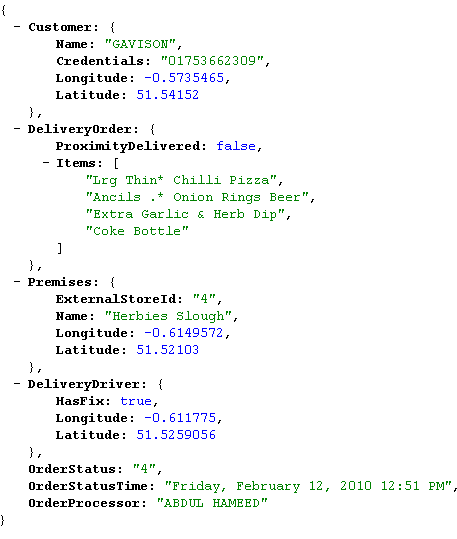
Once an order is cashed off, it will displayed as Delivered.

The second option, which is the preferred, for the client tracking is that the store/chain integrates the map onto their website directly. An example call for Herbies would be:

<http://ordertracking.androtechnology.co.uk/Track/Herbies/GetOrder/07859817181>

This URL should be called no more the once every 10 seconds, the trackers send data every 10 seconds, so it is pointless to call more often.

This call would returns a json request such as this:



DeliverDriver is set to null once the order has been proximity delivered. This is to prevent driver tracking once the order has been delivered.

If a Delivery Driver’s fix is lost, the last known fix is returned as the coordinates. It is best to remove the driver from the map if this occurs (or at least display a different icon) as the trackers are notoriously ‘sensitive’ and may not return a fix for days.

### Future Development

Blackberry/IPhone enabled admin area.

Error handling in the Admin Area, currently there aren’t any data checks on the fields.

Translations to different languages for all the fields in the Admin Area.

When a new store that is going to have GPS Trackers, it requires a long undocumented process with BlueTree, and a server re-boot on their end, which takes down all sites with order tracking. Ideally this should be part of the setup area, we are waiting on a webservice call from BlueTree to facilitate.

Currently under development is the external ‘Client Tracking Area’, an example of this is:

<http://ordertracking.androtechnology.co.uk/Track/Herbies/>

The admin area will contain all the necessary information to create new tracking areas based on chains (eg. Herbies, TBBC, PapaJohns). This process at the moment is a 20 minute job for a developer.

This area can be integrated into existing sites via a popup, or external developers can create their own client order tracking area using an example call below:

[http://ordertracking.androtechnology.co.uk/Track/{ChainName}/GetOrder/{Client\_Order\_Tracking\_Number}](http://ordertracking.androtechnology.co.uk/Track/%7bChainName%7d/GetOrder/%7bClient_Order_Tracking_Number%7d)