MANITOU API

Connected Solutions - Using the service

Jan 15th 2021 - version 2.0

Description : **this document provides information on how to obtain the best results of the MANITOU API service for Connected Solutions range of API products. This documentation is for customer IT teams to understand the service's design and know how to use its features, upon agreement signature and terms & conditions acceptance.**

Preamble

MANITOU API is a data service in the form of APIs (Application Programming Interface) that provides the Customer access to protected resources, in coherence with the Customer's service level of subscription.

These APIs provide a wide range of features that help improve efficiency and productivity. All the features rely on the principles of RESTful APIs, which consider every accessible item as a resource with its unique id that can be used and reused, using a set of methods.

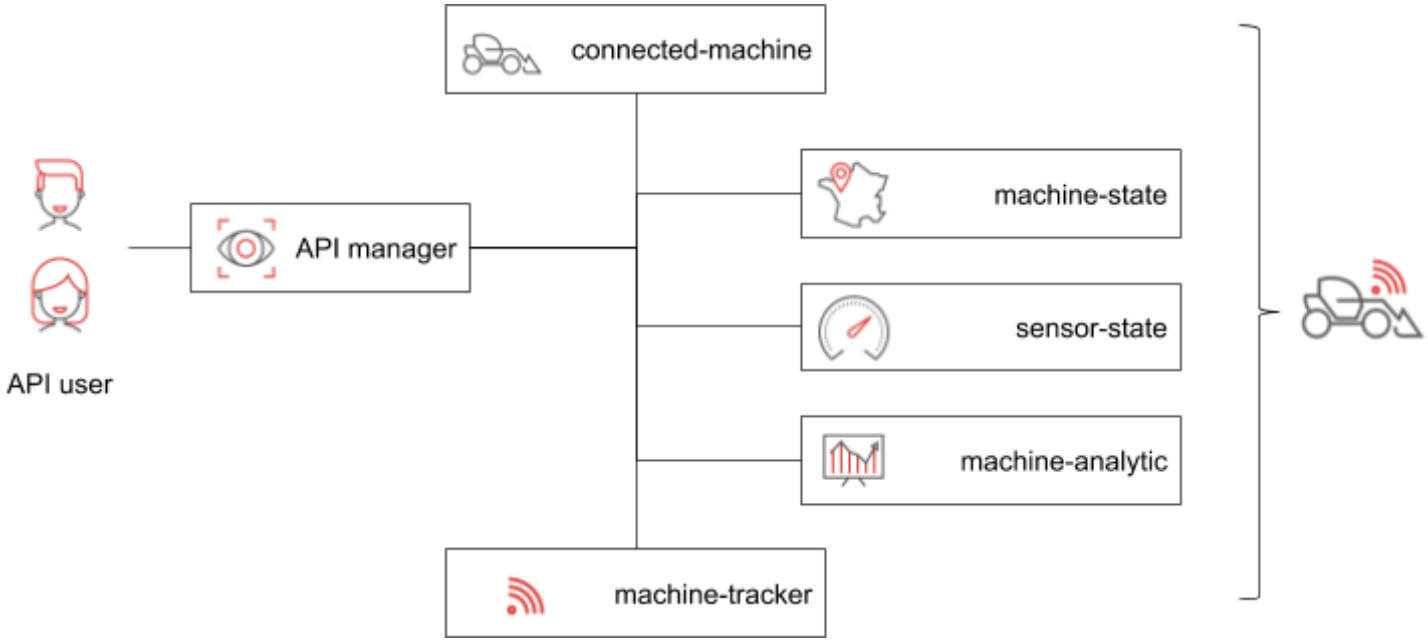
Each API serves a specific set of information, but shares common features of results paging, attribute filtering, records sorting. All these features are described with examples in this document, which is focused on one range of API products named "Connected Solutions".

Using this documentation comes as a second step after creating an account on the MANITOU API Developer Portal.
The user should already have received approval for a subscription to one of the "Connected Solutions" products, obtained keys and a secret user token, and understood the technicalities of the MANITOU API service, as described in the "MANITOU API - How to get onboard" companion document.

