

Engine Angel Admin Layer

ENGINE ANGEL No Account? [Request one](#) [Sign In](#)

WHAT IS ENGINE ANGEL? HOW DOES IT WORK? BENEFITS WHAT IS REQUIRED? HOW DO I GET STARTED? CONTACT US



Diesel, Gasoline, CNG, Bio-fuels



4-stroke AND 2-stroke



Works for all types of reciprocating engines

ENGINE ANGEL No Account? [Request one](#) [Sign In](#)

WHAT IS ENGINE ANGEL? HOW DOES IT WORK? BENEFITS WHAT IS REQUIRED? HOW DO I GET STARTED? CONTACT US



Hauling: People, Products, Resources



On Highway, Off-road, Water, On Rails, In the Air, Stationary



For most vehicles in the transportation industry

ENGINE ANGEL No Account? [Request one](#) [Sign In](#)

WHAT IS ENGINE ANGEL? HOW DOES IT WORK? BENEFITS WHAT IS REQUIRED? HOW DO I GET STARTED? CONTACT US



ENGINE ANGEL IN ACTION

- Fleet Management
- Large Engine Repair & Rebuild
- Used Vehicle Dealer



KNOWLEDGEBASE

- Background Information
- Case Studies
- Science & Engineering



FREQUENTLY ASKED QUESTIONS

- What is Engine Angel?
- How much does it cost?
- How do I sign up?

Go from data about the past to prediction of what is likely next and recommendations to make that future better!

This page intentionally blank.

Engine Angel Documentation

Volume I. Admin Layer

2019-05-07

Contents

Volume I. Admin Layer	3
Introduction	6
Engine Angel Standards	8
Application Environment	8
Authorization	9
Application and Database table design.....	9
Locale-specific data presentation	11
Time Intervals.....	12
Multiple Language Support.....	13
Engine Angel Data Layers (Flows)	15
Admin Layer	15
Vehicles - Background.....	16
Vehicle Manufacturers.....	16
Vehicle Models.....	16
Vehicle Types	17
Vehicle Vocations.....	17
Vehicle Chassis	18
Vehicle Aerodynamic Features	19
Trailer Data	19
Trailer Types.....	20
Trailer Chassis (use Vehicle Chassis screen).....	20
Trailer Aerodynamic Features.....	21
Engines	21

Engine Manufacturers.....	22
Engine Occupations.....	23
Fuel Types	23
Engine Models Maintenance	24
Engine Block Diagrams and Adjacency tables.....	25
Engine Cylinder Offset Diagram	26
Engine Technologies	26
Engine Flaws.....	28
Expense Items- Background.....	29
Axle Types	29
Tire Brands	29
Tire Lines	30
DOT Tire Codes.....	30
Brake Brands	31
Battery Brands	31
Inventory Expense Groups	31
Classification	32
Engine Configurations	32
Engine Conditions	33
Channel Position	33
Code Lists	34
Security Data	35
All People	36
Fleets.....	36
Account Requests.....	36
Localization Data.....	36
Languages.....	37
TimeZones.....	37

Currencies	38
Countries	40
State/Provinces	40
Postal Codes	40
Global Facilities	40
UoM Rule	40
UoM Data	41
Alerts/Content	42
CMS Editor	42
Email Templates	42
Alert Criterion	43
Alert Definitions	44
Diagnostic Text	45
Work Request (Templates)	46
Work Request – Data ¹	47
Wizards Data – Processes	48
Subscription Types	48
Remote Data	48
Data Connects	48
Translation	49
Translate CodeLists	49

Introduction

The documentation for Engine Angel is provided in 5 volumes of system functions and data organized by the sources of data and information – people, sensors, and analytical processes.

1. **Admin Layer (This document)** The **Admin** Layer for the staff of Predictive Fleet Technologies who are responsible for much data that is independent of any customer Fleets (equipment specifications, conversion standards, regulations, industry standards, Expert System rules, etc.)
2. The Fleet **Cockpit** for managers, professional staff responsible for the policy and management of operational people to achieve the organization's goals (managers, analysts, clerical people)
3. The **Operations** people who implement the actions according to policies and guidelines of management resulting in organizational success.
4. The **Remote** functions are those processes that connect sensors and special purpose accumulators of 'short-term' histories of data and transmit, transform and integrate the data into the system's databases to keep data about asset status and activities accurate and timely.
5. The **Scheduled** functions (data integration and analysis) are processes that monitor asset status data against preset threshold values to alert operations and staff personnel of abnormal situations that might need corrective action. In addition, the batch processes use the historical data and models to predict future statuses and compare against the preset thresholds to provide proactive alerts and, using Expert Systems, suggest corrective actions to avert problems.

The following diagram (flow chart) represents these functions and major flows of data through the system.

Engine Angel Standards

Application Environment

The application is developed in PHP under the Microsoft .Net environment.

The EngineAngel homepage is developed under WordPress.

The database management system is MSSQL.

HighCharts (JavaScript) is used for graphic displays of data.

PC components are developed in C##.

Change Control is managed through Team City software.

The production environments are:

EnginePolygraph.com and **EngineAngel.com**

It is occasionally rebuilt upon request from the QA environments.

The development environment consists of three components:

EP-DEV.GLTaC.com/Current and EA-DEV.GLTaC.com/Current

This environment is used by developers.

EP-DEV.GLTaC.com/Nightly and EA-DEV.GLTaC.com/Nightly

This environment is used by testers. The software is updated every early morning from Current

EP-DEV.GLTaC.com/QA and EA-DEV.GLTaC.com/QA

This environment is used by for final integrated testing. The software is updated periodically from Nightly.

Credit card payments from PFT customers are made using Recurly.

ETL uses SSRS to transform and update SQL tables from Remote inputs (files).

Alerts are batch programs that report through SSRS.

Advanced Analytics are developed in R and results reported via csv files.

Various Daily and Monthly batch jobs provide periodic summaries and data maintenance functions controlled by the MS scheduling software.

Authorization

All authorized UserNames are assigned to either the Admin Layer (Predictive Fleet Technologies employees) or the Cockpit/Operations Layers (Customer employees). Once authorized to access a Layer, the UserName is allowed to view all data, except personal data of other people and data under the Security tab.

Authorization to update data is assigned to a UserName by assigning the UserName the corresponding Role (Authorization) on the 'All People' table in the Admin Layer for employees of PFT, or the Role using the Cockpit > Master Data > People table for the Cockpit and Operations Layers.

One UserName can access only the Admin **or** the Cockpit & Operations layers of EngineAngel, not both.

Application and Database table design

Any session can be 'split' into two sessions by clicking on the white + in the red box on the upper right of each screen. This allows each session to have a thread in either of two tabs of your browser.

Each screen contains a System title frame with 'sign on' option at the far right.

- Below that area is a line that contains the Layer title on the left and a DateTime (in Universal Time Coordinated time zone) and a Language pull-down showing the current User Interface language with the option to select a different user presentation language.
- The third line contains a Subject Tab for each of the provided subject areas appropriate to the user name authorized to the Layer.
 - Most of the tabs have a pull-down list of functions appropriate to the Subject tab.

Database tables have a LastUpdateUserName & LastUpdateDateTime to record last change/create source and time.

Most 'function' tabs have a 'list screen' of objects 'maintained' by the function with a string search function for key data from each row (entry). Some of the list screens have pull-down filters to focus on objects of common groupings.

- The list objects are displayed in multiple panels of user selected numbers of rows per panel. At the bottom of the screen, the number of panels presented are displayed with indication of which panel of the set is currently displayed along with options to scroll through the list of panels.
- The sequence of the rows of objects can be altered by clicking on the column title; a second click will reverse the sequence.

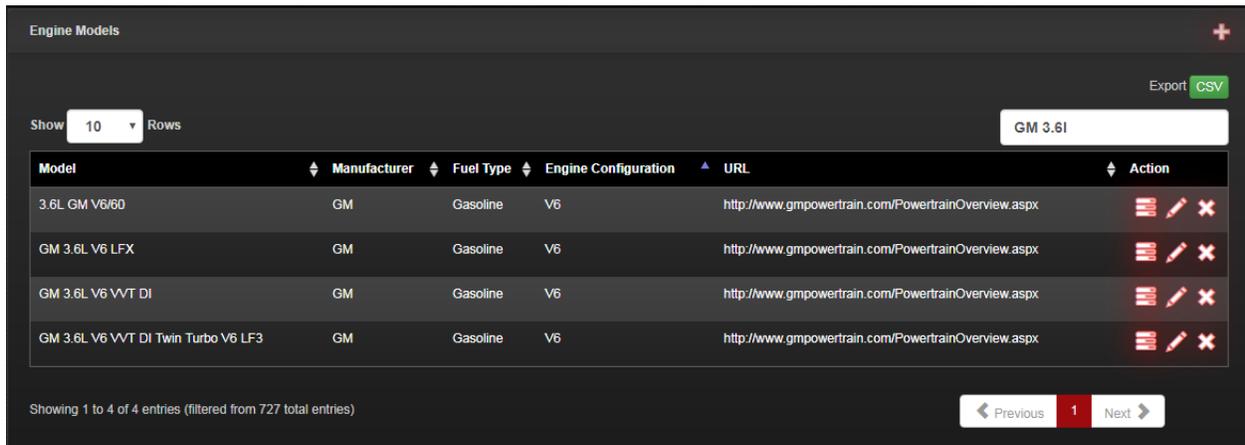
The Export CSV button in the upper right causes the contents of the table of the list screen to be downloaded to your PC as a CSV file for your own reporting.

At the right of a list screen, a pencil might be 'clicked' to get a detail screen for that row showing existing data and, if the user is authorized appropriately, changes can be made and the changes saved.

In many cases, the line will terminate with a bold, reddish 'X'. If it is clicked, you will be prompted if you really want the row deleted. If you are authorized, clicking on 'yes' will cause the row to be deleted if it can be within the structure rules of the database (other things depend on it).

List screens also have bold reddish '+' in the upper right to indicate that you want to add a new entry into the table.

Some List screens end the item line with an 'additional information' icon  that will cause a branch to a screen of information related to the subject of the selected row.



Model	Manufacturer	Fuel Type	Engine Configuration	URL	Action
3.6L GM V6/60	GM	Gasoline	V6	http://www.gmpowertrain.com/PowertrainOverview.aspx	  
GM 3.6L V6 LFX	GM	Gasoline	V6	http://www.gmpowertrain.com/PowertrainOverview.aspx	  
GM 3.6L V6 VVT DI	GM	Gasoline	V6	http://www.gmpowertrain.com/PowertrainOverview.aspx	  
GM 3.6L V6 VVT DI Twin Turbo V6 LF3	GM	Gasoline	V6	http://www.gmpowertrain.com/PowertrainOverview.aspx	  

The above screen shows most of the features of a list screen – the Engine Model table, in this case - that are described in the previous paragraphs.

