aptean

Aptean Proof of Delivery - *Calidus* Edition

CALIDUS VEhub Mobile Device Application Guide

Functional Specification

Contents

Contents

1 Functional Overview

The document is intended to describe the basic functionality of the *CALIDUS* VEhub admin system.

Ready for **What's Next, Now**™

2 *CALIDUS* VEhub

2.1 Login

Login Screen

Enter Company, Username and Password, then press the Login button.

If you have forgotten your login, use the Forgot Password button.

Once logged in, the Dashboard screen will be displayed.

2.1.1 General Screen Details
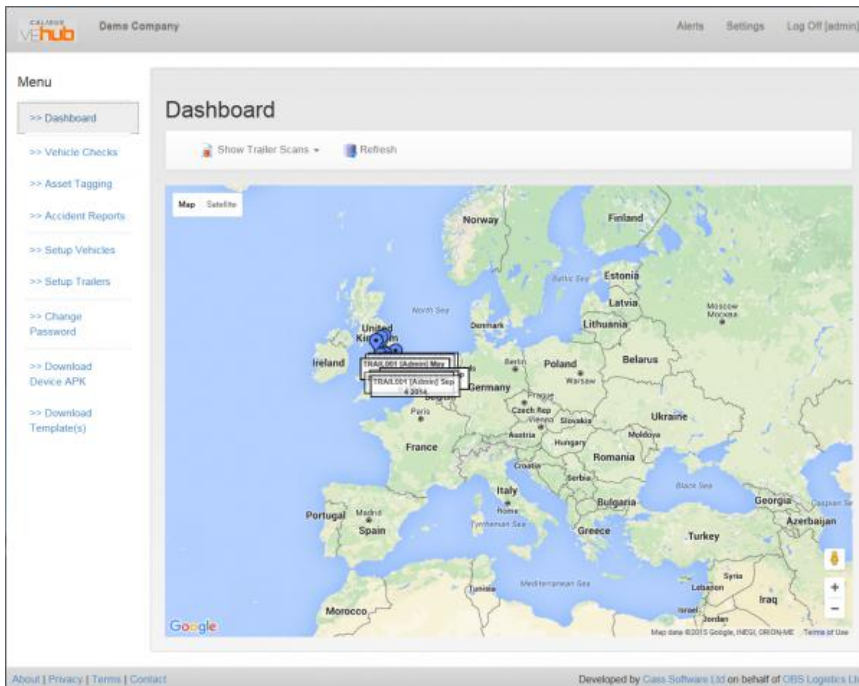
General Screen options

Header:

Ready for **What's Next, Now**™

- Alerts
- Settings
- Log Off

Each screen shows a navigation menu:

- Dashboard - this takes you to the Dashboard screen, and is the first screen shown after logging on.
- Vehicle Checks
- Asset Tagging
- Accident Reports
- Setup Vehicles
- Setup Trailers
- Change Password
- Download Device APK - used to download the application for installing to an Android device.
- Download Templates

2.2 Alerts

Alerts Screen

- Add "Vehicle Check" Alert
- Add "Accident Report" Alert
- Add "Trailer Scan" Alert
- Add "Lapsed Check" Alert

General Alerts properties

Alerts generally allow specification of:
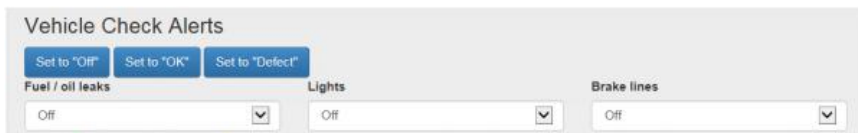
- The Alert Name
- whether the alert is enabled
- the frequency of the alert (Immediate or Daily)
- The Email address

Accident Report alerts may be set against specific entered fields, standard or custom.

Adding Accident Report Alerts

Ready for **What's Next, Now**™

Accident Report alerts may be set against the specific submitted Vehicle or Trailer. Additionally, customer fields may be checked.

Adding Vehicle Check Alerts

Trailer Scan alerts may be set against the specific trailer, or the Scan Code
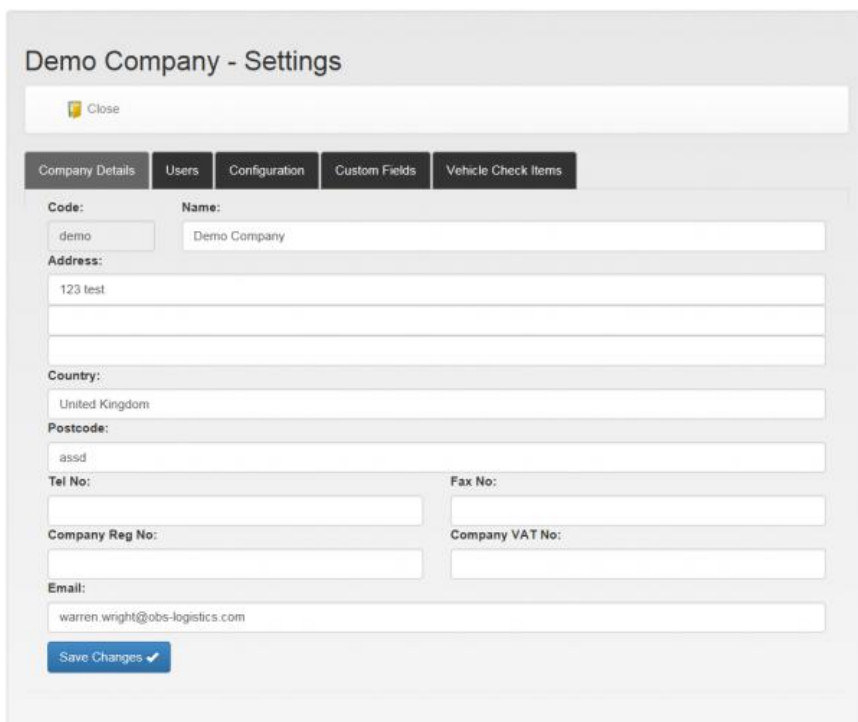
Adding Trailer Scan Alerts

Lapsed Check alerts may be set against the specific vehicle, and may be checked at specific days of the week, and a specific time.

2.3 Settings

2.3.1 Company Details

The screen allows you to amend the Company details.

Company Settings Screen

Setting that may be modified are:

- Code
- Name - Displayed in the title
- Address - up to 3 lines
- Country

Ready for **What's Next, Now**™

- Postcode
- Tel No
- Fax No
- Company Reg No
- Company VAT No
- Email

When changes are complete, click the **Save Changes** button.

2.3.2 Users

This screen allows the users of the system to be maintained.

Users Settings Screen

The screen initially shows a table of all users currently on the system. Each user may be edited or deleted using the **Edit** or **Delete** buttons respectively against the line.

A new user may be added using the **Add New User** button.

Adding a new User

When editing or adding a new user, the edit user screen will be displayed, allowing entry of:

Ready for **What's Next, Now**™

- Username
- Password
- Title
- First Name
- Middle Name(s)
- Surname
- Address - up to 3 lines
- Country
- Postcode
- Permission (Admin, User or Read Only)
- Allow Access to web - Yes or No
- Allow Access to device - Yes or No

The screen also shows when the user was last seen (Last Logged In and Last Ping).

2.3.3 Configuration

This screen allows maintenance of system settings.

Configuration Settings Screen

A setting may be edited with the **Edit** button. This allows changing of the configuration flag.

Editing Configuration

When changes are complete, click the **Save** button. Changes may be discarded with the **Close** button.

Configurable elements are:

- Enable Vehicle Check Alerts - whether to email alerts for vehicle checks with defects.

Ready for **What's Next, Now**™

- Send Vehicle Check Alerts To.. - Email address for vehicle checks with defects.
- Enable Accident Alerts - whether to email alerts for accident reports received.
- Send Accident Alerts To.. - Email address for accident reports received.
- GPS Required - whether GPS connectivity is required before any data captured on the device is recorded.
- Next Check for Unchecked Vehicles at... Default start date for any new vehicles to be checked.
- Enable Unchecked Vehicle Alerts - whether to email alerts for vehicles that have not been checked.
- Send Overdue Vehicle Check Alerts To - Email address for overdue vehicle checks.

2.3.4 Custom Fields

This screen allows custom Vehicle Check and Accident Report fields to be maintained.

Custom Fields Settings Screen

Custom Fields may be edited using the **Edit** button against each field, and may be deleted through the **Delete** button against each field.

New Custom Fields of each type may be added using the appropriate **Add New Field** button in each section.

Ready for **What's Next, Now**™

Adding Custom Fields

When Editing or Adding new fields, the Edit Screen is displayed, and allows editing of:

- Label
- Display Order
- Field Type - one of:
    - Single Line Text
    - Multi Line Text
    - Drop Down
    - True/False
    - Date
    - Time
    - Numeric
    - Radio Options
    - Toggle with Photo
- Mandatory - Yes/No

Depending on the Field Type, data may be allowed for the field.

Custom Data may be edited using the **Edit** button against each field, and may be deleted through the **Delete** button against each field.

New Custom Data may be added using the appropriate **Add New Field** button.

Adding Data to Custom Fields

When Editing or Adding new data, the Edit Screen is displayed, and allows editing of:

- Display Label
- Allow Free Type (Other)
- Data Value
- Display Order

With either screen, when changes are complete, click the **Save** button. Changes may be discarded with the **Close** button.