Understanding the Connected Solution service's design

Overview

The following schema describes the global design of MANITOU API's Connected Solutions service :



Security through API manager

It is necessary for the API user to **be authenticated** when performing a call to data APIs.

Please refer to the "MANITOU API - How to get onboard" documentation for more information on this process and a detailed technical explanation.

Static information with connected-machine API

This API holds **the list of all the machines that are part of the customer fleet.**

Among many static information (serial number, brand, model, etc), this API provides the customer with machine id's, each of which is unique to one machine and permanently affected to it.

The machine id information is the key that lets the customer retrieve dynamic information about the machines through the other data APIs.

This API also provides the user with the reference needed to retrieve information on the tracker the machine is equipped with, when necessary for the customer.

This API only needs to be called once in a while by the customer, each time a machine moves in or out of the fleet, to keep the machine id's catalog up to date.

Static information with machine-tracker API

Using the tracker reference provided by connected-machine, this API lets the user **retrieve information about the telematics device** used to retrieve machine data.

This API only needs to be called once in a while by the customer, each time a machine moves in or out of the fleet, to keep the customer's tracker references catalog up to date.

Dynamic machine state information with machine-state API

Using the machine id provided by connected-machine, this API lets the user **know where the machine is, how much time is on the hourmeter, etc**.

This API is designed to be called several times a day, for the user to keep track of the machine's movement and overall state.

Dynamic sensor information with sensor-state API

Using the machine id provided by connected-machine, this API lets the user **know the latest information of each exposed sensor of a given machine** (CAN-based information).

This API is designed to be called several times a day, for the user to keep track of the sensors' evolution.

Analytic information with machine-analytic API

Using the machine id provided by connected-machine, this API lets the user **access several analytic indicators**.
These indicators are calculated daily, on the basis of the machine sensors evolution, among other composite indicators that provide analytic insight on the machine's usage and performance.

This API is designed to be called once a day, for the user to keep track of the indicators progress over time.

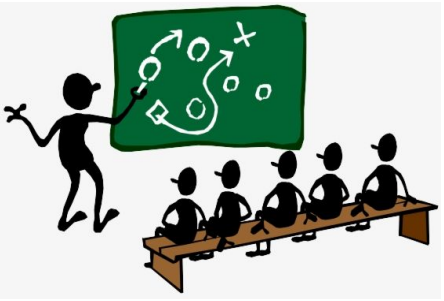
Best practices reminder

When pairing your IT system with the Connected Solutions products, remember to keep an eye on these 4 best practices, detailed in “MANITOU API - How to get onboard” :

Keep the page size low

Pinpoint what you want

Avoid hammering



Design a realistic data request strategy

In details : the connected-machine API

Data set example

The following chart lists the fields exposed by the API :

JSON data	Comments

<pre>{
"data" : [
{
"attributes" : {
"brand" : "Manitou",
"model" : "MRT 123",
"description" : "MRT 123 Forklift",
"serial-number" : "MAN00000Z000000123",
"height" : "312",
"width" : "288",
"length" : "513",
"weight" : "2140",
"build-year" : "2019"
},
"relationships" : {
"tracker" : {
"links" : {
"self" :
"/connected-machine/abcd1234-ab12-34cd-ab12-abcdef123456/relationships/tracker",
"related" :
"/connected-machine/abcd1234-ab12-34cd-ab12-abcdef123456/tracker"
}
}
},
"type" : "connected-machine",
"id" : "abcd1234-ab12-34cd-ab12-abcdef123456"
},
	<pre>Record list start indicator
	Record start indicator
	
	Machine brand
	Machine model
	Machine description
	Machine serial number
	Machine overall height in cm
	Machine overall width in cm
	Machine overall length in cm
	Machine gross weight in kg
	Machine build year
	
	Machine-dependent objects list
	Machine-dependent tracker info start
	
	Resource type
	Resource id (connected-machine id)</pre>

API parameters

This API allows the use of the following parameters :

Parameter name	Type	Manda-tory	Usage
`Ocp-Apim-Subscription-Key`	Header	yes	Customer subscription key (primary or secondary)
`X-token`	Header	yes	Customer secret user token. Provide the API with the secret user token that lets the customer retrieve its resources.
`api-version`	Header	yes	Version number of the API (v1, v2, etc.)