Help Screens are to be provided for each screen for the User to understand the purpose and general features of the data and business process involved.

Locale-specific data presentation

Event **DateTimes** are to be displayed in Fleet TimeZone; system DateTime is in UTC (as shown on the screens near the top right. Format is yyyy-mm-dd hh:mm:sss (I.e., 2017-01-25 22:07.843). In the csv files, the Date and Time are separated by an ' _ '.

Date input fields use a Calendar pull-down in a popup window for ease and consistency of input.

Units of Measurement are to be metric in the Admin layer and as specified in the Fleet configuration data of Cockpit > Manage Fleet(s) > IsMetric (Y/N). Specific data elements can be defined in the Admin > Localization > UoM Data Rules. Conversions between various UoM are to be calculated using standard routines (SP) from data stored in the UoM Data table.

Distance calculations from GPS coordinated are to use the **Haversine method**.

GPS coordinates are to be stored in the DD (Decimal Degrees) format

Longitude values West of the Greenwich median are stored as negative numbers and positive for East of the meridian. Values must be between -180 and +180 so that if less than -180, add 360; if over 180, subtract 360.

Latitude values are between 0 (equator) and 90 at the North pole;
-90 is the South pole. If a latitude is >90, subtract 90; ; if less than -90, add 90.

Altitude is to be in meters (m).

To convert from DD to DMS (degrees, minutes, seconds):

Function Convert_Degree(Decimal_Deg) As Variant With Application

```
'Set degree to Integer of Argument Passed Degrees = Int(Decimal_Deg)
'Set minutes to 60 times the number to the right of the decimal for the variable Decimal_Deg
Minutes = (Decimal_Deg - Degrees) * 60
'Set seconds to 60 times the number to the right of the 'decimal for the variable Minute Seconds
= Format(((Minutes - Int(Minutes)) * 60), "0")
'Returns the Result of degree conversion (for example, 10.46 = 10° 27 ' 36")
Convert_Degree = " " & Degrees & "° " & Int(Minutes) & "' " _ & Seconds + Chr(34)
End With
End Function
```

<http://support.microsoft.com/kb/213449>

To convert from DMS to DD:

Function Convert_Decimal(Degree_Deg As String) As Double

```
' Declare the variables to be double precision floating-point. Dim degrees As Double Dim
minutes As Double Dim seconds As Double
```

```

' Set degree to value before "" of Argument Passed. degrees = Val(Left(Degree_Deg, InStr(1,
Degree_Deg, "") - 1))

' Set minutes to the value between the "" and the "" of the text string for the variable
Degree_Deg divided by 60. The Val function converts the text string to a number.
minutes = Val(Mid(Degree_Deg, InStr(1, Degree_Deg, "") + 2, _ InStr(1, Degree_Deg, "") -
InStr(1, Degree_Deg, _ "") - 2)) / 60

' Set seconds to the number to the right of "" that is converted to a value and then divided by
3600.
seconds = Val(Mid(Degree_Deg, InStr(1, Degree_Deg, "") + _ 2, Len(Degree_Deg) - InStr(1,
Degree_Deg, "") - 2)) _ / 3600

Convert_Decimal = degrees + minutes + seconds

End Function

```

There are three formats on the sidewalls of tires (**Tire size**):

wwldd.d → ww is the width in inches. Convert to metric (cm) for database.

l=internal construction=[R,D,B], usually R=Radial

dd.d = diameter of wheel, normally to .5". Convert to cm for database.

Assume aspect ratio = .9 so section height = .9*ww

www/aaldd.d → www is width in mm. Convert to cm by dividing by 10;

aa = aspect ratio → section height = (aa/100)*width;

dd.d and 'l' are like above.

www/aalddd → www is width in mm. Convert to cm by dividing by 10. Section height =

(aa/100)*width;

ddd = diameter of wheel in mm, ,convert to cm for DB by divide by 10.

Overall diameter = wheel diameter + 2*(Section height)

7.50R16, 245/75R16, 190/65R390 are examples of each

Time Intervals

The human display for time intervals is of format: dd.hh:mm:ss (dd=days, hh = hours, mm = minutes, ss = seconds).

Time interval values are stored in db as seconds (s)

To go from database seconds (dbs) to human display,

dd = Floor(dbs/86400)

```
hh = Floor((dbs-dd*86400)/3600)
mm = Floor((dbs-dd*86400-hh*3600)/60)
ss = Floor(dbs-dd*86400-hh*3600-mm*60)
s = dd*86400+hh*3600+mm*60+ss
```

There are a number of components required for supporting multiple languages in Engine Angel.

Multiple Language Support

The UTF-8 codepage is required in the various text fields to support the various character sets, and ideograms of the various languages of the world.

Language codes are the 2-character, lower case ISO codes.

Since the Admin Layer is only for PTF access, it is designed for users with English literals and messages. However, data is to be visible in any selected language. Adds, changes, and deletes of codes can only be made in English since that is the language code that will update relationships between codes. Translations are managed by separate, usually batch, functions.

A language must be specified for the fleet which will be the language used for all batch reports.

The user may identify a different language for display of data at the time of sign-on (with ability to change during the session). This will enable the system to display standard descriptions, field titles, and error messages in the language requested. Comments made by other members of the fleet should be made and will be displayed in the fleet language.

Field titles and literal texts provided by the GUI are maintained in a ML-Contents table in the various supported languages.

The CodeLists table identifies codes in their relationships in a set of tables maintained in the Admin layer under the classification menu. The translations of those codes, even into English, can be provided with the batch functions in the translations menu of the admin layer. They are stored in ML- extension tables. The CodeLists and Codes can only be added, deleted or changed only in English when using the online screens; non_English Codes and Descriptions can be viewed in the Admin layer by changing the language (Locale) selection.

A number of tables contain language specific descriptions of groups and objects. These tables have ML-extensions for the corresponding translations. The maintenance of the translations are provided by requested downloads of the English content with the corresponding local language content; these tables are sent to translators, updated, and returned for upload into the ML- extension tables.

Field edit messages in maintenance programs are extracted from the programs and stored in the 'messages' table. ML-extensions provide the local language equivalent after translation.

Texts associated with Alerts, Work Orders, and other batch generated reports for users are maintained in the admin layer under the Alerts/Contents menu.

The following pages in this document detail specific features of the MultiLingual capabilities.

ADMIN => Localization => Languages => (below list screen)

EngineAngel > Translation Foundation > Enabling Engine Angel Translation.pptx

EngineAngel > Translation Foundation > ML Table Maintenance.pptx

EngineAngel > Translation Foundation > MultiLingual Application Messages.pptx

EngineAngel > Translation Foundation > CodeList Tasks.pptx

EngineAngel > Translation Foundation > Translation Tools for Translators.pptx

Engine Angel Data Layers (Flows)

The Engine Angel Application functions can be grouped by Interactive Functions, Remote Functions (such as data collection and integration from sensors and special purpose computers) and scheduled (batch) functions to analyze data for status changes and predictions to generate Alerts to people to consider 'corrective' actions.

Admin Layer

Since so many data elements about Fleet assets and parts and supplies are common around the world, Engine Angel provides much data that is available in the manufacturer's catalogues about the models and brands that Fleets buy. And the maintenance requirements are also independent of Fleet, but depend on activities that are recorded by the Fleet personnel and sensors. This section describes the data that are provided by Predictive Fleet Technologies to make data maintenance easier. It is important for you to keep informing us of new products that are important to your Fleet so we can continue to provide you with data from outside of your organization that is relevant to your fleet.

In addition, PFT provides geographic information such as Country, State/Province, Postal codes, TimeZone data with Daylight Savings Rules, and Units of Measure with conversions.

The Admin Layer also provides a library of Alert messages, Asset Maintenance forms, and standard codes via pull-down windows to ease the data 'on-boarding' and maintenance processes.

Vehicles - Background

Vehicles in the Admin Layer identify self-propelled units that typically carry or pull cargo and/or passengers that are available for Fleets to buy or lease and operate. At this level, the data is abstract, representing manufactures' model features and intended uses. The data is available for any Engine Angel customer to select from.

Fleets are provided Vehicles for on-road and off-road vocations. Marine Vessels are also Vehicles, but have not been loaded to the database tables as of the time of this document's publication. Stationary engines are installed in Facilities instead of Vehicles; a number of these engines are provided in our Engine tables, but we are far from complete and expect our customers will let us know the manufacturers and models relevant to current business needs.

Vehicles - Purpose

The purpose of the Vehicle data is to reduce cost and time for data loads by customers and to provide consistency in the coding to support on-going customer evolution to increased system functionality, and comparison of Vehicles (performance, reliability, etc.) between models.

Vehicles - Security

Any user in the Admin Layer (Predictive Fleet Technologies employees only) who has been granted Vehicle Authorization can Add, Change, and sometimes Delete records and data content. All Admin users may view any of the data on these screens.

Much of the data is available via 'pull-down' lists in the Cockpit and Operations layers. Data content is made available where it supports implemented system function.

Vehicle Manufacturers

Vehicle Manufactures are listed as producers of models of vehicles for sale. The name of the organization, an acronym and an URL are available data elements. The acronym and URL are optional.

The primary purpose of the Manufacture identification is to provide subsets of models to users in the cockpit layer for easier and more accurate selection.

We often separate the Manufacturer into two records if a division of the manufacture produces exclusively models of a sub-category, such as Truck, Marine, etc.

The URLs are owned by other organizations so may be obsoleted at any time. The PFT staff updates these when it is observed that one has been changed. Mergers present the biggest challenge here because we do not time-stamp the duration of the trade names nor show evolution of the company structure over time.

A Manufacturer cannot be deleted unless it has no Models attached to it.

Vehicle Models

Vehicle Models are listed 'under' manufactures of models of vehicles for sale. The name of the manufacturer, model name, an acronym and an URL are available data elements. The acronym and URL are optional.

The primary purpose of the Model information is to provide data about the vehicles produced under the model name to users in the cockpit layer for easier and more accurate selection. The data that is input to the system comes from manufacturer spec sheets or other public sources.

The Model record has a 'Default Chassis' configuration and a 'Default Drag Coefficient' that are presented to the authorized Fleet Master Data person to be confirmed or edited as appropriate.

A Model does not have a beginning or end date in the system. This means that a Model may be in production for some time period, but the produced model may be in operation for any number of years later.

Models are identified by common trade name used in the literature, often by the manufacturer's advertisements.

A new model can be 'copied' as a New Entry (Model) by checking the 'Save as New Entry' button and changing the data that is different.

A Model cannot be deleted unless it has no Vehicles attached to it.

