Tracking Extension

Administrator Manual

Table of Contents

Welcome to the JMap Tracking Extension 1

Installing the Tracking Extension 2

Installing Tracking Listener 4

Configuring Tracking Listener 6

Tracking Database 14

Managing Mobile Units and Groups 15

Preparing Data Sources and Layers 17

Configuring the Layers Managed by Tracking 23

Configuring Road Networks for Analysis 27

Configuring Types of Analyses 30

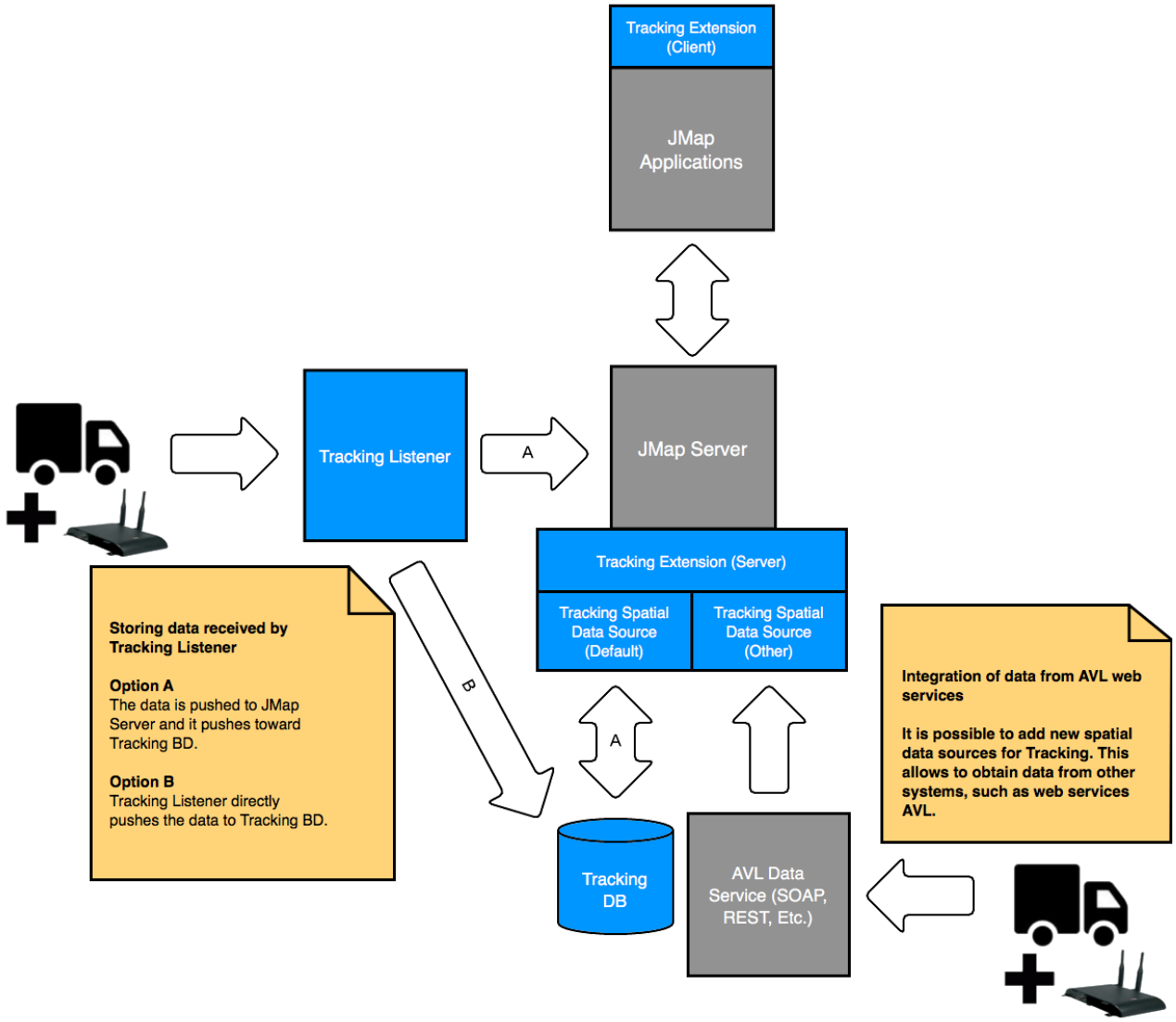
Configuring Geofencing 35

Managing Permissions 36

Welcome to the JMap Tracking Extension

This manual explains how to install, configure and manage the JMap 7.0 Tracking extension.