`page[size]` `page[number]`	Query	no	The API may return many resources records when called. The page[size] allows the user to define the number of records wanted per page, and the page[number] allows the user to jump to the desired result page. Examples : • `page[size]=30` ⇒ the API will return a maximum of 30 records per response • `page[number]=3` ⇒ the API will return page number 3 of all available results page.

`filter[brand]`
`filter[model]`
`filter[description]`
`filter[serial-number]`
	Query	no	The API will filter the result so that only machines matching the given attribute's value are retrieved
			Examples :
			• `filter[brand]=GEHL` ⇒ get only GEHL machines
			• `filter[model]=MT 625 H COMFORT 75K ST5 S1` ⇒ get all machines matching this model
			• `filter[serial-number]=MAN00000A0000000` ⇒ get only the machine matching this serial number

`filter[height]`
`filter[width]`
`filter[length]`
`filter[weight]`
`filter[build-year]`
	Query	no	The API will filter the result so that only machines matching the given attribute's value or provided value comparison are retrieved
			Examples :
			• `filter[height]=lt:2.5` ⇒ get machines under 2.5 meters in height
			• `filter[weight]=ge:3000` ⇒ get machines over or equal 3000 kg in weight
			• `filter[build-year]=ge:2015,le:2018` ⇒ get machines built from 2015 to 2018 (including boundaries)

`sort`	Query	no	The API will order the result records list according to ascending or descending value of the given attribute name. Using a dash ("-") before the attribute name indicates the API to perform the sort in a descending way. Elseway the sort is ascending. Examples : • `sort=height` ⇒ the API will return every recorded machine for the user, ordered from lowest height attribute value to highest • `sort=-weight` ⇒ the API will return every recorded machine for the user, ordered from highest weight attribute value to lowest

Please note that the filter parameters can be combined one with another, so you can retrieve all the machines of a given brand and for a given build year range, for example.

NOTE : as attribute values can differ from numerals to character strings, the attribute name provided to perform the sort or filter must be chosen wisely.
When performing comparisons on a character string attribute, the user must understand that even if the attribute value is expressed as a number (“1234”), it is not considered a number data type, so sorting and comparing is made through lexicographic comparison.

In details : the machine-state API

Data set example

The following chart lists the fields exposed by the API :

JSON data	Comments
{	
"data" : [	Record list start indicator
{	Record start indicator
"attributes" : {	
"latitude" : 56.88721000,	latitude
"longitude" : -111.36433500,	longitude
"altitude" : 0,	Altitude in m
"street-address" : "Unnamed Road",	Approximate street address
"city" : "Division No. 16",	Approximate city
"zip-code" : "T0P",	Approximate zip code
"country" : "CA",	Approximate country code (2 letters)
"engine-status" : 0,	Engine status (0 = off, 1 = on)
"key-status" : 1,	Ignition key status (0 = off, 1 = on)
"odometer" : 124.20,	Odometer in km
"cumulative-operation-hours" : 156,	Tracking device's engine on hours
"cumulative-idle-hours" : 148,	Tracking device's ignition on hours
"input3" : "0",	Digital input 3 value (0 = off, 1 = on)
"input4" : "0",	Digital input 4 value (0 = off, 1 = on)
"input5" : "0",	Digital input 5 value (0 = off, 1 = on)
"input6" : "0",	Digital input 6 value (0 = off, 1 = on)
"input7" : "0",	Digital input 7 value (0 = off, 1 = on)
"input8" : "0",	Digital input 8 value (0 = off, 1 = on)
"input9" : "0",	Digital input 9 value (0 = off, 1 = on)
"input10" : "0",	Digital input 10 value (0 = off, 1 =on)
"gps-fix-time" : "2019-12-08T17:40:50",	Gps positioning timestamp
"message-time" : "2019-12-08T17:40:50",	Machine state timestamp
"battery-voltage" : 14.00	Machine battery voltage in V
},	
"relationships" : {	Machine-dependent objects list
"connected-machine" : {	Machine-dependent connected-machine info start
"links" : {	
"self" : "/machine-state/9e6c5a77-588c-4192-a01b-00cc07baa72a/relationships/connected-machine",	
"related" : "/machine-state/9e6c5a77-588c-4192-a01b-00cc07baa72a/connected-machine"	
},	
"data" : {	
"type" : "connected-machine",	Connected resource type
"id" : "abcd1234-ab12-34cd-ab12-abcdef123456"	Connected resource id **(See parameter "include" below)**
}	
}	
},	
"type" : "machine-state",	Resource type
"id" : "9e6c5a77-588c-4192-a01b-00cc07baa72a"	Resource id (machine-state id)
},	