A Model may have a default Chassis attached and a default Aerodynamic drag coefficient.

Vehicle Types

Vehicle Types are illustrated by an image that represents the shape and to a large extent, purpose of the vehicle.

The primary purpose of the Vehicle Type is to provide a technique for the Fleet personnel to assign the Vehicles they deploy into the fleet operation to a class so that the vehicles can be compared across a Type in the Fleet. The Type also provides information about additional data element requirements and options.

A Vehicle Type does not have a beginning or end date in the system.

Vehicle Types are identified by the stylized image but the shapes will vary over time and manufacturer.

A new Vehicle Type can be 'copied' as a New Entry (Type) by checking the 'Save as New Entry' button and changing the data and image that is different.

A Vehicle Type cannot be deleted unless it has no Vehicle attached to it.

Vehicle Vocations

Vehicle Vocations describe the industry that the Vehicle is expected to perform in. It also has an associated Application code to identify the environment that the Vehicle will operate in, e.g., urban, off-road, line haul, regional, etc.

The primary purpose of the Vehicle Vocation is to provide a technique for the Fleet personnel to assign the Vehicles they deploy into the fleet operation to a class so that the vehicles can be compared across Vocations in the Fleet. E.G., Vehicles in a single Vocation are compared and ranked among themselves.

A Vehicle Vocation does not have a beginning or end date in the system; however, the Vocation of a Vehicle usually does go through multiple Vocations during its useful life; but Vehicle can be assigned only one Vocation at a time.

A new Vehicle Vocation can be 'copied' as a New Entry (Model) by checking the 'Save as New Entry' button and changing the description and Application that are different.

A Vehicle Vocation cannot be deleted unless it has no Vehicle attached to it.

Vehicle Chassis

The Chassis record describes the basic chassis features of a Vehicle. This screen defines Chassis for both Vehicles and Trailers. It specifies the following features that are not likely to change over the life of the Vehicle.

The Chassis code has meaning to aid the user in selecting the appropriate code from a pull-down list: The first position is a 'L' if the Vehicle is a left-side drive (driver sits on the left side of the vehicle while driving) or 'R' for right-side drive; the second part of the code indicates the type of axle of the first axle of the vehicle (Axle Types are ['S' = steering; 'D' = drive; 'T' = trailer]) and followed by the number of consecutive axles that are of the same type. If any axle is a lift axle, it is coded by a preceding 'L'. So 'L-S1-LT1-D2' refers to a Vehicle that is left-hand drive, one steering axle in front, followed by a lift trailer axle, followed by 2 drive axles.

The Chassis record also specifies the number of batteries on the Vehicle/Trailer.

The Chassis code prompts the Fleet user to configure the wheel rims and tires, brakes on each axle and the battery set(s). The configured data defines the Vehicle/Trailer 'map' to prompt shop personnel for inputs during inspections and maintenance operations.

The number of Rims can be changed. But if that is done, a window will pop-up saying 'the number of tires changes when the number of rims change'. You must respond 'OK' to that message and then requested change must be Saved.

A new Vehicle/Trailer can be assigned a Chassis from the pull-down list, take the default from the prompt assigned to the Vehicle model or copied from an existing, configured Vehicle. The 'Distance on tire' will be inherited from the Vehicle/Trailer Master record.

A Chassis cannot be deleted unless it has no Vehicles attached to it.

Vehicle Aerodynamic Features

'Aero components' are illustrated by an image that shows an example of the aerodynamic feature (OEM or after-market) to affect aerodynamic drag.

The Aero components are made available for the Fleet personnel to add and remove such features to Vehicles and Trailers of the Fleet, adjusting the unit's coefficient drag to estimate better the fuel economy to be expected by the Vehicle.

An Aero component does not have a beginning or end date in the system. But while the feature is attached to the Vehicle, it is active for its affect on the aerodynamics and drive fuel economy.

The adjustment to the Vehicle coefficient of drag from wind tunnel experiments is provided.

A new Aero component can be 'copied' as a New Entry (Type) by checking the 'Save as New Entry' button and changing the data and image that is different.

An Aero component cannot be deleted unless it has no Vehicle attached to it.

Trailer Data

Trailers data in the Admin Layer identify non self-propelled units that typically carry cargo and/or passengers that are available for Fleets to buy or lease and operate. At this level, the data is abstract, representing Trailer Types and intended uses. The data is available for any Engine Angel customer to select from.

The Trailer industry is much more specialized around the world than Vehicles so that we do not provide Manufacturer nor Model data to the Fleets, but they provide that data in the Fleet Trailer record in the Cockpit Layer.

Trailers may have Engines attached at the Fleet level, e.g., reefers or cement mixers

Trailer Data- Purpose

The purpose of the Trailer data is to reduce cost and time for data loads of customers and to provide consistency in the coding to support on-going customer evolution to increased system functionality and comparison of Trailers (performance, reliability, etc.) between Types.

Trailer Data - Security

Any user in the Admin Layer (Predictive Fleet Technologies employees only) who has been granted Trailer Authorization can Add, Change, and sometimes Delete records and data content. All Admin users may view any of the data on these screens.

Much of the data is available via 'pull-down' lists in the Cockpit and Operations layers. The limited data content is made available where it supports implemented system function.

Trailer Types

Trailer Types are illustrated by an image that represents the shape and to a large extent, purpose of the Trailer.

The primary purpose of the Trailer Type is to provide a technique for the Fleet personnel to assign the Trailers they deploy into the fleet operation to a class so that the Trailers can be compared across a Type in the Fleet. The Type also provides information about additional data element requirements and options.

Trailer Type data can ONLY be changed and added in English. Translations of the description and code must be done with batch spreadsheets. Chassis and Drag coefficient cannot be different by language.

A Trailer Type does not have a beginning or end date in the system.

Trailer Types are identified by the stylized image but the shapes will vary over time and manufacturer.

A new Trailer Type can be 'copied' as a New Entry (Type) by checking the 'Save as New Entry' button and changing the data and image that is different.

A Trailer Type cannot be deleted unless it has no Trailer attached to it.

Trailer Chassis (use Vehicle Chassis screen)

The Chassis record describes the basic chassis features of a Trailer. This screen defines Chassis for both Vehicles and Trailers. It specifies the following features that are not likely to change over the life of the Trailer.

The Chassis code has meaning to aid the user in selecting the appropriate code from a pull-down list: The first position is a 'L' if the Trailer is pulled by a left-side drive (driver sits on the left side of the vehicle while driving) or 'R' for right-side drive; the second part of the code indicates the type of axle of the first axle of the vehicle (Axle Types are ['S' = steering; 'D' = drive; 'T' = trailer]) and followed by the number of consecutive axles that are of the same type. If any axle is a lift axle, it is coded by a preceding 'L'. So 'L-LT1-T2' refers to a Trailer that follows a left-hand drive Vehicle, first axle is a lift trailer axle, followed by 2 Trailer axles.

The Chassis record also specifies the number of batteries on the Trailer, usually if the Trailer has an Engine.

The Chassis code prompts the Fleet user to configure the wheel rims and tires, brakes on each axle and the battery(ies). The configured data defines the Trailer 'map' to prompt shop personnel for inputs during inspections and maintenance operations.

A new Trailer can be assigned a Chassis from the pull-down list or copied from an existing, configured Trailer.

A Chassis cannot be deleted unless it has no Trailers attached to it.

Trailer Aerodynamic Features

'Aero components' are illustrated by an image that shows an example of the aerodynamic feature (OEM or after-market) to affect aerodynamic drag.

The Aero components are made available for the Fleet personnel to add and remove such features to Trailers of the Fleet, adjusting the unit's coefficient drag to estimate better the fuel economy to be expected by the Vehicle pulling it.

An Aero component does not have a beginning or end date in the system. But while the feature is attached to the Vehicle, it is active for its affect on the aerodynamics and drive fuel economy.

The adjustment to the Trailer coefficient of drag from wind tunnel experiments is provided.

A new Aero component can be 'copied' as a New Entry (Type) by checking the 'Save as New Entry' button and changing the data and image that is different.

An Aero component cannot be deleted unless it has no Vehicle attached to it.

Engines

Engines in the Admin Layer identify manufacturers and their models for use by Engine Angel customers. This engine data is also available for use by the Engine Polygraph application to aid in interpretation of the FirstLook® Signatures.

There are other Engine/Engine Polygraph related tables under the Classifications menu option: Configurations, Conditions, and Channel Positions.

This data is collected and provided by Predictive Fleet Technologies to customers as an aid in the efficiency, accuracy and consistency of engine identification & specification.

Engines - Purpose

The purpose of the Engine data is to reduce cost and time for data loads of customers and to provide consistency in the coding to support on-going customer evolution to increased system functionality and comparison of Engines (performance, reliability, etc.) between models. Some of the data is used by Engine Polygraph for the analysis of the EP Signatures and the 'Abnormal Observation' Diagnostics.

Engines - Security

Any user in the Admin Layer (Predictive Fleet Technologies employees only) who has been granted Engine Authorization can Add, Change, and sometimes Delete records and data content. All Admin users may view any of the data on these screens.

Much of the data is available via 'pull-down' lists in the Cockpit and Operations layers. Data content is made available where it supports implemented system function.

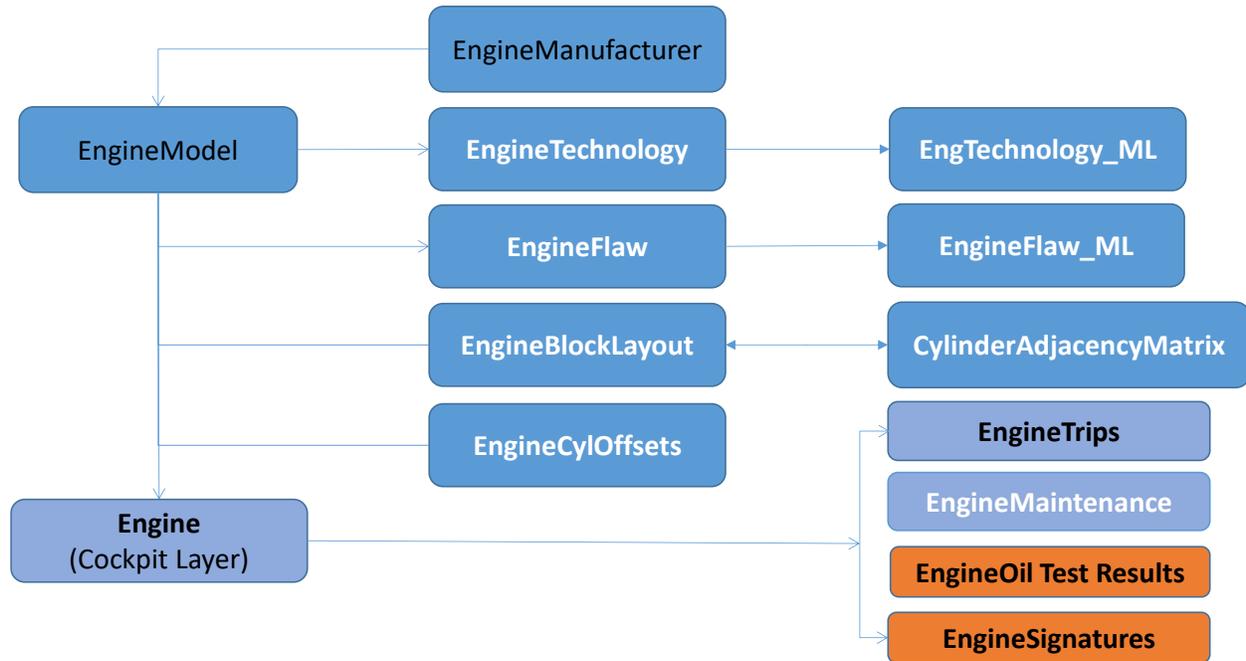


Figure xx. Relationships of a number of components of the Engine Model data in Admin and key data in the Cockpit layer.