Ready for **What's Next, Now**™

2.3.5 Vehicle Check Items

This screen allows the Vehicle Check items to be maintained.
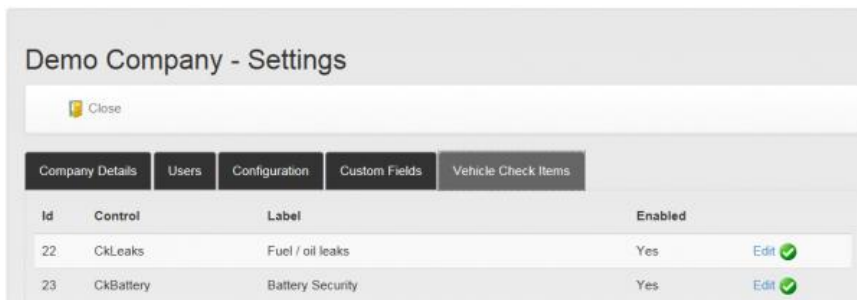
Vehicle Check Items Settings Screen

Each control may be edited using the **Edit** button against the line. This shows the following:

- Control Name (May not be amended)
- Label
- Enabled - Yes/No

When changes are complete, click the **Save** button. Changes may be discarded with the **Close** button.

2.4 Dashboard Screen

The Dashboard screen shows the location of all actions taken by users within your company.
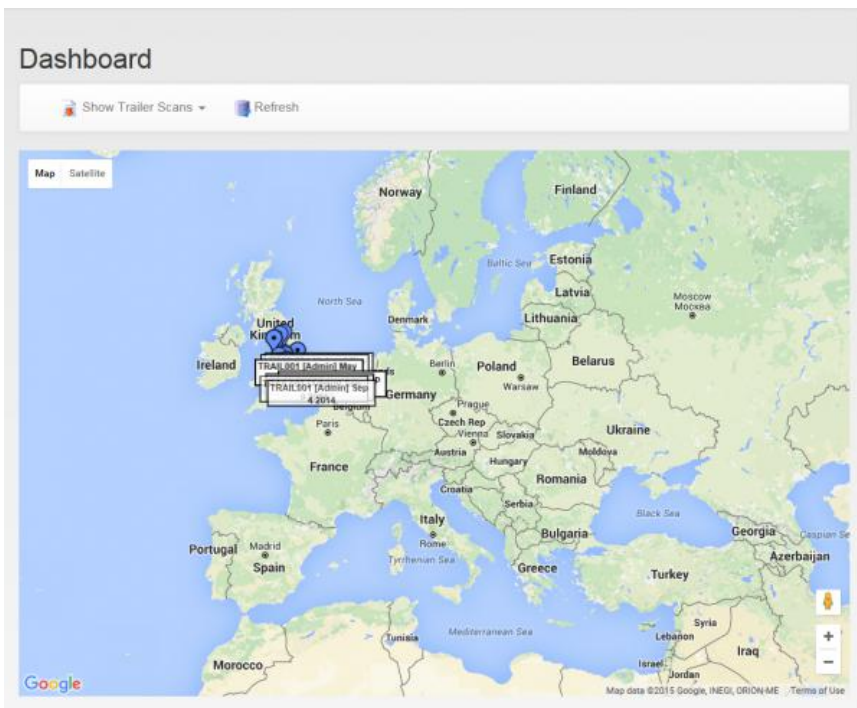
Dashboard Screen

By default, the screen shows all Trailer Scans. This may be changed using the filter provided. When the filter is changed, the screen will refresh automatically.

The Refresh button will redisplay all the selected events.

The map may be zoomed in and out using the mouse scroll button, or the buttons provided in the bottom right of the map.

Clicking on an event will show a pop-out label, showing the some details of the event.

Ready for **What's Next, Now**™

2.5 Vehicle Checks Screen

This screen allows the Vehicle Checks completed for this company to be maintained.

Vehicle Checks Screen

The screen first displays a list of all the vehicle checks, showing:

- ID
- Checked By
- Date Checked
- Vehicle
- Trailer
- Odometer
- Defects - whether there were defects
- Rectified - whether defects were rectified
- Rectified By - who rectified the defects

Selecting a line will highlight it, then an action can be taken against that line. the actions are shown on the top of the screen:

- **Open Record** - shows the details of the checks in a Vehicle Check Details screen
- **Delete Record** - delete the line

The screen also allows some general functions:

- **Print** - print the full list of checks
- **Refresh** - refresh the list
- **Import** - import the list of checks
- **Export To** - export the list of checks

Importing and Exporting of checks can be done in several formats:

- XML
- CSV
- JSON

2.5.1 Vehicle Check Details Screen

The screen shows all the details captured from the vehicle checks, and also allows comments and rectifying defects.

The screen shows several sections:

- Details

Vehicle Checks Details - Details and Checks Section

- ♦ Driver's Name
  - ♦ Date
  - ♦ Vehicle
  - ♦ Trailer Fleet/Serial #
  - ♦ Odometer
- Checks
  - ♦ Any standard checks that were completed

- Reported Defects

Defects Section

- ♦ Defects
  - ♦ Rectified By
  - ♦ Rectified Date
  - ♦ Rectified Notes
  - ♦ Signature

- Comments

Ready for **What's Next, Now**™

Comments Section

- ♦ Author
  ♦ Date
  ♦ Comment

New comments may be added with the **Add Comment** button.

Adding a Comment

Once a comment is added, this may be saved with the **Save** button.

- • Custom Data - any customer checks configured for the company.

Custom Data Section

- • Location - A map of the location where the vehicle checks were completed (if provided).

Location Section

- • Images - Any images associated to the standard checks.

Ready for **What's Next, Now**™

Images Section

Action may be taken through the buttons provided on the top of the screen:

- **Print** - print the screen
- **Export To** - Export the checks to XML, CSV or JSON format.
- **Close** - Close the screen and return to the Vehicle Checks screen.

2.6 Asset Tagging Screen

This screen allows the Vehicle Checks completed for this company to be maintained.

Asset Tagging Screen

The screen first displays a list of all the vehicle checks, showing:

- ID
- Trailer Code
- Trailer Name
- Scanned By
- Scan Date
- Scan Value

Selecting a line will highlight it, then an action can be taken against that line. the actions are shown on the top of the screen:

- **Open Record** - shows the details of the checks in an Asset Tagging Details screen
- **Delete Record** - delete the line

Ready for **What's Next, Now**™

The screen also allows some general functions:

- **Print** - print the full list of checks
- **Refresh** - refresh the list
- **Import** - import the list of checks
- **Export To** - export the list of checks

Importing and Exporting of checks can be done in several formats:

- XML
- CSV
- JSON

2.6.1 Asset Tagging Details Screen

The screen shows all the details captured from the asset tagging.

Asset Tagging Details Screen

The screen also allows some general functions:

- **Print** - print the details
- **Export To** - export the Asset Tagging details to XML, CSV or JSON
- **Close** - return to the Asset Tagging screen

2.7 Accident Reports Screen

This screen allows the Accident Reports completed for this company to be maintained.

Ready for **What's Next, Now**™

Accident Reports Screen

The screen first displays a list of all the vehicle checks, showing:

- ID
- Raised By
- Date Raised
- Vehicle
- Trailer
- Location
- Weather
- Modified By
- Modified Date

Selecting a line will highlight it, then an action can be taken against that line. the actions are shown on the top of the screen:

- **Open Record** - shows the details of the checks in an Accident Reports Details screen
- **Delete Record** - delete the line

The screen also allows some general functions:

- **Print** - print the full list
- **Refresh** - refresh the list
- **Import** - import a list
- **Export To** - export the list

Importing and Exporting can be done in several formats:

- XML
- CSV
- JSON

2.7.1 Accident Reports Details Screen

The screen shows all the details captured from the Accident Reports.

Ready for **What's Next, Now**™

The screen displays in several sections:

- Accident Details

Accident Reports Details Screen

- - ◆ Accident Date
    - ◆ Weather/Road Conditions
    - ◆ Location of Accident
    - ◆ Accident Details

- Damage Descriptions

Damage Descriptions section

💡 **Note:** Data can be entered for Your and the other Vehicle:

- - ◆ Damage
    - ◆ Injuries
    - ◆ Passengers
    - ◆ Towing Company Name/Phone

- Other Driver/Vehicle Information

Other Driver/Vehicle Information section

- - ◆ Owner's Name
    - ◆ Owner's Address
    - ◆ Owner's Phone
    - ◆ Vehicle Reg
    - ◆ Make

Ready for **What's Next, Now**™

- ◆ Model/Year
- ◆ Colour
- ◆ Insurance Company
- ◆ Agent Name & Phone
- ◆ Other Driver's Name
- ◆ Other Driver's Address
- ◆ Other Driver's Phone

- Police Information

Police Information section

-
  - ◆ Officer Name
  - ◆ Department
  - ◆ Phone
  - ◆ Badge #
  - ◆ Other Info

- Witness Information

Witness Information section

-
  - ◆ Name
  - ◆ Address
  - ◆ Home Phone
  - ◆ Work Phone

- Custom Data

Custom Data section

- Comments

Ready for **What's Next, Now**™

Comments section

- ♦ ID
  ♦ Author
  ♦ Date
  ♦ Comment

A new comment may be added by clicking the **Add Comment** button, which will display the Add Comment screen.

Add Comment screen

Once completed, click **Save** to save the comment and return to the Accident Report Details screen. To discard entry and return to the Accident Report Details screen, click **Close**.

- Location - A map of the location where the Accident Report was completed (if provided).

Location Section

- Images - Any images associated to Accident Report.