API parameters

This API allows the use of the following parameters :

Parameter name	Type	Manda-tory	Usage
`Ocp-Apim-Subscription-Key`	Header	yes	Customer subscription key (primary or secondary)
`X-token`	Header	yes	Customer secret user token. Provide the API with the secret user token that lets the customer retrieve its resources.
`api-version`	Header	yes	Version number of the API (v1, v2, etc.)
`include=connected-machine`	Query	no	The API will return the id of the corresponding connected-machine in the "relationships/connected-machine/data" sub-schema.
`page[size]` `page[number]`	Query	no	The API may return many resources records when called. The page[size] allows the user to define the number of records wanted per page, and the page[number] allows the user to jump to the desired result page. Examples : • `page[size]=30` ⇒ the API will return a maximum of 30 records per response • `page[number]=3` ⇒ the API will return page number 3 of all available results page.
`filter[attribute name]`	Query	no	Similarly to the filtering feature of connected-machine API, this API will filter the result so that only machine state records matching the given attribute's value are retrieved Examples : • `filter[message-time]=ge:2019-12-01T10:00:00,le:2019-12-01T11:00:00` ⇒ the API will return every recorded machine state for the user, comprised between 10am and 11am for Dec 1st 2019

			• `filter[connected-machine.id]=abcd1234-ab12-34cd-ab12-abcdef123456` ⇒ the API will return every available machine-state record for the given machine id (as retrieved from connected-machine API) • `filter[battery-voltage]=lt:10.5` ⇒ the API will return every available machine-state record indicating a machine battery voltage under 10.5 volts
`sort`	Query	no	The API will order the result records list according to ascending or descending value of the given attribute name. Using a dash (“-”) before the attribute name indicates the API to perform the sort in a descending way. Elseway the sort is ascending. Examples : • `sort=battery-voltage` ⇒ the API will return every recorded machine state for the user, ordered from lowest battery voltage attribute value to highest • `sort=-odometer` ⇒ the API will return every recorded machine state for the user, ordered from highest odometer attribute value to lowest

The filter parameters can be combined one with another, so you can retrieve all the machine-state records obtained today that show a cumulative-operation-hours over 3000 and a battery voltage under 10 volts, for example.

NOTE : as attribute values can differ from numerals to character strings, the attribute name provided to perform the sort or filter must be chosen wisely.
When performing comparisons on a character string attribute, the user must understand that even if the attribute value is expressed as a number (“1234”), it is not considered a number data type, so sorting and comparing is made through lexicographic comparison.