Engine Manufacturers

Engine Manufacturers are listed as producers of models of Engines for sale. The name of the organization, an acronym and an URL are available data elements. The acronym and URL are optional.

The primary purpose of the Manufacture identification is to provide subsets of models to users in the cockpit layer for easier and more accurate selection.

We often separate the Manufacturer into two records if a division of the manufacture produces exclusively models of a sub-category, such as Truck, Marine, etc.

The URLs are owned by other organizations so may be obsoleted at any time. The PFT staff updates these when it is observed that one has been changed. Mergers present the biggest challenge here because we do not time-stamp the duration of the trade names nor show evolution of the company structure over time.

A Manufacturer cannot be deleted unless it has no Models attached to it.

Engine Occupations

Engine Occupations describe the workload that the Engine is expected to perform under. The PM cycle is sometimes dependent on the Occupation that the Engine is in. E.G., Reefer and APU engines often run at constant rpm.

The primary purpose of the Engine Occupation is to provide a technique for the Fleet personnel to assign the Engines they deploy into the fleet operation to a class so that the Engines can be compared across Occupations in the Fleet. E.G., Engines in a single Occupation are compared and ranked among themselves.

An Engine Occupation does not have a beginning or end date in the system. An Engine can be assigned only one Occupation at a time.

A new Engine Occupation can be 'copied' as a New Entry (Occupation) by checking the 'Save as New Entry' button and changing the Occupation Type and Description as appropriate.

An Engine Occupation cannot be deleted unless it has no Engine Model assigned to it.

Fuel Types

Fuel Types are provided to link an Engine to the dominant fuel that it consumes. Drive fuel economy and price are very dependent on the fuel consumed.

Currently, only one Fuel Type can be assigned to an Engine, although it may be changed over time.

New Fuel Types may be added directly or by using the change function on a previous Fuel Type and clicking the 'Save as New Entry' button, making the desired change and then pressing Save.

A Fuel Type cannot be deleted unless it has no Engine Model assigned to it.

Engine Models Maintenance

Engine Models are listed 'under' manufactures of models of Engines for sale. The name of the manufacturer, model name, an acronym and an URL are available data elements. The acronym and URL are optional.

The primary purpose of the Model information is to provide data about the Engines produced under the model name to users in the cockpit layer for easier and more accurate selection. The same table is used in the Engine Polygraph application. The data that is input to the system comes from manufacturer spec sheets or other public sources.

A Model does not have a beginning or end date in the system. This means that a Model may be in production for some time period, but the produced model may be in operation for any number of years later.

Models are identified by common trade name used in the literature, often in the manufacturer's advertisements.

A new model can be 'copied' as a New Entry (Model) by checking the 'Save as New Entry' button and changing the data that is different and saving the result.

A Model should not be deleted if it is referenced in any Engine Polygraph signature! A Model cannot be deleted unless it has no Engines attached to it.

Many data elements defining the engine model specifications can be maintained by clicking on the red + above the green **CSV** button for ADD; clicking on the **pencil icon** for view or change. More details can be maintained by clicking on the **details icon**.

Engine Block Layout identifier can be input in the Engine Block column based on the naming convention: Engine Configuration + Bank identifier that contains cylinder 1 [null, Right, Left] + cylinder numbering convention [Sequential, Alternating, Uturn]. The cylinder adjacency table is provided for each Block Layout. Images can be viewed before making an assignment by opening another tab to view the Block Layouts:

1. Click on the red bar with + in the upper right of the screen to get a new view of the Engine Models screen in another tab;
2. Click on the Engines menu option and click on the Engine Layouts option;
3. Scroll down the images to find the appropriate Block Layout and copy the 'Engine Layout' code;
4. Paste the value on the original Engine Maintenance tab, Engine Block column.
5. Then Save.

Cylinder Offset Diagram cannot be directly input but is built-up from other data elements: Number of Strokes, number of cylinders, and engine firing order. The image of the Cylinder Offsets can be observed from the **details icon**.

The **Technologies** button allows viewing previously assigned Technologies to this Engine Model or maintaining the relevant entries. Engine Technologies that are significant for interpretation of Engine Polygraph waveforms for engine problem diagnosis. To add a new Technology to the list for this Engine Model, click on the 'Select technology' bar to get a list of unassigned technologies. You may enter a keyword to get a list of technologies that contain your input string; then, only technologies containing the string will be displayed. Click on the black + to the right of the selection list to add it to this Engine Model. Then click on **Save**.

The **Flaws** button allows viewing previously assigned engine model flaws or maintaining the relevant entries. Engine Flaws are commonly occurring problems with a specific Engine Model or set of Engine Models that occur with engines made between two 'dates' – from when engines with the flaw were first produced until the manufacturing process eliminated the flaw. To add a new Flaw to the list, click on the 'Select flaw' bar to get a list of unassigned flaws. You may enter a keyword to get a list of flaws that contain your input string; then, only Flaws containing the string will be displayed. Click on the black + to the right of the selection list to add it to this Engine Model. Then click on **Save**.

Engine Block Diagrams and Adjacency tables

Engine Block Diagrams are required to relate the firing order of cylinders and associated cylinder performance data with the physical locations of those cylinders. Although the special relationships are very easily seen by humans with decent eyesight, the Adjacency tables provide that information to systematic analysis of relationships of different cylinders and potential synergistic behaviors. The Engine Block Diagram and firing order are provided by the manufacturers; PNG images are loaded to the table using this functionality. The Adjacency table is generated by humans with a few rules:

1. Each cylinder has 1 row in the Adjacency table.
2. Additional columns in the row contain the number of each physically adjacent cylinder in the same bank of cylinders (share a common head gasket).

The Engine Blocks are identified by the following naming convention that consists of several components: Engine Configuration + [null, Right, Left] + number sequencing rule.

1. The Engine Configuration followed by the number of cylinders in the engine.
2. If the engine is NOT 'Inline', the code 'Right' or 'Left' to indicate whether cylinder 1 is on the right or left side of the engine where the 'side' is as observed when standing in front of the cooling fan or the serpentine belt power of the engine (flywheel is at the back of the engine)
3. If the engine is NOT 'Inline', the number sequencing rule is either 'Alt' (alternating) or 'Seq' if the naming goes increasing from '1' to the back of that bank and then continues on the front of the next bank to the back, etc. A rare orientation is called 'Uturn' when the non-inline engine is sequential from the front to the back and then continues to the back of the next bank and increases cylinder assignment coming to the front.

Cylinder 1 is set by the convention that cylinder 1 is closest to the fan if the engine configuration is 'in-line'; Right vs. Left is determined from the view of an observer standing in front of the engine 'fan side', opposite the flywheel side. If cylinder 1 is to the left side of the observer, we assign the value 'Left'; conversely, 'Right'. In US and European autos, the 'Right' designation usually refers to the driver side; whereas in the UK, Japan, Australia and some other Asian countries, 'Right' refers to the passenger side. For traverse-mounted engines, 'driver's side' or 'passenger side' are not relevant for describing the bank containing cylinder 1.

The Engine Model is assigned a Block Diagram and associated Adjacency diagram in the Engine Model Maintenance screen. An Engine Block Layout cannot be deleted unless no Engine Model is assigned to it.

Engine Cylinder Offset Diagram

The Engine Cylinder Offset diagrams show the connection of the cylinder firing order and crankshaft angular rotation. This diagram depends on the number of strokes in the engine cycle (the time of the first power stroke in cylinder 1 to (but NOT including) the next time cylinder 1 initiates a power stroke).

The Cylinder Offset diagram is NOT dependent on the Engine Block diagram but is dependent on the firing order.

This diagram is important to understand data when an engine is not running properly so that sensors might be detecting inputs from several cylinders in different strokes. The constraint that the sensor reads data instant-by-instant allows reference to the diagram to see where in each stroke each cylinder is in at that time (crankshaft angle).

The Engine Model is assigned an Engine Cylinder Offset diagram 'key' in the Engine Model Maintenance screen.

An Engine Cylinder Offset diagram cannot be deleted unless it has no Engine Model assigned to it.

Engine Technologies

Any Engine Model employs a number of Technologies. Many of these are present in all internal combustion engines. The list of Technologies maintained here are quite new and important in understanding the pressure pulses analyzed for any engine of this particular Engine Model.

The functionality defined here provides ability to add a (new) Technology to the table so it may be assigned to any appropriate Engine Model as described earlier. Any Technology consists of a code, a Description, and an optional parameter.

The code is an acronym or word string that is likely recognizable by any mechanic. The Description provides some words/sentences to more completely define the Technology and distinguish it from closely related Technologies.

The optional parameter is applicable to some Technologies to specify operational features for use in any signature analysis from engines with this Technology.

Flaws that are assigned to Engine Models are presented on Diagnostic reports for engines by Engine Polygraph.

The screen allows the addition of a new Technology by clicking on the **red +** in the upper right of the Engine Technologies screen to get a 'blank' screen or by click on the pencil icon of an existing similar Technology and selecting the radio button 'Save as New Entry' where only distinguishing features need to be added. The columns of the screen are language specific, so it is assumed that the language specified in the upper-right Language pull-down is being used to describe the Technology. Save after the inputs have been completed.

The **engine icon** in the Action column displays a list of Engine Models that have been assigned the Technology described on that row.

Choose the **pencil icon** to display the data for the Technology on that row for update if the user is so authorized. Save after the inputs have been completed or Exit to go back to the 'list' screen if no changes were made or are not to be saved.