Images Section

The screen also allows some general functions:

- **Print** - print the details
- **Export To** - export the details to XML, CSV or JSON
- **Close** - return to the Accident Reports screen

Ready for **What's Next, Now**™

2.8 Setup Vehicles

This screen allows the maintenance of Vehicles for this company.

Setup Vehicles Screen

The screen first displays a list of all the vehicles, showing:

- ID
- Vehicle Reg
- Make
- Model

Selecting a line will highlight it, then an action can be taken against that line. the actions are shown on the top of the screen:

- **Add New** - allows a vehicle to be added through an Add Vehicles screen
- **Open Record** - shows the details of the vehicle in an Edit Vehicles screen
- **Delete Record** - delete the line

The screen also allows some general functions:

- **Refresh** - refresh the list
- **Print** - print the full list
- **Import** - import a list
- **Export To** - export the list

Importing and Exporting can be done in several formats:

- XML
- CSV
- JSON

Ready for **What's Next, Now**™

Add Vehicles Screen

Edit Vehicles Screen

2.9 Setup Trailers

This screen allows the maintenance of Trailers for this company.

Ready for **What's Next, Now**™

Setup Trailers Screen

The screen first displays a list of all the trailers, showing:

- ID
- Vehicle Reg
- Make
- Model

Selecting a line will highlight it, then an action can be taken against that line. the actions are shown on the top of the screen:

- **Add New** - allows a trailer to be added through an Add Trailers screen
- **Open Record** - shows the details of the trailer in an Edit Trailers screen
- **Delete Record** - delete the line

The screen also allows some general functions:

- **Refresh** - refresh the list
- **Print** - print the full list
- **Import** - import a list
- **Export To** - export the list

Importing and Exporting can be done in several formats:

- XML
- CSV
- JSON

Ready for **What's Next, Now**™
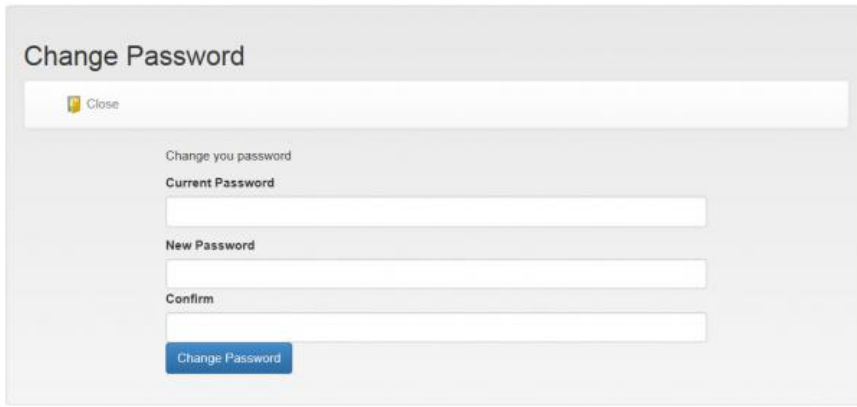
Add Trailers Screen

Edit Trailers Screen

Edit Trailers - QR Section

Edit Trailers - Locations section

Ready for **What's Next, Now**™

2.10 Change Password
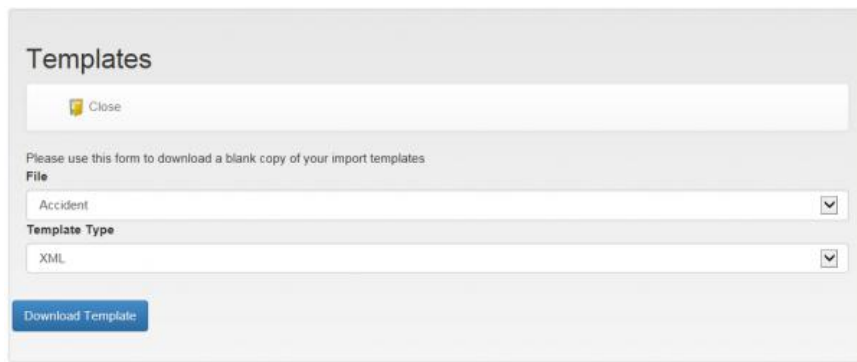
Change Password Screen

2.11 Download Template(s)

Use this form to download a blank copy of your import templates.

Templates Screen

File:

- Accident
- Trailer
- Trailer Scan
- Vehicle
- Vehicle Check

Type:

- XML
- CSV
- JSON

Ready for **What's Next, Now**™

3 Appendix A: Document References

A.1 References

Ref No	Document Title & ID	Version	Date
1			

A.2 Glossary

Term	Definition
EPOD	Electronic Proof of Delivery. The OBS EPOD system is *CALIDUS* ePOD.
CALIDUS eSERV	The OBS mobile system to complete Service functionality in the field. This is part of the *CALIDUS* ePOD system.
PDA	The mobile device on which the C-ePOD system will run in the field. This can be a Phone, EDA or industrial PDA, running Android.
DAL	Data Access Layer. A mechanism for accessing data by the system that is removed from the application, allowing for simplified access and providing protection to the data, as only approved DAL methods can be used to modify it.
GPS	Global Positioning System. A mechanism of retrieving accurate positioning information in the form of Latitude and Longitude (Lat-Long) co-ordinates from a device.
GPRS, 3G, HSDPA, Data Service	All terms referring to mobile device network connectivity, and the speed at which the device connects to the internet.

A.3 Authorised By

Murray Middleton   OBS Project Manager   _____

Tony Walker   OBS Consultant   _____

Ready for **What's Next, Now**™

4 Functional Overview

The document is intended to describe the basic functionality of the *CALIDUS* VEhub mobile device application.

Ready for **What's Next, Now**™

5 *CALIDUS* VEhub App

5.1 Login

VEhub Login

Enter a valid Customer ID, User Name and Password.

Click **Sign In** to go to the Main Screen, or Setup to configure the application.

5.2 Setup

This screen allows configuration of the communications for the application.
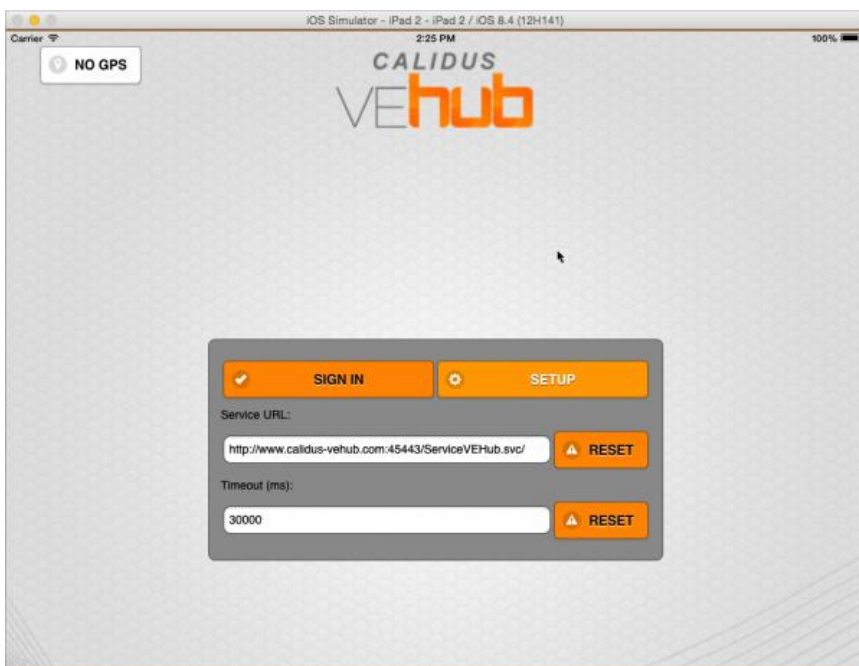
App Configuration

Ready for **What's Next, Now**™
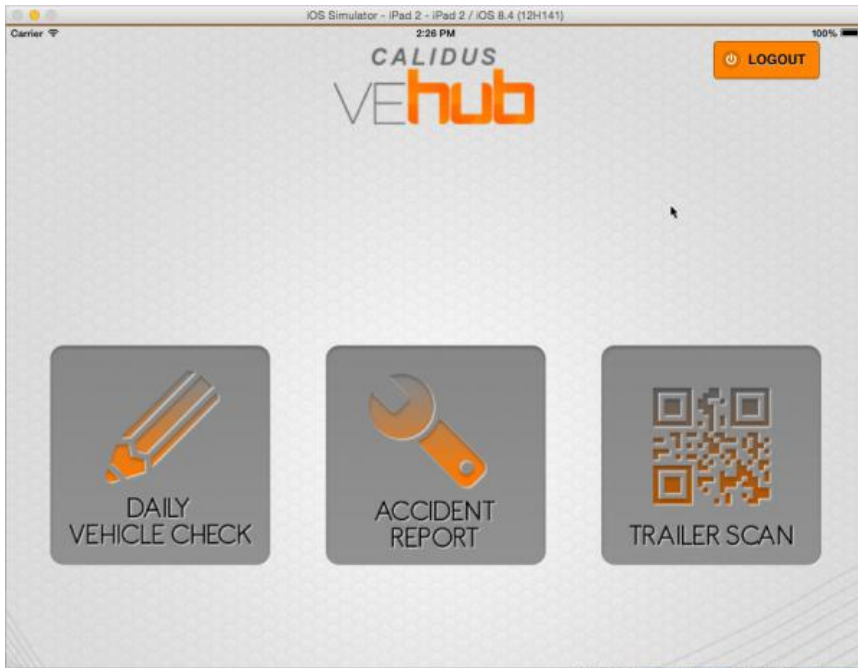
5.3 Main Screen

CALIDUS VEhub Main Screen

From here, the following 3 options may be selected:

- Vehicle Checks
- Accident Report
- Trailer Scan

5.4 Vehicle Checks

Page 1: Basic Details

Vehicle Checks Page 1

Ready for What's Next, Now™

Buttons:

- **Home** - return to the Main Screen.
- **Notes** - show the last submitted vehicle check
- **Trash** - reset all entered vehicle check information to this point, and start again. The device will prompt for confirmation first.
- **Confirm** - move to the next stage

Fields:

- Driver's Name
- Date
- Vehicle - enter through drop-down list of vehicles set up for this company
- Trailer - enter through drop-down list of trailers set up for this company
- Odometer - numeric-only

Page 2/3: Basic Checks

Vehicle Checks Page 2

Ready for **What's Next, Now**™

Vehicle Checks Page 3

Check boxes for all the configured basic checks.