In details : the sensor-state API

Data set example

The following chart lists the fields exposed by the API :

JSON data	Comments
{	
"data" : [	Record list start indicator
{	Record start indicator
"attributes" : {	
"sensor-value" : "6.86",	Sensor value (expressed in sensor unit)
"tracker-Sensor-Id" : 10007,	Sensor id
"sensor-name" : "Engine Percent Load At Current Speed",	Sensor name
"sensor-unit" : "%",	Sensor unit
"timestamp" : "2019-12-08T23:15:02"	Sensor record timestamp
},	
"relationships" : {	Sensor-state dependent objects list
"equipment" : {	Sensor-state dependent equipment info start
"links" : {	
"self" : "/sensor-state/ca058771-0d50-4392-a14f-7d19303896fe/relationships/equipment",	
"related" : "/sensor-state/ca058771-0d50-4392-a14f-7d19303896fe/equipment"	
},	
"data" : {	
"type" : "connected-machine",	Dependant resource type
"id" : "abcd1234-ab12-34cd-ab12-abcdef123456"	Dependent resource id
}	**(See parameter "include" below)**
}	
},	
"type" : "sensor-state",	Resource type
"id" : "ca058771-0d50-4392-a14f-7d19303896fe"	Resouce id (machine-state id)
},	
[...]	...other records of sensor-state…
"links" : {	Result pages link records start
"next" : "/sensor-state?page[size]=10&page[number]=2&filter[id]=abcd1234-ab12-34cd-ab12-abcdef123456",	Next result page link
"last" : "/sensor-state?page[size]=10&page[number]=652&filter[id]=abcd1234-ab12-34cd-ab12-abcdef123456"	Last result page link
},	
"meta" : {	Total number of available records matching the request
"total-records" : 6513	
}	
}	

API parameters

This API allows the use of the following parameters :

Parameter name	Type	Manda-tory	Usage
`Ocp-Apim-Subscription-Key`	Header	yes	Customer subscription key (primary or secondary)
`X-token`	Header	yes	Customer secret user token. Provide the API with the secret user token that lets the customer retrieve its resources.
`api-version`	Header	yes	Version number of the API (v1, v2, etc.)
`include=equipment`	Query	no	The API will return the id of the corresponding equipment (connected-machine) in the "relationships/equipment/data" sub-schema.
`page[size]` `page[number]`	Query	no	The API may return many resources records when called. The page[size] allows the user to define the number of records wanted per page, and the page[number] allows the user to jump to the desired result page. Examples : • `page[size]=30` ⇒ the API will return a maximum of 30 records per response • `page[number]=3` ⇒ the API will return page number 3 of all available results page.
`filter[attribute name]`	Query	no	Similarly to the filtering feature of connected-machine API, this API will filter the result so that only sensor state records matching the given attribute's value are retrieved Examples : • `filter[tracker-Sensor-id]=10001` ⇒ the API will return every recorded sensor state for the user, for sensor id 10001 (see list in the appendix) • `filter[timestamp]=ge:2019-12-01T10:00:00,le:2019-12-01T11:00:00` ⇒ the API will return every recorded sensor state for the user, comprised between 10am and 11am for Dec 1st 2019 • `filter[equipment.id]=abcd1234-ab12-34cd-ab12-abcdef123456` ⇒ the API will return every available sensor-state record for the given equipment id (as retrieved from connected-machine API, provided the equipment related to the sensor is of the connected-machine type)

			• `filter[sensor-value]=lt:10` ⇒ the API will return every available machine-state record indicating a sensor-value under 10
`sort`	Query	no	The API will order the result records list according to ascending or descending value of the given attribute name. Using a dash (“-”) before the attribute name indicates the API to perform the sort in a descending way. Elseway the sort is ascending. Examples : • `sort=sensor-value` ⇒ the API will return every recorded sensor state for the user, ordered from lowest sensor-value attribute value to highest • `sort=-sensor-value` ⇒ the API will return every recorded sensor state for the user, ordered from highest sensor-value attribute value to lowest

NOTE : as attribute values can differ from numerals to character strings, the attribute name provided to perform the sort or filter must be chosen wisely.
When performing comparisons on a character string attribute, the user must understand that even if the attribute value is expressed as a number (“1234”), it is not considered a number data type, so sorting and comparing is made through lexicographic comparison.