The following diagram provides an overview of Tracking's architecture.



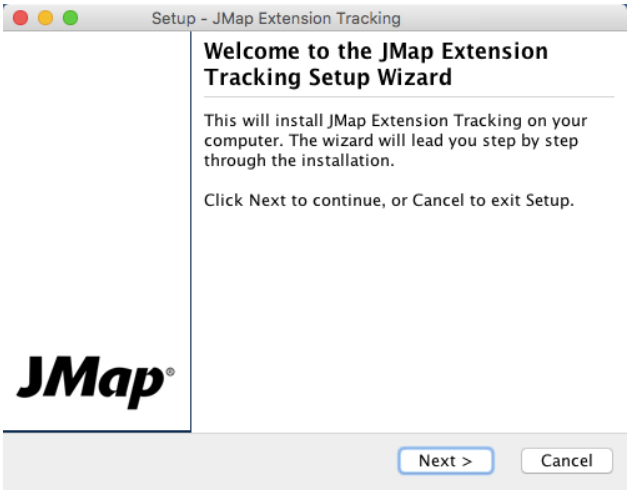
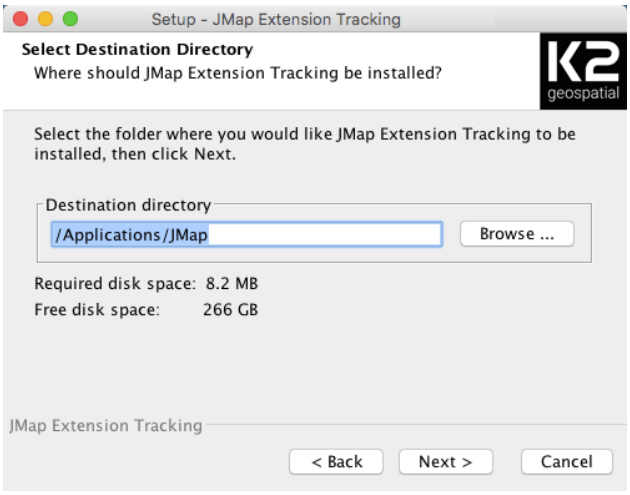
Installing the Tracking Extension

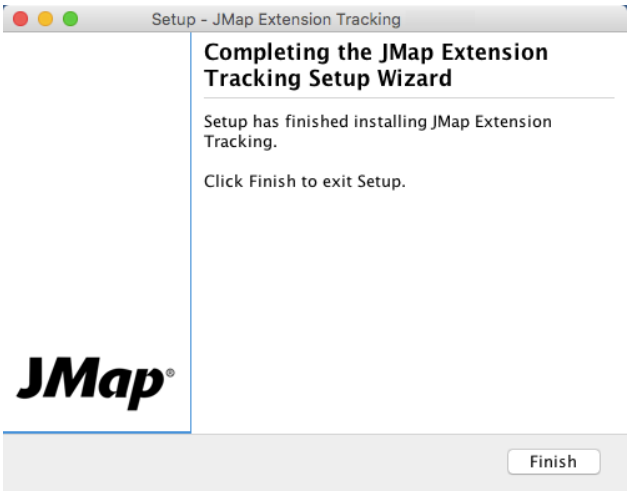
Tracking is installed by running the installation program for your platform.

Note: Tracking requires JMap 7.0 or a later version.