Buttons:

- **Home** - return to the Main Screen.
- **Back** - return to the previous screen.
- **Confirm** - move to the next stage

Each item may be checked as Passed (a tick) or Failed (a cross).

If failed, an image may be taken, through a popup button for images.

Page 4: Defects

Vehicle Checks Page 4

Buttons:

Ready for **What's Next, Now**™

- **Home** - return to the Main Screen.
- **Back** - return to the previous screen.
- **Confirm** - move to the next stage
- **Additional Info** - enter any custom fields set up against the vehicle checks

Fields:

- Defects

Page 5: Signature
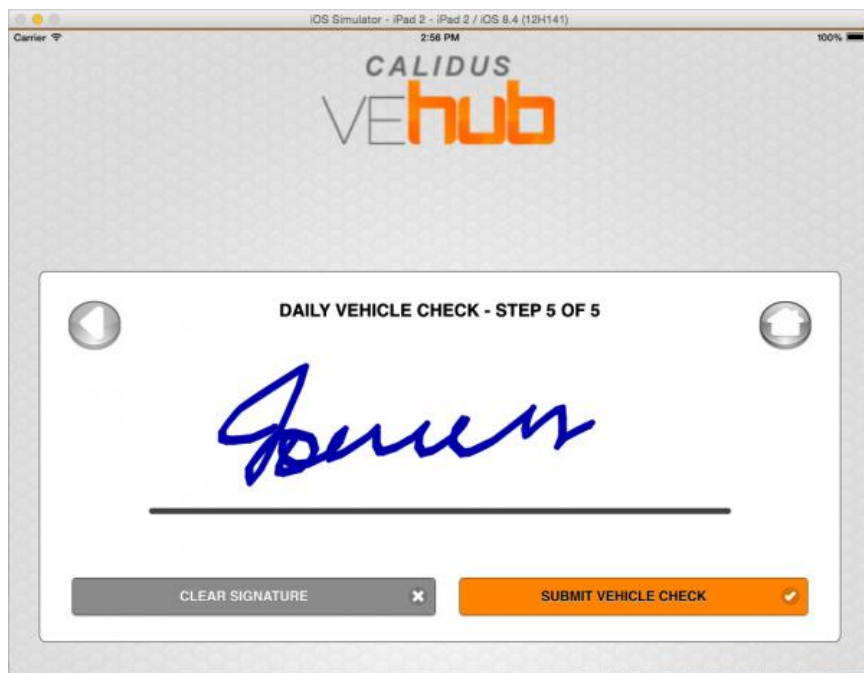
Vehicle Checks Page 5

- **Home** - return to the Main Screen.
- **Back** - return to the previous screen.
- **Clear Signature** - clear the entered signature
- **Submit Vehicle Check** - save the vehicle checks

Once completed, the checks will be sent back and stored in the system.

The last submitted vehicle checks are visible from the **Notes** button on the first page:

Ready for **What's Next, Now**™

Viewing the Last Submitted Vehicle Checks

5.5 Accident Report

Page 1: Basic Details

Accident Report Basic Details

Buttons:

- **Home** - return to the Main Screen.
- **Trash** - reset all entered vehicle check information to this point, and start again. The device will prompt for confirmation first.
- **Confirm** - move to the next stage

Fields:

Ready for **What's Next, Now**™

- Location - In addition to GPS data
- Weather/Conditions
- Details

Page 2: Damage Descriptions (1)

Damage Descriptions

Buttons:

- **Home** - return to the Main Screen.
- **Back** - return to the previous screen.
- **Confirm** - move to the next stage

Fields:

- Vehicle - enter through drop-down list of vehicles set up for this company
- Trailer - enter through drop-down list of trailers set up for this company
- Your Vehicle Damage
- Other Vehicle Damage

Page 3: Damage Descriptions (2)

Ready for **What's Next, Now**™

Damage Descriptions

Buttons:

- **Home** - return to the Main Screen.
- **Back** - return to the previous screen.
- **Confirm** - move to the next stage

Fields:

- Your Towing Company
- Your Towing Company Tel
- Other Towing Company
- Other Towing Company Tel

Page 4: Other Driver / Vehicle Info (1)

Other Driver / Vehicle Info

Ready for **What's Next, Now**™

Buttons:

- **Home** - return to the Main Screen.
- **Back** - return to the previous screen.
- **Confirm** - move to the next stage

Fields:

- Other Owner Name
- Other Owner Address
- Other Vehicle Model
- Other Vehicle Reg
- Other Owner Phone
- Other Vehicle Make
- Other Vehicle Colour

Page 5: Other Driver / Vehicle Info (2)

Other Driver Insurance Details

Buttons:

- **Home** - return to the Main Screen.
- **Back** - return to the previous screen.
- **Confirm** - move to the next stage

Fields:

- Other Vehicle Insurance
- Other Driver Name
- Other Driver Address
- Other Vehicle Agent Tel
- Other Driver Phone

Page 6: Passengers/Injuries

Passengers/Injuries

Buttons:

- **Home** - return to the Main Screen.
- **Back** - return to the previous screen.
- **Confirm** - move to the next stage

Fields:

- Your Vehicle Passengers
- Your Vehicle Injuries
- Other Vehicle Passengers
- Other Vehicle Injuries

Page 7: Police & Witness Information (1)

Police & Witness Information

Ready for **What's Next, Now**™

Buttons:

- **Home** - return to the Main Screen.
- **Back** - return to the previous screen.
- **Confirm** - move to the next stage
- **Additional Info** - enter any custom fields set up against the Rccident Report, through a popup window.

Fields:

- Police Name
- Police Dept
- Police Badge
- Police Phone
- Police Other Info
- Witness 1 Name
- Witness 1 Home Phone
- Witness 1 Work Phone

Additional Info

Page 8: Police & Witness Information (2)

Ready for **What's Next, Now**™

Police & Witness Information

Buttons:

- **Home** - return to the Main Screen.
- **Back** - return to the previous screen.
- **Take Photos** - Take and store photos
- **Submit Report** - Save the Accident Report

Fields:

- Witness 1 Address
- Witness 2 Name
- Witness 2 Home Phone
- Witness 2 Address
- Witness 2 Work Phone

The screen will validate that all information is entered correctly before submitting the report.

Once completed, the report will be sent back and stored in the system.

5.6 Trailer Scan

Ready for **What's Next, Now**™
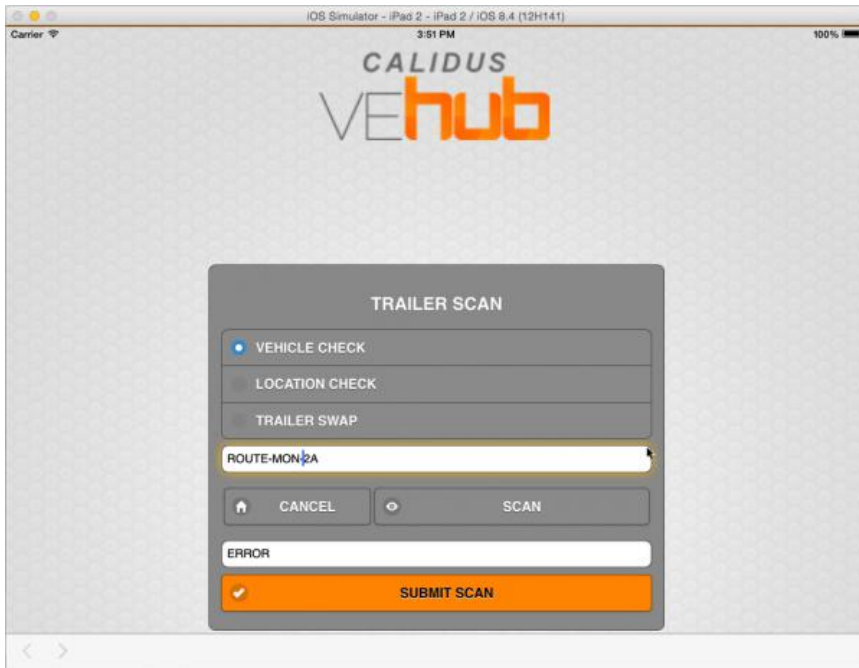
Trailer Scan Screen

Fields:

- Type - select from
  - ♦ Vehicle Check
  - ♦ Location Check
  - ♦ Trailer Swap
- Load Ref

Buttons:

- **Cancel** - return to the main screen
- **Scan** - Scan a QR barcode
- **Submit Scan** - submit the scan and save it on the system

Ready for **What's Next, Now**™

6 Appendix A: Document References

A.1 References

Ref No	Document Title & ID	Version	Date
1			

A.2 Glossary

Term	Definition
EPOD	Electronic Proof of Delivery. The OBS EPOD system is *CALIDUS* ePOD.
CALIDUS eSERV	The OBS mobile system to complete Service functionality in the field. This is part of the *CALIDUS* ePOD system.
PDA	The mobile device on which the C-ePOD system will run in the field. This can be a Phone, EDA or industrial PDA, running Android.
DAL	Data Access Layer. A mechanism for accessing data by the system that is removed from the application, allowing for simplified access and providing protection to the data, as only approved DAL methods can be used to modify it.
GPS	Global Positioning System. A mechanism of retrieving accurate positioning information in the form of Latitude and Longitude (Lat-Long) co-ordinates from a device.
GPRS, 3G, HSDPA, Data Service	All terms referring to mobile device network connectivity, and the speed at which the device connects to the internet.