In details : the machine-analytic API

Data set example

The following chart lists the fields exposed by the API :

JSON data	Comments
{	
"data" : [	Record list start indicator
{	Record start indicator
"attributes" : {	
"reference-number" : "MAN00000Z000000123",	Machine reference number (S/N)
"message-date" : "2019-11-12T00:00:00",	Calculation date
"variable-id" : 10003,	Sensor id
"var-name" : "Engine Total Hours of Operation",	Sensor name
"uom" : "hr",	Sensor unit of measure
"value-min" : 29.75,	Min sensor value of the day
"value-max" : 36.049999237060547,	Max sensor value of the day
"value-avg" : 33.072158813476562,	Average sensor value of the day
"value-sum" : 5820.7001953125,	Sum of available values of the day
"value-count" : 176.0	Nb of available values of the day
},	
"relationships" : {	Machine-analytic dependent objects list
"equipment" : {	Machine-analytic dependent equipment info start
"links" : {	
"self" : "/equipment-analytic/6ccae061-e613-4fc7-9da4-0139dc68777d/relationships/equipment",	
"related" : "/equipment-analytic/6ccae061-e613-4fc7-9da4-0139dc68777d/equipment"	
}	
}	
},	Resource type
"type" : "equipment-analytic",	Resouce id (equipment-analytic id)
"id" : "6ccae061-e613-4fc7-9da4-0139dc68777d"	
},	...other records of machine-analytic…
[...]	Result pages link records start
"links" : {	Last result page link
"last" : "/equipment-analytic?page[size]=10&page[number]=1&filter[reference-number]=MAN00000Z000000123&filter[message-date]=ge:2019-11-12&filter[message-date]=le:2019-11-12"	
},	Total number of available records matching the request
"meta" : {	
"total-records" : 2	
}	
}	

API parameters

This API allows the use of the following parameters :

Parameter name	Type	Manda-tory	Usage
`Ocp-Apim-Subscription-Key`	Header	yes	Customer subscription key (primary or secondary)
`X-token`	Header	yes	Customer secret user token. Provide the API with the secret user token that lets the customer retrieve its resources.
`api-version`	Header	yes	Version number of the API (v1, v2, etc.)
`page[size]` `page[number]`	Query	no	The API may return many resources records when called. The page[size] allows the user to define the number of records wanted per page, and the page[number] allows the user to jump to the desired result page. Examples : • `page[size]=30` ⇒ the API will return a maximum of 30 records per response • `page[number]=3` ⇒ the API will return page number 3 of all available results page.
`filter[`*`attribute name`*`]`	Query	no	Similarly to the filtering feature of connected-machine API, this API will filter the result so that only sensor state records matching the given attribute's value are retrieved Examples : • `filter[variable-id]=10001` ⇒ the API will return every recorded machine-analytic for the user, for sensor id 10001 (see list in the appendix) • `filter[message-date]=ge:2019-12-01T10:00:00,le:2019-12-01T11:00:00` ⇒ the API will return every recorded machine-analytic for the user, comprised between 10am and 11am for Dec 1st 2019 • `filter[reference-number]=MAN00000Z000000123` ⇒ the API will return every available machine-analytic record for the given machine serial number
`sort`	Query	no	The API will order the result records list according to ascending or descending value of the given

			attribute name. Using a dash (“-”) before the attribute name indicates the API to perform the sort in a descending way. Elseway the sort is ascending. Examples : • `sort=value-max` ⇒ the API will return every recorded machine-analytic for the user, ordered from lowest value-max attribute value to highest • `sort=-value-count` ⇒ the API will return every recorded machine-analytic for the user, ordered from highest value-count attribute value to lowest

NOTE : as attribute values can differ from numerals to character strings, the attribute name provided to perform the sort or filter must be chosen wisely.
When performing comparisons on a character string attribute, the user must understand that even if the attribute value is expressed as a number (“1234”), it is not considered a number data type, so sorting and comparing is made through lexicographic comparison.