Choose the **large X** to display the data for a Technology that you wish to delete. A delete is allowed only if no Engine Model is currently assigned to this Technology. Save if this Technology should be deleted (usually after realizing that a new add is really a duplicate).

The **engine icon** is used to see a list of all Engine Models that are currently assigned this Technology.

Annual Data Updates

Engine Technologies of significant impact are available from Wards/Mahle (www.WardsAuto.com). The Technologies that we are currently interested in are organized around Aspiration, CAM operation, Cylinder deactivation, Fuel Delivery, and Valve Control. The data from WardsAuto also includes identification of metals of construction for major components of the engine: Block, manifolds, push rods, rocker arms, etc.

The annual WardsAuto data is presented as a spreadsheet (Pivot table) with columns for Engine Manufacturer (Maker), Model identification as presented by the Maker, number of cylinders, Displacement, Bore and stroke, number of engines produced in the year, etc. with a string of Technology codes separated with a '/'.

Mass update can be done by parsing the string of Technology codes and pairing with engine data to match with the EngineModel table rows. The codes of Technologies can be aligned with the emtId of the Technology Types in the EngineTechnology table. The EngineModelId and emtId pairs can be inserted (don't add duplicates) to the EngineModelTechConn table in the EngineAngel database.

Engine Flaws

Any Engine Model may exhibit any number of Flaws. Many of these are present in more than one internal combustion engine model. The list of Flaws maintained here are reported as recalls or by mechanics reporting on public blogs. Flaws that are assigned to Engine Models are presented on Diagnostic reports for engines by Engine Polygraph.

The functionality defined here provides ability to add a (new) Flaw to the table so it may be assigned to any appropriate Engine Model as described earlier. Any Flaw consists of a Code, a Component, a Description, a Date range and reference for more detail.

The Code is an acronym or word string that is likely recognizable by any mechanic, especially in reference to the specified Component. The Description provides some words/sentences to define the Flaw and distinguish it from closely related Flaws.

The Reference is applicable to some Flaws to document the recall or mechanic's observations.

The screen allows the addition of a new Flaw by clicking on the **red +** in the upper right of the Engine Flaws screen to get a 'blank' screen or by click on the pencil icon of an existing similar Flaw and selecting the radio button 'Save as New Entry' where only distinguishing features need to be added/changed. The columns of the screen are language specific, so it is assumed that the language specified in the upper-right Language pull-down is being used to describe the Flaw. Save after the inputs have been completed.

Choose the **pencil icon** to display the data for the Flaw on that row for update if the user is so authorized. Save after the inputs have been completed.

Choose the **large X** to display the data for a Flaw that you wish to delete. A delete is allowed only if no Engine Model is currently assigned to this Flaw. Save if this Flaw should be deleted (usually after realizing that a new add is really a duplicate).

The **engine icon** is used to see a list of all Engine Models that are currently assigned this Flaw.

Expense Items- Background

Expense Items in the Admin Layer identify manufacturers and their models for use by Engine Angel customers.

This data is collected and provided by Predictive Fleet Technologies to customers as an aid in the efficiency, accuracy and consistency of Expense Item identification & specification

The purpose of the Expense Item data is to reduce cost and time for data loads of customers and to provide consistency in the coding to support on-going customer evolution to increased system functionality and comparison of Expense Items (performance, reliability, etc.) between models.

Any user in the Admin Layer (Predictive Fleet Technologies employees only) who has been granted Expense Item Authorization can Add, Change, and sometimes Delete records and data content. All Admin users may view any of the data on these screens.

Much of the data is available via 'pull-down' lists in the Cockpit and Operations layers. Data content is made available where it supports implemented system function.

Axle Types

Axle Types describe the features for each axle type to aid in the Chassis configuration. Key features are:

- Description
- Number of axles in a group (e.g., tandem or tridem)
- A Position code [steer, drive, trailer]
- Image of an example.

All axles have bearings, brakes and rims. Any axle might be a 'lift' axle.

Steer axles require linkage to the steering mechanism to be checked periodically.

Drive axles have a differential that need lubrication and occasional replacement.

Trailer axles usually have distance measured with hubometers.

Tire Brands

Tire Brands and manufactures are identified in this table. Tires are defined more by the Brand than the Manufacturer because the Brand provides the specification and might employ various manufacturers to produce tires under the Brand specification. The name of the manufacturer, abbreviation and Brand Name are stored with the URL for internet data. The acronym and URL are optional.

The primary purpose of the Tire Brands information is to provide data about the Tires produced under the Brand name to users in the cockpit layer for easier and more accurate selection. The Tire Lines (designs) are connected to the Brands and the Tire DOT Codes are connected to the Manufacturing plant to aid in recalls.

A Tire Brand does not have a beginning or end date in the system. This means that a Tire Brand may be in production for some time period, but the produced model may be in operation for some years later.

Tire Brands are identified by common trade name used in the literature, often by the manufacturer's advertisements.

A new Brand can be 'copied' as a New Entry (Brand) by checking the 'Save as New Entry' button and changing the data that is different.

A Tire Brand cannot be deleted unless it has no Tires attached to it.

Tire Lines

Tire Lines are connected to the Brand that manages the Tire specification. The Tire Lines under the Brand define specific designs and sizes for various axle types.

A Tire Line does not have a beginning or end date in the system. A Tire Line can be assigned only one Brand at a time.

A new Tire Line can be 'copied' as a New Entry (Tire Line) by checking the 'Save as New Entry' button and changing the Tire Brand and Tire Line as appropriate.

A Tire Line cannot be deleted unless it has no Tires assigned to it.

DOT Tire Codes

Tire DOT Codes are provided to identify the Brand and location of manufacture of the tire to aid in case of recalls. The DOT Code and the week & year of production is branded on each tire sold in the US.

Format on Tire: 'DOT' indicates the tire meets DOT safety standards; 2-3 characters (DOT Tire Code) identifies mfg plant code; 2 digits = tire size; Brand code; Date of Mfg = week number of year and last 2 digits of year.

A new DOT (Plant) Code may be added directly or by using the change function on a previous DOT (Plant) Codes and clicking the 'Save as New Entry' button, making the desired change and then pressing Save.

A DOT Code cannot be deleted until there are no Tires attached to it.

Brake Brands

Brake Brands are in the BrakeManufacturer table which really stores the Brake Brand names.

A Brake Brand does not have a beginning or end date in the system. A Brake Brand can be assigned only one Brand at a time.

A new Brake Brand can be 'copied' as a New Entry (Brake Brand) by checking the 'Save as New Entry' button and changing the Brake Brand info as appropriate.

A Brake Brand cannot be deleted unless it has no Brakes assigned to it.

Battery Brands

Not yet implemented.

Inventory Expense Groups

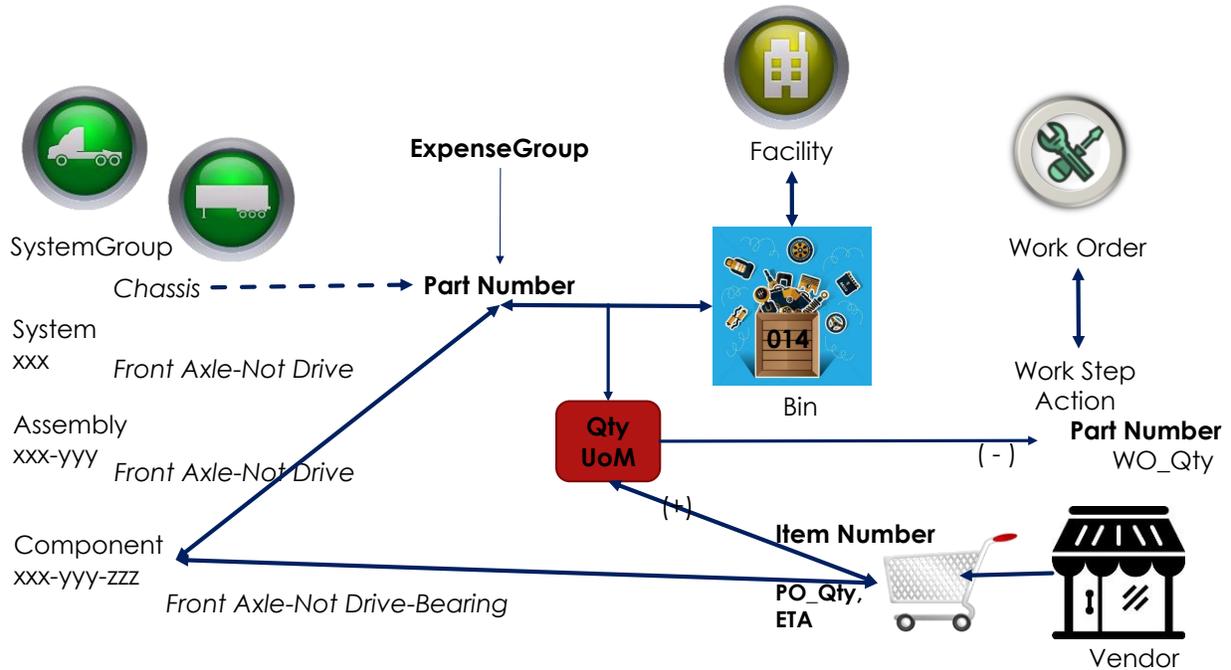
Not yet implemented.

Inventory Expense Groups (IEG) organize expense items into similar groups. For example, engine oils and hydraulic oils might be under lubricants.

IGEs relate to RepairSystems, a CodeList in the Classification system. Repair systems are components of the vehicle or trailer (unit) that together provide a specific feature to the unit. For example, the chassis includes the axles, bearings, suspension system, brakes, wheel rims and tires.

Expense items that are in inventory must be assigned to an Expense group and to at least one Repair System.

Relationships between Data Objects of Inventoried Items



Classification

The Classification sub-menu contains a set of system functions to define classes of items in the engine polygraph application and a large number of classes as Code Lists. The classes are used largely to populate pulldown lists in the Cockpit and the Operations layers. As such they are enabled to have the class names translated into the languages supported by Engine Angel.

The purpose of the Classification data is to reduce cost and time for data loads of customers and to provide consistency in the coding to support on-going customer evolution to increased system functionality and comparison of the various elements of the organization and between organizations.

Any user in the Admin Layer (Predictive Fleet Technologies employees only) who has been granted Classification Authorization can Add, Change, and sometimes Delete records and data content. All Admin users may view any of the data on these screens.

There is no breakdown nor hierarchy of the classification authorization; i.e., a person who can update any one of the classification elements, is authorized to maintain all of the classification elements.

Much of the data is available via 'pull-down' lists in the Cockpit and Operations layers. Data content is made available where it supports implemented system function.