A.3 Authorised By

Murray Middleton   OBS Project Manager   _____

Tony Walker   OBS Consultant   _____

Ready for **What's Next, Now**™

7 VEhub Admin Release Process

VEHub Server Release Process

7.1 INFO:

1. VEHub code can be found in the repository: https://dev.azure.com/Aptean/Calidus/_git/Vehub
2. VEHub test web server is 172.183.45.55 (requires VPN, Runcorn, for access).
3. VEHub test database is on 172.164.45.205 (connect from local PC with VPN using MS SSMS).
4. VEHub live web server is the shared web (172.183.45.50)
5. VEHub live database is on 172.164.45.200 (connect from local PC with VPN using MS SSMS).
6. VS = Microsoft Visual Studio (currently using version 2019 for server dev, incl. builds)
7. MS SSMS = Microsoft SQL Server Management Studio
8. Anywhere in this document that refers to _0.0.0.0_ this should be replaced by/read as the version number of the existing installation (i.e. the version on the target server before the release).
9. Anywhere in this document that refers to _9.9.9.9_ this should be replaced by/read as the version number being built/released, i.e. the new version number (after it has been incremented in step 2 in the Dev/Changes section below).
10. Anywhere in this document that refers to _env_vers_, env should be replaced by/read as the environment (Test/Live) being released to and vers should be replaced by/read as the version being released in the format xxxxxxxx with each portion of the version being padded to 2 digits (for example if releasing version 2.5.0.4 to test env_vers should be replaced by/read as Test_02050004).
11. Anywhere in this document that refers to _env_versold_, env should be replaced by/read as the environment (Test/Live) being released to and vers should be replaced by/read as the version being replaced in the format xxxxxxxx with each portion of the version being padded to 2 digits (for example if releasing to test, replacing version 2.0.5.3, env_versold should be replaced by/read as Test_02000503).
12. Examples in this document will assume releasing version 2.0.5.4, replacing an existing version 2.0.5.3, into the test environment in the example.

7.2 Dev/Changes:

Before building/releasing the following should be completed

1. Any code changes should be completed and committed/merged into the ?VehubDev? branch.
2. If the code changes did not include incrementing the version this should be incremented now by incrementing the following:
    1. Version number in AssemblyVersion and AssemblyFileVersion settings in the AssemblyInfo.cs files, found in the properties sub-folder, in the ?AdminTool > VEHub.admin?, ?Core > VEHub.Core?, ?Services > Emailer > Emailer?, ?Services > TomTomLink > TomTomLink?, ?Services > Poller > VEHub.WindowsService?, ?WcfService > VEHub.service? and ?WebPortal > VehubUI? projects.
    2. The version number portion of the publishUrl value in the ?WcfService > VEHub.WindowsService > Properties > PublishProfiles > Local.pubxml? file and the ?WebPortal > VehubUI > Properties > PublishProfiles > Local.pubxml? file.
3. All of the required changes, including the version number increment should be merged from the ?VehubDev? branch into the ?VehubQa? branch.
    1. **Note** Any changes to ?config? files are not included in a release but ideally, we should keep settings used for test/live servers in QA branch whereas we may have settings for dev environment in Dev branch.

7.3 Build:

These steps are performed on your local PC and need to be performed ahead of any release after all steps in the Dev/Changes section above have been completed.

1. Create a new folder called ?VEHub _9.9.9.9_?. Suggest creating this in C:\Vehub Builds. This VEHub _9.9.9.9_ folder is the **release folder**.
2. Publish the VEHub.UI project (from the QA branch) in VS.
    1. Publish the WebPortal > VEHub.UI project by right clicking the VEHub.UI project in the Solution Explorer section and selecting the Publish option. In the window this opens ensure the Local.pubxml file is selected (drop-down near the top) and click the Publish button. This publishes to the **UI Publish** folder (should be C:\Vehub Builds\VEhub.UI 9.9.9.9, the actual folder used is taken from the Local.pubxml file

    and is displayed in the Output window).
2. Delete the web.config from the **UI publish** folder.
3. Copy/move the **UI publish** folder into the release folder.

3. Publish the VEHub.service project
    1. Publish the WcfService > VEHub.Service project by right clicking the VEHub.Service project in the Solution Explorer section and selecting the Publish option. In the window this opens ensure the Local.pubxml file is selected (drop-down near the top) and click the Publish button. This publishes to the Service Publish folder (should be C:\Vehub Builds\VEhub.Service _9.9.9.9_, the actual folder used is taken from the Local.pubxml file and is displayed in the Output window).
    2. Delete the web.config file from the service publish folder.
    3. Copy/move the Service publish folder into the release folder.

4. Build the Emailer project
    1. Create a local emailer build folder called VEhub.Emailer _9.9.9.9_ (e.g. C:\Vehub Builds\VEhub.Emailer 2.0.5.4). This is the **Emailer Build** folder.
    2. Build the Service > Emailer > Emailer project by right clicking the Emailer project in the Solution Explorer section and selecting the Build option. Take note of the configuration (Release or Debug) used (shown in drop-down on toolbar). Open the folder in file explorer (right click the Emailer project and select the ?OpenFolder in File Explorer? option), go into the bin folder. From there go into the Release or Debug folder (depending on which configuration was used, you can check the date/time of the Emailer.application or Emailer.exe files to make sure this is the correct folder), this folder is also shown in the output window. Copy all of the contents of this Release/Debug folder into the **Emailer Build** folder.
    3. Delete the Emailer.exe.config file in the **Emailer Build** folder.
    4. Copy/move the **Emailer Build** folder into the release folder.

5. Build the Vehub.WindowsService project (i.e. the poller project)
    1. Create a local poller build folder called VEhub.Poller 9.9.9.9 (e.g. C:\Vehub Builds\VEhub.Poller 2.0.5.4). This is the **Poller build** folder.
    2. Build the Services > Poller > VEHub.WindowService project by right clicking the VEHub.WindowService project in the Solution Explorer section and selecting the Build option. Take note of the configuration (Release or Debug) used (shown in drop-down on toolbar). Open the folder in file explorer (right click the Emailer project and select the ?OpenFolder in File Explorer? option), go into the bin folder. From there go into the Release or Debug folder (depending on which configuration was used, you can check the date/time of the VEHub.WindowsService.exe file to make sure this is the correct folder), this folder is also shown in the output window. Copy all of the contents of this Release/Debug folder into the **Poller build** folder.
    3. Delete the VEHub.WindowsService.exe.config file in the **Poller build** folder.
    4. Copy/move the **Poller build** folder into the release folder.

6. Zip the release folder to create a VEHub _9.9.9.9_.zip file, this is the **release Zip file**.
7. Copy the **release Zip file** into the \\DGA1FS01OBS\Projects\Product\VEHUB\5. Releasing\Server\Releases directory on the network drives.

7.4 Release:

```
These steps are mainly performed on the target (test or live) web server and need to be performed ahead of
```

1. Copy the **release ZIP file** for the version to be released onto the target server. There is a ?Release working folder? on the same level as the IIS folders that can be used. Note you may have to copy the zip file onto the desktop then move it to this directory.
2. Extract the release Zip file to create a **release folder**.
3. Release the UI programs
    1. Create a directory called VEHub.UI._env_vers_ at the same level as the existing UI site. To do this right click the existing UI site in IIS manager and select the Explore option to open the existing directory in file explorer. Go up one directory and create the new VEHub.UI._env_vers_ directory there, i.e. alongside the existing directory. For example create a new C:\inetpub\wwwroot\VEHub.UI.Test_02050004 directory.
    2. Copy the Web.config from the existing UI directory into the newly created UI directory. For example copy the web.config file from the existing C:\inetpub\wwwroot\VEHub.UI.Test_02050003 directory to the new C:\inetpub\wwwroot\VEHub.UI.Test_02050004 directory.
    3. Copy all contents of the VEHUb.UI folder (in the **release folder**) into to the new VEHub.UI._env_vers_ folder.
4. Create the UI application pool
    1. In the IIS manager right-click on the Application Pools ad select the ?Add Application Pool? option.
    2. In the opened pop-up set the Name to VEHUB.UI._env_vers_
    3. Click the OK button

Ready for **What's Next, Now**™

1. Create the UI IIS site
    1. In IIS Manager right-click the Sites folder and select the ?Add website?? option.
    2. Set the Site Name to VEHub.UI._env_vers_
    3. Check the application pool is set to the VEHUB.UI.env_vers application pool created in step 3 above.
    4. Set the Physical path to the directory created in step 2a above
    5. In the binding section set the Type to http, set the IP Address to All Unassigned and set the Port to any unused port number (suggest 2099).
    6. Uncheck the ?Start Website immediately? checkbox.
    7. Click OK