Appendix : list of sensor id's and description

This list details all availables sensor id's. Please note that not all sensors listed below do exist on every machine : depending on the machine range, model and options, some may not be available.
Please note also that this list may be updated as new machines and options are made available.

Sensor id	Description
101	Temperature1
102	Temperature2
498	Engine Starter Mode
512	Ambient Air Temperature
884	Transmission Oil Temperature 1
1296	attachment recognition
1300	Boom movement cut off
1301	Current engine status
1302	Current ignition status
1303	Attachment Confirmed
1304	Fuel less than 10%
1305	Command position
1306	Machine Type (PLUS 2150 = 1;Plus 2550 = 2; etc)
1307	Engine type (Mercedes-Benz = 1; Perkins = 2; etc)
1308	transmission type (Sauer = 1; Rexroth = 2;etc)
1310	Diesel Particulate Filter Status
1311	SPN Error code from CPC4 (Mercedes-Benza Master ECU)
1312	FMI Error code from CPC4 (Mercedes-Benza Master ECU)
1313	Distributor errors
1314	Transmission errors

1315	Manitou Error codes
1316	Manitou Warning message
1317	Actual load
1319	Radius
1320	Height
1321	turret position
1322	Angle
1323	Lmi percentage
1324	SPN Error code from MCM
1325	FMI Error code from MCM
1326	SPN Error code from ACM
1327	FMI Error code from ACM
1471	Transmission Oil Pressure
1475	Total Vehicle Distance
1476	Engine Intake Air Temperature
1770	Auxiliary I/O #03
1771	Auxiliary I/O #02
1783	Ambient Air Temperature
1792	Engine Coolant Level 1
1793	Engine Air Filter 1 Differential Pressure
1794	Aftertreatment Diesel Particulate Filter Active Regeneration Status
2030	Telescop In status
2031	Cab/ Platform /RC mode status
2032	Fork / Bucket / Suspended load mode status
2033	DEF tank level below 10%
2034	Direction engaged

2035	Strain gauge (Max/min)
2036	Strain gauge (Average)
2077	Engine Oil Temperature 1
2084	Diesel Particulate Filter Active Regeneration Inhibited Due to Inhibit Switch
2085	Aftertreatment SCR Operator Inducement Severity
2086	Aftertreatment 1 Diesel Particulate Filter Soot Load Percent
2087	Aftertreatment 1 Diesel Particulate Filter Ash Load Percent
2106	STOP Lamp
2107	WARNING Lamp
2108	Servicing Lamp
2109	Active Error Code
2950	Active Diagnostic Trouble Codes
3660	Door opened while driving
3661	Travelling with boom angle high
4873	Driving without seatbelt
10001	Engine Coolant Temperature
10002	Engine Speed
10003	Engine Total Hours of Operation
10004	Maximum load
10005	Engine Total Fuel Used
10006	Engine Oil Pressure
10007	Engine Percent Load At Current Speed
10008	Aftertreatment 1 Diesel Exhaust Fluid Concentration
10009	Aftertreatment Diesel Particulate Filter Status
10010	Aftertreatment 1 Diesel Exhaust Fluid Tank Level

10011	Air Filter Clogging lamp
10012	Alternator Not Charging lamp
10013	Wheel-Based Vehicle Speed
10014	Coolant Temperature lamp
10015	Dpf lamp
10016	Engine Fuel Rate
10017	Engine Oil Pressure lamp
10018	Exhaust System High Temperature Lamp Command
10019	Fault Braking lamp
10020	Fuel Level
10021	Hydraulic Filter Clogging lamp
10022	Low Brake Fluid Level lamp
10023	Low Coolant Fluid Level lamp
10024	Outriggers on ground
10025	Override
10026	Scr lamp
10027	Seat
10028	Steering Default lamp
10029	Transmission Oil Pressure lamp
10030	Transmission Oil Temperature lamp
10031	Water In Fuel Indicator 1