Engine Configurations

Engine Configurations are codes that describe the architecture of an internal combustion engine block. The codes follow a quite standard industry structure: a letter representing the shape of the engine block, and a number representing the number of cylinders in the engine.

These **codes** are to be presented in the pulldown lists; this is unlike many classifications where the codes have a description that might be translated into multiple languages and populate the pull-down lists.

The Engine Configuration codes are attached to engine models in the Engine Model master database and in the **Engine Polygraph** application.

New codes can be added by clicking on the red '+' in the upper right of the screen or by clicking on the pencil, and then clicking on the "Add as new entry". Descriptions can be changed by clicking on the pencil and changing the text on the detail screen. A code can be deleted, provided it is not used elsewhere, by clicking on the red 'X' to the far right of the screen.

Engine Conditions

Engine Condition codes are used to describe the operating condition of the engine of interest. It is very important not to change these codes, because the codes are used in the programs to control the flow of the logic.

The Engine Condition codes are attached to engine models in the Engine Model master database and in the **Engine Polygraph** application.

The descriptions of the codes will be displayed to the user's in the appropriate language, but it is important to use the codes themselves when passing data between applications.

Channel Position

Channel Positions are used to identify locations where various engine sensors are deployed for testing an engine. The most commonly used positions for **Engine Polygraph** are the 'exhaust' and the 'oil tube', meaning 'oil dipstick tube'.

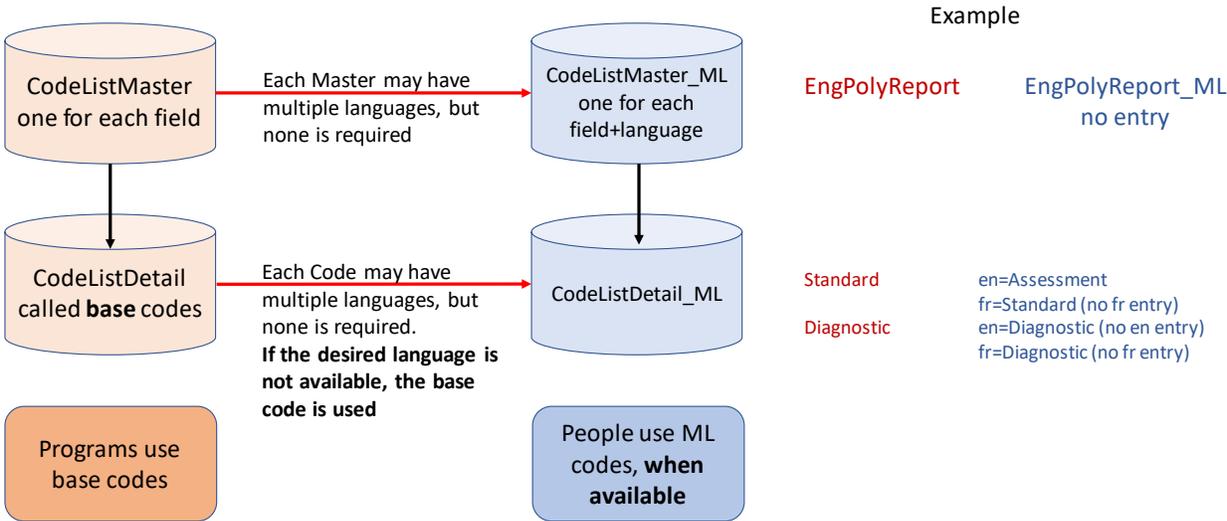
New codes can be added by clicking on the red '+' in the upper right of the screen or by clicking on the pencil, and then clicking on the "Add as new entry". Descriptions can be changed by clicking on the pencil and changing the text on the detail screen. A code can be deleted, provided it is not used elsewhere, by clicking on the red 'X' to the far right of the screen.

Code Lists

CodeLists are lists of codes to provide user interface options for specifying inputs or for the system/data to display information to the user. The purpose of bundling the many codelists (53 at the time of this writing) into one table and maintenance tool is standardization of programming and User interface.

Codes are defined in the CodeMaster and CodeDetail tables to organize various text parameters:

- a. The Master defines the concept being specified
- b. The Detail lists the acceptable options; the code values can be used for controlling logic in programs
- c. The `_ML` tables store the code values in the users' language for more accurate selection.
 - i. There are the Code and the CodeDescription that are translated. The `ML_Code` values are for display on screens and reports
 - ii. The CodeDescriptions are used by the translator and eventually to be incorporated in the Help system that is not yet in place
- d. **If there is no `_ML` code entry, use the base (CodeListDetail) code**



CodeLists have a Master 'header' record to provide a list name, description and purpose. The code names and descriptions can be translated into various supported languages.

Only PTF employees can maintain these CodeLists, but the Codes and Code descriptions are available to the customer's employees.

CodeLists are maintained using a modified approach to most other data maintenance screens. The Master List screen shows data from the Code List headers. New CodeLists can be defined by clicking on the red + in the upper right of the screen. The Description and Purpose can be changed by clicking on the pencil to the right of the line; and a CodeList can be deleted (if it has no codes under it) by clicking on the X at the far right of the screen.) That much looks similar to other data maintenance functions.

But the codes under the CodeList are displayed by clicking on the line describing the CodeList. Codes can be added and deleted (if unused in the other databases) or text changed in the standard methods of + , pencil, and X on each line.

The Maintenance screens are different from other parts of Engine Angel in that translations can be provided for the description in other languages.

After a Code has been added, a Language can be added. Initially, the English Description will be provided and may be translated into the 'target' language. This process can be repeated for all required languages.

In a similar fashion, the CodeList Description (Title) can be translated.

Presentation of Codes to users:

- There should be common code for the retrieval of the values to present to users
- **Database records** should only use the entry from the CodeList**Detail** table
- **Displays and reports** should use the value retrieved from the User language (for screens), the Fleet language for Fleet reports, 'en' for reports where no language code is available, or as otherwise specified
- For **Pulldown lists**, get all the Detail (base) codes under the CodeListName, then find the equivalent for the appropriate language. The ML table has the id to the equivalent CodeListDetail so that plus language code provide quick access to the ML value if present. If not present, you already have the base (Detail) value. But store the base value in any database record.

Security Data

All People

The All People functionality allows authorized PFT employees to see and change 'People' data from any Fleet and to Add or Change information about Predictive Fleet Technologies employees, including authorization and status changes.

The primary purpose for the access of customers' employees is to enable Technical Support to customers who may have experienced critical employee turnover without adequate training of new employees.

Fleets

The Fleets screen allows authorized PFT employees to view and change data about all Fleets in the Engine Angel system.

The primary purpose is to enable Technical Support to customers who may have experienced critical employee turnover without adequate training of new employees.

Account Requests

Account Requests provide a listing and details for all un-processed requests from prospects to obtain a subscription to Engine Angel.

The options are to change and Save the changes about the request, Exit the screen without saving changes, or Send an invite to access the system.

To proceed to the invite, Fleet information must be provided to create the basic records in the Engine Angel system to enable the requestor to login.

Localization Data

This Localization data is provided to enable standardization of data geographically and culturally as well as chronologically.

Where possible, we base our coding on ISO standards but also provide cross-walks to various industry and regional conventions.

Here we provide the options and in many cases, methods for conversions between standards. In a number of cases, the relationships change over time, e.g., Currencies and Currency Exchange rates.

The options are provided to the Fleet customers by means of pull-down lists.

Countries
State/Provinces
Postal Codes
Global Facilities

Languages
UoM Rule
UoM Data
Currencies
TimeZones

Languages

The Language table provides the list of languages that can be selected in Engine Angel and Engine Polygraph. That does not mean that the translations are complete in those languages, but that the system is prepared to accept the translations.

The Language code is the ISO two-character, lower case code that is connected to a 'Display Language' value that is the ISO code followed by the name of the language expressed in that language and to the 'Locale code' that combines the language code with a ISO Country code (3-character, capitalized letters).

The multilanguage capability is supported by several constructs:

The screen literals are stored in a Content table and associated Content_ML for language-specific renditions for the term.

The system messages to the User in the GUI are stored in a GUIMessages table and associated GUIMessages_ML table.

Codes and Code Descriptions for pull-down lists are stored in the CodeListMaster & CodeListDetail tables with the associated _ML tables.

Other tables with language-specific fields that are to be made available in multiple languages are stored in the 'base' table and associated _ML tables.

TimeZones

TimeZones and associated data to define a TimeZone are made available in the TimeZone table to enable shifting 'local' times of events to the Fleet timezone so that events can be synchronized across the Fleet operation.

TimeZones can be 'Standard' or 'Daylight Savings'. Most Daylight Savings times are for only part of a year and so require Daylight Savings Rules. Daylight Savings Rules can be 'Fixed', meaning that the shift to and from Daylight Savings always occurs on a specific day of the year; or 'Variable', meaning that the shift occurs on a specific Day-of-Week (usually a Sunday early morning).

TimeZones

TimeZones and Daylight Savings Rules are assigned by various levels of political divisions within an area, with Country being the highest level and going down to the Postal code area.

TimeZones are assigned in a number of places in the system. When a more precise location is identified, the TimeZone might be updated:

- Country default
- State/Province default
- Postal Code
- Geocoded address for addresses

TimeZone are shifts in the UTC (Universal Time Coordinated) that are adopted by various levels of government over the territory of the government. The level of detail that may choose a TimeZone varies by country: many TimeZones are assigned to an entire country; some allow variation by State/Province; some go down to Postal code.

There must be at least 24 Standard TimeZones but many have Daylight Savings Times (Summer Times) and these often are coordinated by neighboring countries. If a TimeZone has a Daylight Savings Time rule (DST), the Rule name is provided in the TimeZone table. The TimeZone row specifies the amount of time shift relative to UTC (often called Greenwich Mean Time).

TimeZones Variable Daylight Savings Time (DST) Rules

A DST Rule specifies the date that the DST Rule starts to modify the Standard time and to return back to Standard time. In addition, the Rule specifies how much time is 'shifted' between Daylight Savings and Standard times.

Each Rule is given a 'name' that may be referenced by any TimeZone that is not a Standard TimeZone. The Rule has an Effective Date that indicates when the Rule starts to apply and will continue to apply until another Rule with a later effective data takes over.

The shift is represented in $\pm hh:mm$ and the date in Variable Rules define the Day of Week, Week of Month, and Month of Year and hour of day for the switch – for the start of the Savings time and the return to Standard time.

TimeZones Fixed Daylight Savings Time (DST) Rules

A DST Rule specifies the date that the DST Rule starts to modify the Standard time and to return back to Standard time. In addition, the Rule specifies how much time is 'shifted' between Daylight Savings and Standard times.

Some countries stay on DST and so use a Fixed Rule to indicate that.