1. Release the Service programs
    1. Create a directory called VEHub.Service._env_vers_ at the same level as the existing Service site. To do this right click the existing Service site in IIS manager and select the Explore option to open the existing directory in file explorer. Go up one directory and create the new VEHub.Service._env_vers_ directory there, i.e. alongside the existing directory. For example create a new C:\inetpub\wwwroot\VEHub.Service.Test_02050004 directory.
    2. Copy the Web.config from the existing Service directory into the newly created Service directory. For example copy the web.config file from the existing C:\inetpub\wwwroot\VEHub.Service.Test_02050003 directory to the new C:\inetpub\wwwroot\VEHub.Service.Test_02050004 directory.
    3. Copy all contents of the VEHUb.Service folder (in the **release folder**) into to the new VEHub.UI._env_vers_ folder.
2. Create the Service application pool
    1. In the IIS manager right-click on the Application Pools ad select the ?Add Application Pool? option.
    2. In the opened pop-up set the Name to VEHUB.Service.env_vers
    3. Click the OK button

1. Create the Service IIS site
    1. In IIS Manager right-click the Sites folder and select the ?Add website?? option.
    2. Set the Site Name to VEHub.Service._env_vers_
    3. Check the application pool is set to the VEHUB.Service._env_vers_ application pool created in step 6 above.
    4. Set the Physical path to the directory created in step 5a above
    5. In the binding section set the Type to http, set the IP Address to All Unassigned and set the Port to any unused port number (suggest 45499).
    6. Uncheck the ?Start Website immediately? checkbox.
    7. Click OK

1. Create Emailer service
    1. Create a folder called VEHub.Emailer._env_vers_ within the C:\VEHub Services directory
    2. Copy the Emailer.exe.config from the existing Service directory into the newly created Service directory. For example copy the Emailer.exe.config file from the existing C:\VEHub Services\VEHub.Emailer.Test_02000503 directory to the new C:\VEHub Services\VEHub.Emailer.Test_02000504 directory.
    3. Copy all contents of the VEHUb.Emailer folder (in the **release folder**) into to the new VEHub.Emailer._env_vers_ folder.
    4. Run the following command (using an elevated command prompt, i.e. run command prompt as administrator):

`sc.exe create VEHub.Emailer.env_vers binpath= "C:\VEHub Services\VEHub.Emailer.env_vers\Emailer.exe"`

1.   1. Again using an elevated command prompt, run the following command:

`sc.exe description VEHub.Emailer.env_vers "Sends VEHub emails using release env_vers"`

1.   1. Open the services app and check the service for name VEHub.Emailer.env_vers has been created and has the Startup Type set to Manual and is not running (Status is blank).
2. Open the services app and check the service for name VEHub.Emailer._env_vers_ has been created and has the Startup Type set to Manual and is not running (Status is blank).
3. Create Poller service

1. Create a folder called VEHub.Poller._env_vers_ within the C:\VEHub Services directory
2. Copy the VEHub.WindowsService.exe.config from the existing Service directory into the newly created Service directory. For example copy the VEHub.WindowsService.exe.config file from the existing C:\VEHub Services\VEHub.Poller.Test_02000503 directory to the new C:\VEHub Services\VEHub.Poller.Test_02000504 directory.
3. Copy all contents of the VEHUb.Poller folder (in the release folder) into to the new VEHub.Poller._env_vers_ folder.
4. Run the following command (using an elevated command prompt, i.e. run command prompt as administrator):

```
`sc.exe create VEHub.Poller.env_vers binpath= "C:\VEHub Services\VEHub.Poller.env_vers\VEHub.WindowsService
```

1.   1. Again using an elevated command prompt, run the following command:

```
`sc.exe description VEHub.Poller.env_vers "Checks for overdue VEHub vehicle checks using release env_vers"`
```

1. Open the services app and check the service for name VEHub.Poller._env_vers_ has been created and has the Startup Type set to Manual and is not running (Status is blank).
2. Prepare DB Scripts incl. Rollback script
    1. Create a new folder, called Release Scripts inside the release folder. This new folder is the release scripts folder.
    2. Find any update script folders for any versions included by this release. For our example of releasing version 2.0.5.4, replacing 2.0.5.3, this would include any folder for version 2.0.5.4. This should include folders for any version higher than the existing version up to and including the release version. So if the existing version was 2.0.5.1 (instead of 2.0.5.3, but still releasing version 2.0.5.4) then this would include any folders for versions 2.0.5.2, 2.0.5.3 and 2.0.5.4. These folders can be found in the bin/Scripts/Updates folder within the UI folder (either in the release ZIP file, the release file or the VEHub.UI.env_vers folder from step 2 above). Copy all SQL scripts from these folders into the release scripts folder. If there are no update scripts folders/update scripts you can skip the rest of this step 13.
    3. Create an empty SQL file called Rollback to V0.0.0.0.sql in the release scripts folder.
    4. Go through each of the update scripts found/copied in step 13.b above ad add SQL to undo the changes to the rollback SQL file. Note If there are multiple changes to the same records/file/stored procedures the Rollback SQL only needs to set these back to their original state (i.e. before the update scripts are applied). The original state of the database records/file/stored procedures can be found on the target database. Note 2 The test and live databases can both be accessed remotely using MS SSMS (Microsoft SQL Server Management Studio).

7.5 Switch Over

These steps are performed on the target (test or live) web server and need to be performed after all steps in the Release section and release prep above have been completed. Note: These will require the target system being taken down and so for a release to the live system clients should be warned of the planned down-time.

1. In IIS manager stop the VEHub.UI.env_versold site.
2. Swap the port numbers used on the VEHub.UI._env_versold_ and VEHub UI._env_vers_ sites. As both sites should be stopped at this point it will allow you to set the same port on both but will issue a warning.
3. In IIS manager stop the VEHub.Service._env_versold_ site.
4. Swap the port numbers used on the VEHub.Service._env_versold_ and VEHub Service.env_vers sites. As both sites should be stopped at this point it will allow you to set the same port on both but will issue a warning.
5. In the windows services app stop the VEHub.Emailer._env_versold_ service. Set the startup type to Disabled (by double-clicking on the service and changing the startup type drop-down)
6. In the windows services app stop the VEHub.Poller._env_versold_ service. Set the startup type to Disabled (by double-clicking on the service and changing the startup type drop-down)
7. Apply all Update scripts (not the Rollback script) from the release scripts folder (created in step 3.b above) to the target database.
8. Start the VEHub.UI._env_vers_ site in IIS manager.
9. Start the VEHub.Service._env_vers_ site in IIS manager.
10. Start the VEHub.Emailer._env_vers_ service.
11. Start the VEHub.Poller._env_vers_ service.

Ready for What's Next, Now™

>>>>>>>>

7.6 Tidy Up

These steps are intended to prevent the build-up of old IIS sites and windows services over tine and should only be performed after we can be sure everything is OK and no Rollback will be required (e.g. after a test release has been passed or a live release has been used by clients with no issues for some time).

1. Delete the old VEHub.Poller._env_versold_ service using:

   `sc.exe create VEHub.Poller.env_versold`

1. Delete the old VEHub.Emailer._env_versold_ service using:

   `sc.exe create VEHub.Emailer.env_versold`

1. Delete the old VEHub.UI._env_versold_ site in IIS manager, making sure to remove the directory it points to.
2. Delete the old VEHub.Service._env_versold_ site in IIS manager, making sure to remove the directory it points to.

Ready for **What's Next, Now**™

8 VEhub Architecture

8.1 Infrastructure

See http://obs-webdev:8036/index.php?title=Vehub_Environments for details such as IP addresses etc

8.2 Web Server

The following is a list of application specific components, a description of what they are and how theyinterface with other components.

Windows Server 2012 R2

The host operating system. Requires zip file support to unzip TCM software releases. Not OBS have limitedpermissions on the box.

For external access see http://obs-webdev:8036/index.php?title=Vehub_Environments - This link includesVPN access details and RDP access details.

8.2.1 VeHub

For a list of OBS Logistics applications and access to them (including VeHub) see http://10.43.6.50:8036/index.php?title=Vehub_Environments - this link includes the address of theapplication and username/passwords etc. The application is installed at C:\inetpub\wwwroot\VeHub.UI

8.2.2 NTFS (NT File System) Security Settings (Local Users & Groups)

The server must be configured to grant access to the application directories to the <MACHINE-NAME>/Users group.

8.2.3 IIS (Internet Information Services) 8.5

The Microsoft web server responsible for managing requests to the application and returning responsesconsisting of web pages.

IIS delegates the work required respond to a request for a

ASP.Net page to the ? ASP.Net WorkerProcess? (refer to APPENDIX A). The Worker Process constitutes the page for IIS to respond back to the user.

IIS must remain active. If it is stopped for any reason the application will no longer be available, all users areredirected to the ?Page Unavailable? page.

Someone needs to be responsible for monitoring the status of IIS and sending appropriate alerts if itis stopped and cannot be automatically restarted.

There is 1 website setup for VeHub and this points to the application directory.

The application is configure to run on port 2080

The application has its own Application Pool which is used to manage connections between the applicationand the CALIDUS VeHub database.

8.2.4 Microsoft .NET Framework 4.5

This component provides the software layer to run ASP.NET applications. More generally it includeseverything needed to run applications developed using the .NET Framework. VeHub currently utilises version4.5 of the .NET Framework SQL Net Client

This SQL component enables client connections to a remote SQL database. To optimize performance theapplication uses a pool of established database connections. The web server acts as the client connecting tothe CALIDUS VeHub SQL database hosted on the Linux server.

Ready for **What's Next, Now**™

VeHub is installed with its own individual SQL client rather than relying on specific versions to be installedon the server that it is put on. This removes associated issues with DLL versions. The managed client doesnot rely on the tnsnames.ora file containing the correct connection definition and as such manages thatitself by defining it in the web.config file of the application. An example of the configuration section is below

8.2.5 Microsoft Data Access Components

This component provides the framework to allow applications to access any data store. The MDAC architecture may be viewed as three layers: a programming interface layer, consisting of ADO.Net

(refer to fig. 1) the database access layer developed by database vendors (SQL*Net), and the databaseitself (SQL).

8.2.6 Event Viewers and Logs

Event Viewer is a component of the Microsoft's Windows operating system that lets administrators andusers view and search the event logs on a local or remote machine. Applications and operating system components can make use of this centralized log service to report eventsthat have taken place, such as a failure to start a component or complete an action. The system defines three log sources, "System", "Application", and "Security". The System and Application log sources are intended for use by the Windows operating system andWindows applications respectively; the Security log source, however, is only directly writable by the LocalSecurity Authority Subsystem Service (lsass.exe) The VeHub application audits various events to the standard event log. [Screenshot needed] The events need to be periodically monitored and any items for investigation passed onto the VeHubSupport team. The events can be manually viewed using the Microsoft Event Viewer, this utility can bestarted from the ?Administrative Tools? menu option within ?Control Panel?. All events are logged as type?Application? with source of "Terminal Container Management" In terms of support call investigation, checking the event viewer for issues is a good place to start.

8.2.7 Application Configuration & Management

The following is a list of items that relate to application configuration and management

8.2.7.1 Web.config

This file is used to maintain application-specific settings. The file is held in the application install folder. Generally any changes to this file will be delivered in the formof step-by-step instructions of changes rather than the file itself. The settings contained in this file include: Application Specific Configuration webpages:Version - Web Pages version number webpages:Enabled - ??? PreserveLoginUrl - ??? ClientValidationEnabled - Client validation should be enabled and working UnobtrusiveJavaScriptEnabled - Use Unobtrusive Ajax on client pages NewCustomerAlertEmailsTo - a comma seperated list of all emails to send alerts to when a newcustomer signs up PdfWebApi - Url to Pdf Web API HtmlGenApi - Url to Html Generated API reCaptchaPublicKey - Public key for reCaptcha validation reCaptchaPrivateKey - Private key for reCaptcha validation Session Framework Target Framework Version - i.e. 4.5. < httpRuntime targetFramework = "4.5" /> SQL Connectivity Connection Strings

8.2.8 ASP.NET Configuration in IIS

This provides a mechanism to configure the application settings. Changes made using this feature will update the web.config file. Selecting one of the various categories (seescreenshot) will allow settings relating to the category to be updated in the web.config file.

8.2.9 Login / Authentication

The application requires a company, username and password for a given user to access it.

8.2.10 Web Browsers

The Admin application is designed to run in a modern web browser such as Firefox, Chrome or Edge. Theapplication requires JavaScript and cookies to be enabled. < sessionState mode = "InProc" customProvider = "DefaultSessionProvider" > < providers > < add name = "DefaultSessionProvider" type = "System.Web.Providers.DefaultSessionStateProvider, System. </ providers > </ sessionState > < connectionStrings > < add name = "Remote" providerName = "System.Data.SqlClient" connectionString = "Password=Password < add name = "Local" providerName = "System.Data.SqlClient" connectionString = "Password=Password123$; </ connectionStrings >

9 VEhub Project Details

About

9.1 Fleet efficiency and safety

Maintaining the correct safety information across your fleet just got simpler and faster for you and your drivers. In the process of doing away with most of the paper, OBS Logistics and our strategic partner TomTom reinvented the process of keeping your fleet safe and in compliance with Driver & Vehicle Standards Agency (DVSA - previously VOSA) requirements. We provide a timely capability for recording accident information, as well as knowing where your key transport resources are.

9.2 Accessible technology

CALIDUS VEhub runs as a straightforward App on Android devices, and specifically for TomTom followers, will deploy on all 827x series devices. This gives you an all-in-one driver-orientated dashboard device that acts as both Sat Nav and driver information centre.

9.3 Easy access

All your driver has to do is log on and, through a few easy steps, they can capture all the safety checking information the law requires before starting their journey. Photo capture and defect reporting make the process more consistent and accurate than paper, not to mention faster as an end to end process.

9.4 Accident records

If the worst happens and a driver is involved in an accident, information is best captured as soon as possible.

VEhub has a full accident logging capability, including accident scene and vehicle damage photo capture.

The format driven recording helps to ensure that the details that are required are fully captured. The accident report can then be sent back to the transport department for processing rather than waiting for the driver to return to base.

9.5 Assets located and logged

Add a low cost barcode scanning App to your VEhub and it becomes a rapid data capture tool. All you have to do is attach barcode labels to your fleet assets and your staff can rapidly scan them with VEhub logging the precise location and time, making tracking fast, accurate and above all easy.

9.6 Reporting

CALIDUS VEhub has a full suite of reports so that you can rapidly assess the condition of the fleet and take proactive action to keep your vehicles operating in optimum condition.

9.7 Extra

The Vehub App communicates with a WCF Service to fetch data. The Vehub App is built using the Cordova framework.

The App is available on the google play store

9.8 Product Folder:

\\spekefs2012\Projects\Product\VEHUB

9.9 1. Development

This contains the original requirements, updates and the source code for the current VeHub

Ready for **What's Next, Now**™

9.10 1. Original Requirements

This contains documents laying out the original requirements for VeHub

9.11 2. Product Updates

This contains documents explaining product updates since the original requirements

9.12 3. Source

This contains source code backups and a place to put temporary code to be shared

9.13 Quality Control

This contains QA documents that have been put together when testing VeHub

9.14 Quality Control - 18072017

This contains QA documents that have been put together when testing VeHub

9.15 Web Tools

This contains tools obtained from the web

9.16 2. Management

Project Management related documentation such as handover documentation

9.17 3. Projects

Project Management related documentation such as project requirements

9.18 5. Releasing

Documentation on how to release certain parts of the project

9.19 6. Documentation

This contains any number of documents related to VeHub

9.20 APK - Android

This contains the Android Builds of VeHub

9.21 APK - IOS

This contains the IOS Builds of VeHub

9.22 Source Code location

All of the source code is housed in our web development server's Azure source control and development management system. The home page for the Vision is located here:

Visual Studio 2017(Warning: Cordova project will fail to load in Visual Studios 2019) you can view the solution locally by navigating here and cloning: se your Aptean credentials when prompted.

There are two repos but this is the one that should be used, ?VeHub?

Ready for **What's Next, Now**™

9.23 Git

9.24 Source/Version Control

Track progress of files by creating a series of snapshots and branches which you can move back and forth between.

Why

1. Distribute file changes overtime

2. Prevent data loss/damage

3. Manage complex project structures Created by guy who made Linux to manage Linux, The most popular source control software in the world with lots of documentation and support, and lots of integration with other applications (SourceTree, Heroku, GitHub).

Mercurial - a popular alternative to Git

9.25 Branches

Allow you to create a series of commits on a separate timeline without the need to taint your master branch. Be sure to do an initial commit before creating branches or you risk the master being set as one of the branches.

Pull/Fetch Requests git pull is a Git command used to update the local version of a repository from a remote repository

9.26 Push Requests

The git push command is used to upload local repository content to a remote repository. Pushing is how you transfer commits from your local repository to a remote repository

9.27 Local vs Remote Repositories

Git will allow you to have both Local and Remote versions of your repositories, Local repositories are stored on your local computer, changes made to a local repository won?t be reflected in the associated Remote repository until a push request is sent.

Remote Repositories are available to everyone with access and changes to the remote repository can be seen at any time, and with a pull/fetch request a local repository can be created/updated with the latest changes.

9.28 The Source Code

The source code is a .NET Framework 4.5 Visual Studio solution.¬ This started out as a paid, externally developed application which was brought in house and developed further.

The solution contains 2 websites, 1 Cordova project and 5 C# projects

9.29 Projects

9.30 VEHub.Admin

9.31 VEHub.Core

The Core Project is used to contain all of the core functionality for the application, from business logic, to helper classes, this allows for greater separation and can be used to create a cross platform application.

9.32 Calidus VEHub Device (DEV)

The android device project, a Cordova application using JavaScript to handle the page layout and and logic.

Ready for **What's Next, Now**™

〉〉〉〉〉〉

9.33 Emailer

The Vehub emailer is a windows services that checks the Email table in the database and sends emails out using the details created by the system Alerts.cs.

The emails are sent via SMTP.

If an email should fail to send then email is flagged as unsent and the error is logged.