Currencies

Currency tables are made available to prompt/convert inputs of financial transaction in the Fleet currency. Currency conversions will be performed by SaaS transactions for situations when current conversion rates are indicated.

The currency table contains the ISO Currency Code, a name, name of the unit of currency, commonly used symbol, and the StartDate when it was (will be) put into circulation and an EndDate (9999-12-31 if unknown). There is a row for each 3-character ISO Country code to show the legal currency for that country.

Countries

The Country table provides the standard identifiers for Country codes and names in Engine Angel.

The Codes are upper-case English letters. We use the 3-letter ISO codes in the pull-downs. The 2-letter codes are provided to support 'cross-walks' to interfacing systems as needed.

The Country code provides geographic information as well as legal and financial context.

The Country table stores the legal Currency Code of the Country.

In addition to the English name of the country, we provide a default TimeZone code that might be valid for the entire country, otherwise, the TimeZone column should have a NULL value for the Country.

The Country code is a prerequisite code for the State/Province table to group all political subdivisions within the Country.

State/Provinces

The State/Province table stores the names and codes for the political sub-divisions of the Country. These may be Counties or go by other names – but the intent is to standardize the sub-divisions immediately below the Country national) identification.

The Country name and the State/Province names are presented in the table. In addition to the English name of the State/Province code, we provide a default TimeZone code that might be valid for the entire region; otherwise, the TimeZone column should have a NULL value for the State/Province.

The Regional Dept of Transportation (DOT) URL is provided to obtain data about road conditions. The Date of last successful access is stored to support occasional reviews of the persistence of the URL.

Postal Codes

The Postal Codes are assumed to be managed at the national level. The Country code, Postal code, and State/Province code are provided to show the hierarchy.

The Postal Code Place name (usually City) is made available in a Pull-down list to increase consistency in address specification.

The TimeZone, GPS coordinates, and the Altitude are all stores as available defaults. A Postal Code status and type are maintained to indicate possible restriction on the use.

Global Facilities

Global Facilities are made available with system function very much like the Cockpit > Master Data > Facilities management, except that here, the Facilities added from this screen are available to all Fleets (within a Fleet's assigned Country).

UoM Rule

UoM Rules define exceptional Unit of Measure values in specific table columns. The Basic Rules are:

UoM for numeric values in the Admin Layer are metric,

UoM for numeric values in the Fleet Layers (Cockpit and Operations) are specified by the Fleet table (Cockpit > Account > Dashboard > Fleet Information).

Unless otherwise specified, the UoM is defined by the 'Base' UoM for the assigned dimension.

An Override Rule can be assigned to a specified column by selecting the table and then the column in that table. Values are to be provided for each of the 3 allowed UoM systems in the Fleets: [Metric, US, Marine]

Unit of Measure Rules define specific values for various types of data elements and in the case of traditional Units of Measure, conversion equations between units that are in the same Dimension and linearly related.:

Numeric: Numeric values can be unitless (e.g., counts) or dependent on a specified device to measure the quantity in a unit of a dimension; the relevant data are stored in the UoM Data table. Here we identify which UoM is to be used for the Fleet UoM System, as specified in the Fleet configuration.

If the Fleet specifies '**Marine**', distance and velocity will be stored in the unit identified in the 'marine' column.

If the Fleet specifies '**Metric**', the unit in the 'metric' column is to be used.

If the Fleet specifies '**US**', the unit in the 'US' column is to be used.

TimeZone: The TimeZones of the World and Fleet customization are stored elsewhere. Here we store whether the Fleet Language is to be stored or if UTC is to be used. (UTC is the standard for times in the Global tables and System columns (e.g., LastUpdateDateTime).)

GPS coordinates: Here we specify which of the two formats of GPS location is to be used: DD = Degree Decimal, or DMS = Degree-Minute-Seconds. A standard function is to be used to perform the conversions between them for consistency. DD is the format to be used in the 'Global' tables.

Language: The language codes and descriptions are stored in the Language table. Here we specify if the Fleet language or English is to be used.

Currency: *Details for conversion and options are stored in the Currency table. Since the conversion rates dynamic, rules for which rate to use must be specified [Current, Budget (time period), transaction date, etc.]. Here we identify 'Fleet' currency or USD.*

UoM Data

The UoM Data table contains common acronyms and symbols used to specify units of measure. Many of these are set by ISO sub-groups, but others are conventions of use in various disciplines.

The table also identifies a 'Base' UoM for each Dimension; that Base unit is used to enable 'pivot' conversions from any UoM to any other UoM in the same Dimension.

Each 'metric' UoM is provided the typical equivalent UoM in the non-metric system, and vice-versa.

The coefficients of the linear equation that convert a quantity in one UoM to another UoM in the same dimension are provided. There is a standard function that employs these coefficients to perform the conversions.

The UoM maintenance screen has a Test feature to convert an input number from the Test Input Number of Unit Code to Calculated Base Units. The program also shows what would be the result if the input number were in the base units and calculates the equivalent number of selected code units and puts the result in Reverse calculation of input Base units to unit Code.

The System of Record is provided as a reference to check the accuracy and completeness of the data.

Alerts/Content

CMS Editor

The CMS Editor is a WordPress editor used to maintain the Home Page screen.

Email Templates

Email templates are provided to standardize various communications from the system to Users and potential Users. The language specific texts are provided in multiple languages for several purposes:

- The Account Invite message welcomes a new prospect to the EngineAngel application for creating the Fleet in the system.
- The Reset Password request message is sent to the user who requested to reset his/her password.

Other common messages that are to be provided in multiple languages are to be maintained with this method.

Alert Criterion

Alerts/Content Data

As data is analyzed and events predicted in Engine Angel, data & predictions can be compared with triggers setup by the Fleet management for each Vocation to have the system alert them to situations that deserve their attention.

The alerts will be sent via email and logged into a Fleet table of Alerts sent.

Alerts will be classified as 'Important' or 'Urgent'. (Urgent are usually 'past due' items.)

Store Alert texts (Language specific) in Admin Layer under **WebSiteContent**

Incorporates variables from the event (embedded '{{0}}' by calling program)

Alert Hierarchy

1. Criteria for Alerts **are grouped** into AlertCriterionGroups that are defined in a CodeList: AlertCriterionGroup
2. An Alert Criterion **is assigned** to an AlertCriterionGroup in Admin level > Alert Criteria
3. The Alert Criterion **is defined** in Admin level > Alert Definitions

SourceSystem (must be established on CodeList: AlertSourceSystem) identifies which function in EngineAngel has the code to test operational values against Criteria to trigger the Alert.

SourceKey (free text) identifies what variable is tested.

Description (free text) of the situation when the Alert is triggered.

Priority (on CodeList: Priorities) indicates the priority of the Alert in terms of which of multiple Alerts should be addressed first.

Format (CodeList: MsgFormat) indicates the content format of the message content of the template for this Alert.

Routing (Role) of people to receive the Alert in this Fleet) – assumes there may be more than one person in the Fleet's Role.

Criteria are 'configured' for the Fleet by two functions under the Fleet Maintenance menu item: Cockpit > Account > Account Dashboard > Manage Fleet (pencil) > Fleet Information > Fleet Alert Criteria.

Fleet Alert Criteria are set for Vehicles assigned to particular Vocation. (Extreme conditions vary for the equipment and driver behaviors depending on what the job is: Line haul, delivery, construction, passenger transportation, etc.)

The criteria available (that have been programmed) are obtained from the Admin level Criteria Group > Alert Criterion

The Values assigned by the Fleet are assumed to be in the Fleet's UoM system.

A specified 'Low' value means if the observed value is less than the indicated value, the Alert should be generated, If there is no 'Low' value in the Fleet Criterion for this Vocation, ignore this test.

A specified 'Hi' value means if the observed value is greater than the indicated value, the Alert should be generated, If there is no 'Hi' value in the Fleet Criterion for this Vocation, ignore this test.

Alert Definitions

Alert Definitions are descriptions of the Alert situation that are sent to appropriate personnel in the Fleet based on assigned Roles. The list of PFT provided messages are listed with many of the codes describing the Criterion, Source system, Source Key, Priority and Title along with the Role of a person designated to receive (and resolve) the issue.

The add, Change or Delete can be requested for any Definition. Notice that the priority can be different for different values of 'trigger' variable. Usually, 'urgent' is reserved for issues that are past 'End of Life' (EoL).

The details screen allows specification of the Alert and the text of the Alert message to be presented. The messages can be text strings or HTML format. The content can include variables from the data associated with the Alert. Variables to be included must be surrounded by double brackets: '{{' and '}}'. Any translation of the Alert must employ the same variables but the syntax of the language will dictate the sequence.

Alert Routing

Since texts are in batch, the messages will be in the language of the To: FleetRolePerson (Default to the Fleet Language) as contained in table: FleetAlertRouting

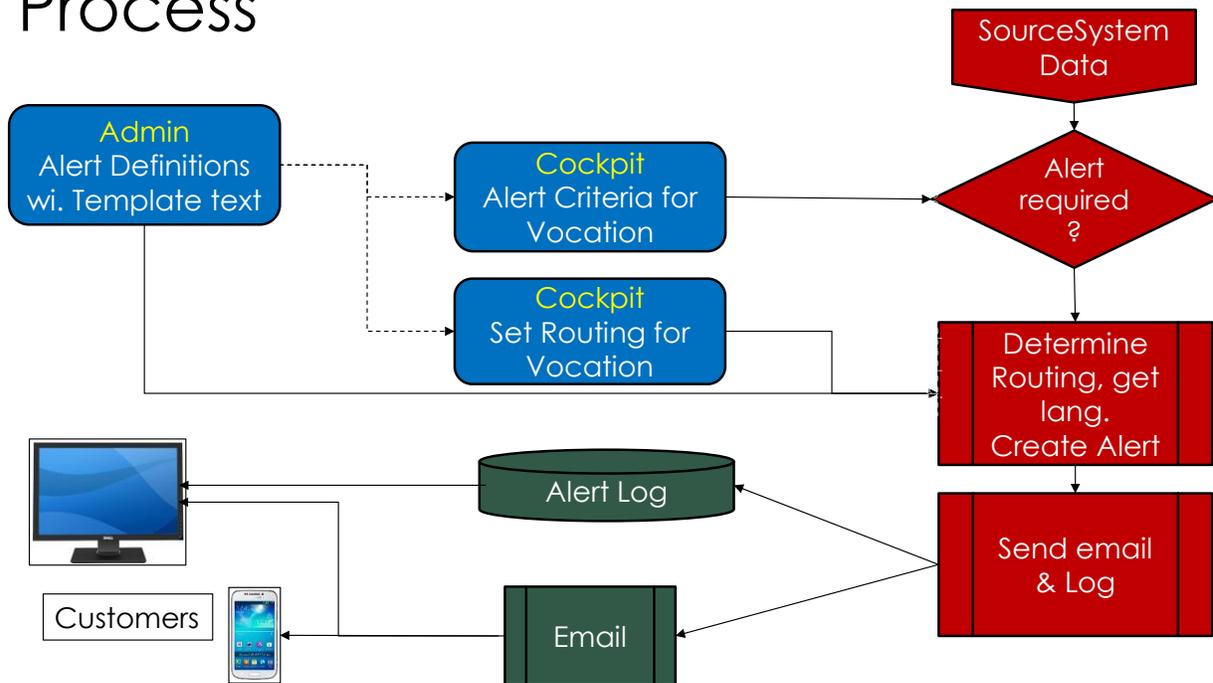
The Alert will be associated with a Fleet 'Role' default assignment to all people assigned 'Role' in Fleet

Alerts are responsibility of only one Role

Fleet can alter the distribution by specifically routing to UserName/email address by Vocation of Vehicle

It is important that system doesn't flood employees.

Process



Diagnostic Text

Diagnostic Text is a table to describe Abnormal Observations (AbnObs) from diagnostic procedures, Causes (Cuz), and Suggested Remedies (Fix) to be printed appropriately on Engine Polygraph and Engine Angel Diagnostic Reports. The functionality under this tab allows the addition of new texts, modification of existing texts, and the Deletion of unneeded texts. In addition, Causes can be listed under Abnormal Observations and Suggestions can be connected to Causes.

The Diagnostic reports display likely Causes followed by attached Abnormal Observations to show the rationale for selecting the Cause as likely. In addition, the Suggested Remedies attached to the Cause are displayed for the user to consider.

The text rows are organized by several concepts:

- **Sources** refer to the origin of the AbnObs and dependents. The allowed values are stored in the CodeList, OCF_Source and include things like Engine Polygraph hot engine analysis (EPhot), Cold-crank analysis (EPcold), Engine oil test results (EngOilTR), Technician answers to questions (Person), and analysis warnings (caveats) from the exploratory data analysis (EDA) in the EA analysis engine.

- **Diagnostic Text Types** are assigned to each row in the table and are from the CodeList, DiagnosticTextType: obs, cuz, fix. The identifier controls the management of allowed linkages between the texts.
- **Ignition Method** indicates if the text (observation, cause, remedy) applies to All ignition types, spark ignition engines, or diesel engines. It is controlled by CodeList, IgnitionMethod. Some causes apply only to spark ignitions, some to only diesel ignition, but most apply to 'All' internal combustion engine ignitions.

The List screen of texts displays the short description and long description satisfying the conditions of the filters. The list can be downloaded to a file by using the button, csv, above and on the right side of the list panel.

The **action** column entries indicate desired actions the authorized user may perform:

- New texts can be added by first clicking on the red + in the upper right corner of the page.
- Click on the 'details' icon to show all texts that have been attached to that row of text.
- Use the pencil to modify the text and short description and change the classification codes.

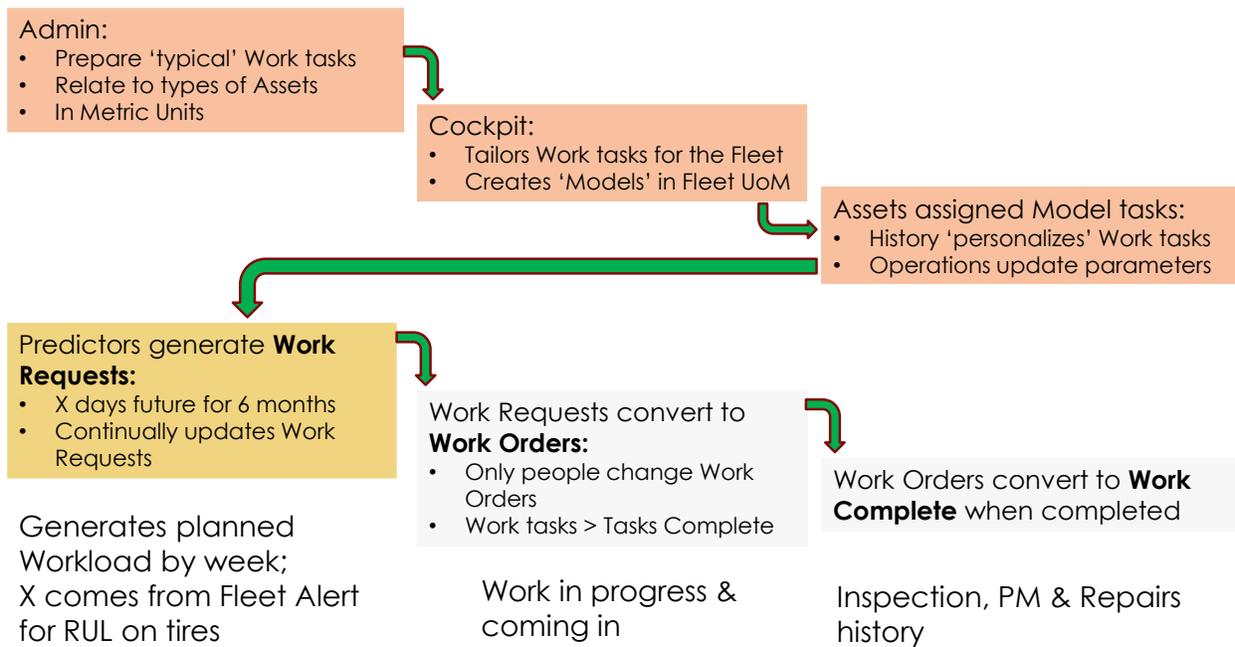
The modification screen allows connection of allowed texts and display of all 'parents' to the row selected. All allowed unselected choices are displayed under the original text. Click on the red + on the right to add a choice to the original text. The list of connected texts is displayed below the candidates. Any connected text can be deleted by clicking on the X on the right and verifying the request.

The 'parents' of the original text can be displayed using the other button below the connections.

Work Request (Templates)

Work Templates are defined at the Admin Layer to provide a starter set of Preventive Maintenance, Inspections, Work Order for repairs and betterments, etc. for any Fleet to maintain. Once a template is 'completed', it becomes visible to the Fleet on the Cockpit > Dashboard > Work Forms tab for download. Once downloaded, the Cockpit user may tailor the document to the standards and needs of the Fleet requirements. When a Work Form is 'completed', it becomes available in the Master Data > [Vehicle, Trailer, *Driver*] > Work Request Forms. Here, once the form has been approved for use by the technicians, it is marked 'completed'; it may then be 'assigned' to specific Assets [Vehicle, Trailer, *Driver*] or to reference vehicles with a leading 'REF: ' in the Vehicle Number.

The Assigned Work Request Forms are associated with 'wear rate' prediction rules, incorporating data from trips and inspections to predict the Remaining Useful Life (RUL) and End of Life (EoL) for the component base on forecasted daily usage of the Asset. The PM should be scheduled to be completed before that EoL Date.



Data Flow of Work Documents

Work Request – Data¹

The data in a Work Request is in two types of tables:

The header provides:

Title – Descriptive identifier to aid in retrieval of the Work Order Form

Work Type – from a CodeList that defines the prediction logic for the specific Asset component wear-rate.

Schedule – Suggests the primary type of logic used to schedule the Tasks of the Work Request.

Status, Form No., and Version for document control.

Resource Type, Asset Type, and Vocation to provide context for the Asset usage

Typical intervals in terms of distance, hours of operation (HoO), or Calendar days (as for license renewals or government filings)

Typical incurred costs such as hours the Asset is Out of Service for the PM, Labor hours involved in the PM, and cost of Parts & Supplies for a typical WR.

The second type of table in a Work Request contains parameters specifying the data elements for the Tasks of the Work Request; note that it does NOT specify the contents of the Task lines.

The Detail Lines define each Task to be performed and documented using fields specific to the task:

Section & Seq (number) isolate the step into sections & tasks within the section

Title indicating the action to be taken

AxleMap, BrakeMap, TireMap indicators if the 'maps' are to be provided for the technician to record the data

CSA to show the CSA points involved with an inspection showing non-compliance with the stated claim for the task

Tech Condition Comment on the results of the task

Action

Measure

Part Number

Picture

Damage Cause

Part Unit Cost

Part Units

Labor hours, DownTime (in hours)

The marked indicators specify the data elements that are expected to be available for input on the final document.

Wizards Data – Processes

Subscription Types

Remote Data

Data Connects

DataConnects documents the established (supported) interfaces from 3rd parties to Fleets. The details of any interface between any Fleet and 3rd parties are stored at the Cockpit layer.

The list screen shows the set of eligible interfaces with some description and link to the software that provides the interface. The list of supported interfaces is controlled by table DataConnectType.

The current supported interfaces are:

1. Collecting, transmitting, and integrating ECM data from Detroit engines with DDEC software.

2. Accepting EnginePolygraph® (EP) signatures from PicoScope oscilloscopes into the web-based Engine Polygraph application.
3. EPReader: Here we provide the software for deployment on shop PCs to collect and transmit EP signatures to the EP Analytics and integration of the results into the EA Engine Condition reports.

Remote Data - Processes

In the future, we intend to include integration with additional OEM ECM softwares, *Oil Test Results*, and with *Telematics* (with Electronic Logs) .

Translation

Translate CodeLists

The Translations tab currently has one entry: Translate CodeLists.

The purpose of **Translate CodeLists** is to provide a tool enabling the use of XML spreadsheets to provide changes in CodeList content to translators and accept their new translations for updating the CodeList tables.

The Translate CodeLists screen contains two panels:

1. PROVIDE DETAILS TO GENERATE EXCEL

This screen provides an XML spreadsheet for Excel to show all codes in the CodeList system with an English description and a description for the indicated language and such that either has been modified 'recently' (within the last 'No. of Days' input on the screen).

2. IMPORT TRANSLATED FILE

The second panel allows the user to specify the name of an XML Excel table that contains updates to the Codes and Code Descriptions from the translators to update the table.

For a code to be considered for the download, it must have:

1. An English segment with the Code and Description in English
2. A segment for the specified language (perhaps filled with the base code and description or with an entry translated by Google or someone else)
3. The Last Update DateTime for the English or the specified language must be more recent than the current date minus the input 'No. of Days'.

The downloaded XML file is to be opened in Excel as an XML table. Changes to the table are to be made in the last three columns on the right of the rows: Date, Code, and Code Description. Date should be the Date that the update of the row is made and the rightmost Code and Code Description are to be changed to the appropriate translation.

If no changes to a row are made, that code will not be updated in the table when later uploaded.
The Excel file must be saved as “XML data” from the SaveAs option.