The Emailer Is Running as a service and is located here on the server

The Settings for the Emailer service are stored in the Emailer.exe.config file:

```
<Emailer.Properties.Settings> <!--Emailer Service Error Log location--> <setting name="LogPath" serializeAs="String">
<value>C:\Services\Emailer\log\</value> </setting> <!--SMTP From Name--> <setting name="SMTP_FromName"
serializeAs="String"> <value>VEHub</value> </setting> <!--SMTP From Address--> <setting
name="SMTP_FromAddress" serializeAs="String"> <value>vehub@obs-logistics.com</value> </setting> <!--SMTP
Server--> <setting name="SMTP_Server" serializeAs="String"> <value>10.43.0.20</value> </setting> <!--SMTP PORT
Number--> <setting name="SMTP_Port" serializeAs="String"> <value>25</value> </setting> <!--SMTP Requires
Authentication Flag--> <setting name="SMTP_RequiresAuth" serializeAs="String"> <value>True</value> </setting>
<!--SMTP UserName--> <setting name="SMTP_User" serializeAs="String"> <value>vehub</value> </setting> <!--SMTP
PASSWORD--> <setting name="SMTP_Pass" serializeAs="String"> <value>{smtpPassword}</value> </setting> <!--SSL
Required for SMTP--> <setting name="SMTP_Ssl" serializeAs="String"> <value>False</value> </setting> <!--How Often
it checks for email--> <setting name="CheckFrequency" serializeAs="String"> <value>60000</value> </setting>
<!--Connection String--> <setting name="DB_String" serializeAs="String"> <value>{connectionString}</value> </setting>
<setting name="UseChilkat" serializeAs="String"> <value>False</value> </setting> <!--PDF Attachments location -->
<setting name="PdfAttachments" serializeAs="String"> <value>C:\inetpub\wwwroot\VEHub.Service\pdf</value>
</setting> </Emailer.Properties.Settings>
```

The Emailer Services uses the configuration to set up and send emails.

9.34 Emailer Service Installation

The services can be installed using a batch file, e.g.: @echo off

9.35 Location of installutil.exe

cd C:\Windows\?\Framework\v4.0.30319

9.36 Location of Service

sc create Emailer binpath="C:\Services\New Emailer{ServiceName}.exe" pause

if ERRORLEVEL 1 goto error

exit pause

error

echo There was a problem pause and uninstalled using a Powershell script: @echo off cd
C:\Windows\Microsoft.NET\Framework\v4.0.30319

sc stop {ServiceName} sc delete {ServiceName}

pause

if ERRORLEVEL 1 goto error exit pause

error

echo There was a problem pause

Ready for **What's Next, Now**™

9.37 Poller

The Poller service works similar to the emailer service, it checks the system for vehicles that are overdue a vehicle check then creates an email alert to be sent out.

The OverDue Poller is Running as a service and is located here on the server:

9.38 TomTomLink

Used to connect and interact with the TomTom elements of the application.

9.39 VEHub.Service

The VeHub Web Services that the device applications use to Fetch/Create/Update/Delete data through

9.40 VEHub.UI

This is the PC site for handling of data/account information that the devices are used to record and update.

9.41 Vehub PDF Printer Service

Vehub uses the PDF Printer service to generate pdf reports for gift aid.

The PDF Printer service takes in html generated by vehub and returns a pdf document.

Vehub points to the pdf service via an API endpoint.

This endpoint is stored in the config file of vehub.ui and vehub.service.

<add key="PdfWebApi" value="http://localhost:3001/%22/> The Vehub Web Portal has an endpoint that uses the razor engine to generate strongly typed html pages. The html is generated and passed on to the PdfWebApi to generate the pdf.

9.42 Dev Environment

The system is available for use at OBS

9.43 Server Details

Server name: SHARED-WEB-R OS: Windows 2012 Standard IP: 10.43.6.50 RDC User: SHARED-WEB-R <Own User> RDC Password: <Own Password> IIS Version 8.0.9200.16384

9.44 Database

Server name: SHARED-SQLSVR-R OS: Windows 2012 Standard IP: 10.43.6.51 RDC User: SHARED-SQLSVR-R<Own User> RDC Password: <Own Password> SQL server: Microsoft SQL Server 2012 - 11.0.5058.0 (X64) SA: sa / sa_epod13- Epod user: epod / ePOD1234

9.45 Live and Test Environments

9.46 Live

9.47 Server Details

Server name: SHARED-WEB-R OS: Windows 2012 Standard IP: 172.183.45.50 RDC User: SHARED-WEB-R <Own User> RDC Password: <Own Password> IIS Version 8.0.9200.16384

Ready for **What's Next, Now**™

9.48 Database

Server name: SHARED-SQLSVR-R OS: Windows 2012 Standard IP: 172.184.45.200 RDC User: SHARED-SQLSVR-R<Own User> RDC Password: <Own Password> SQL server: Microsoft SQL Server 2012 - 11.0.5058.0 (X64) SA: sa / caL1duS# Epod user: epod / ePOD1234

9.49 Test

9.50 Server Details

Server name: SHARED-WEB-R OS: Windows 2012 Standard IP: 172.183.45.55 RDC User: SHARED-WEB-R <Own User> RDC Password: <Own Password> IIS Version 8.0.9200.16384

9.51 Database

Server name: SHARED-SQLSVR-R OS: Windows 2012 Standard IP: 172.184.45.205 RDC User: SHARED-SQLSVR-R<Own User> RDC Password: <Own Password> SQL server: Microsoft SQL Server 2012 - 11.0.5058.0 (X64) SA: sa / sa_epod13- Epod user: epod / ePOD1234

9.52 Test VeHub SMTP service is running using Telnet

Open a command prompt as an administrator and enter the following line by line: telnet 10.43.0.20 25 helo ehlo MAIL RCPT DATA This is a email sent via Telnet .

QUIT This assumes the SMTP service is still 10.43.0.20 on port 25, obviously this may change at some point. The recipient email address should be changed to one that the tester has access to.

9.53 Google Play Store Deployment

9.54 Update Version Numbers

Using Visual Studio Open the config.xml file in the solution. In the Common Tab Update the version number:

For Android builds, update the version number in the android tab:

9.55 Do a Release Build

Deploy the app by selecting Deploy in the context menu. Visual Studio will build an APK for the app and output this to the bin directory.

9.56 Upload to Play Store

The Google account that the app is registered to is: ? Google Account: ? Password: THaAw Used for google services, like analytics, playstore , captcha etc? anything google that?s related to dev work:? Google Account: ? Password: _Liverp Play Store upload link: ? ? T ? ?

9.57 Checking for Errors and Faults

System Log Table

SELECT * FROM SystemLog

The Vehub emailer is a windows services that checks the Email table in the database and sends emails out using the details created by the system Alerts.cs.

The emails are sent via SMTP. If an email should fail to send then email is flagged as unsent and the error is logged.

9.58 Vehub Custom Vehicle Checks Flag

Ready for **What's Next, Now**™

9.59 Vehub Custom Vehicle Checks Flag

VeHub Customers that had an account prior to the new registration will need to have a flag set in order to activate the custom fields functionality.

The flag enables custom fields for the device(app) and the web-portal.

When the flag is set users will be able set custom-fields and drivers will be able to use these fields on the device.

To get started you will need the Customer Code to be able to run the query.

With the customer code, Query the customers table to get the customer Id: SELECT * FROM Customer WHERE CustomerCode ='Demo'

Once you have the Customer Id , use it to run the stored procedure uspCustomerLegacySettingSave :

To Test this, login to the web portal using the account and check the settings tab. What used to be Vehicle Check Items is now Custom Vehicle Check Items :

With the flag now set, the customer can create custom vehicle check items.

9.60 VeHub Inaccessible Vehicle Checks Issue

When trying to access vehicle checks the following error message is displayed: SYSTEM ERROR

Internal Server Error [500].

It is believed that this is caused when Custom Vehicle Check Templates are introduced after a customer upgrade: these don?t have Vehicle Check Items. Therefore if old "vehicle check" indicators are set, for instance by the use of an older version of the App, that doesn?t support the newer checks, after the upgrade causing the checking of "vehicle check" items which can no longer be accessed.

The proposed solution is to run a script that will set all the old indicators back to 0 for the new "vehicle check templates". However if there is still a customer device operating the old "vehicle checks" App then the problem will occur again.

1. "Vehicle check" reset script

9.61 IMPORTANT:

? This process needs to be tested before being implemented on a live customer? Replace the Template:CustomerId placeholder in the script

UPDATE VehicleCheck SET
CkLeaks = 0, CkBattery = 0, CkTyres = 0, CkSpray = 0, CkSteering = 0, CkLoad = 0,
CkMirrors = 0, CkLights = 0, CkReflector = 0, CkIndicators = 0, CkWipers = 0,
CkWashers = 0, CkHorn = 0, CkExhaust = 0, CkBrakeLines = 0, CkCoupling = 0,
CkElectrics = 0, CkBrakesABS = 0, CkBodywork = 0, CkMarkers = 0, CkGlass = 0
WHERE TemplateId NOT IN
(SELECT VehicleCheckTemplateId FROM dbo.VehicleCheckItem WITH (NOLOCK)
WHERE CustomerId = Template:CustomerId GROUP BY VehicleCheckTemplateId)

Ready for **What's Next, Now**™

10 VEhub Support

This page identifies common support tasks and resolutions

10.1 Requirements

- SQL Server Management Studio
- Access to SharedWeb database

10.2 Updating Admin and PDA users

- Open the right database on SharedWeb e.g. VEHUB_NEW
- Find the customer

```
SELECT TOP 1000 [Id]
      ,[CustomerCode]
      ,[CustomerName]
      ,[Address1]
      ,[Address2]
      ,[Address3]
      ,[Country]
      ,[Postcode]
      ,[TotalBackOfficeLicenses]
      ,[TotalPDALicenses]
      ,[ModifiedBy]
      ,[ModifiedDate]
      ,[CustomerImageSrc]
  FROM [VEHUB_NEW].[dbo].[Customer]
  where customerName like '%partialCustName%'
```

- Note the customer ID.
- Edit Customer record and change these values:
    - ♦ [TotalBackOfficeLicenses]
    - ♦ [TotalPDALicenses]

```
UPDATE [VEHUB_NEW].[dbo].[Customer]
SET [TotalBackOfficeLicenses] = x
, [TotalPDALicenses] = y
WHERE CustomerId = z
```

Ready for **What's Next, Now**™