Step-by-step installation

To install Tracking, JMap Server must be shut down. If JMap Server is running, it will be automatically shut down.

Step	Instructions
	Click on **Next**.
	Select the installation directory for Tracking. Tracking must be installed with JMap 7.0 or a later version. By default, the installation program automatically identifies the JMap 7.0 folder. Click on **Browse** to select a different folder. Click on **Next**.



This window appears after the files are copied.

Click on **Finish** to complete the installation.

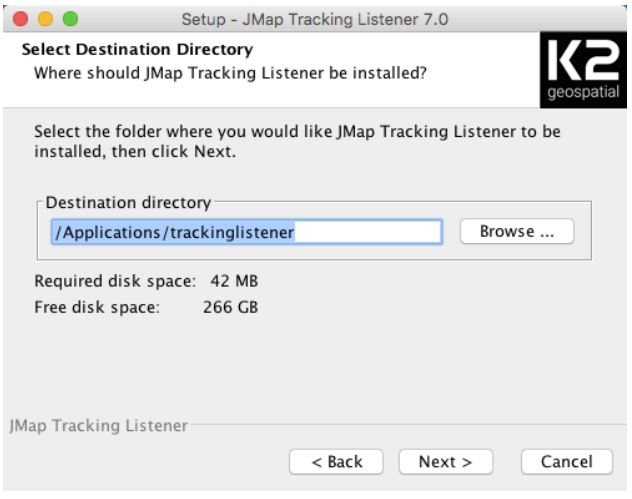
Installing Tracking Listener

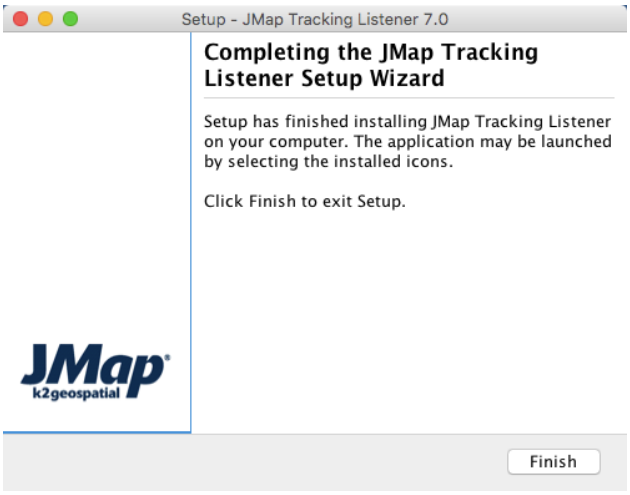
Tracking Listener is an application that receives the data destined to the Tracking extension. This data usually comes from tracking equipment installed in vehicles or on people. This can include GPS modems in vehicles, connected objects or applications on mobile devices.

The Tracking Listener application is fully independent of JMap Server and must be installed on a server that can be accessed through the network using equipment that transmits data.

Step-by-step installation

Tracking Listener requires Java SE 7 (JRE) or a later version.

Step	Instructions
	Click on **Next**.
	Select the installation folder for Tracking Listener. Click on **Browse** to select a different folder. Click on **Next**.



Click on **Finish** to complete the installation.

Configuring Tracking Listener

Tracking Listener does not have a graphical interface. It must be configured using the settings files located in the **trackinglistener/conf** folder. The settings are described below.

tracking.properties file

This is the main configuration file. Other configuration files may be required, depending on the options selected. Each parameter is made up of a **key=value** pair. The keys are unique. Lines starting with # are comments.

The various parameters are described below.

Receivers

```
#Receivers to load (one or many)
receiver.1=jmap.extensions.tracking.listener.receiver.TrackingTCPReceiver
receiver.2=jmap.extensions.tracking.listener.receiver.TrackingUDPReceiver
receiver.3=jmap.extensions.tracking.listener.receiver.FilesReceiver
#receiver.4=jmap.extensions.tracking.listener.receiver.Simulator
```

This section contains the configuration of the Tracking Listener's receivers. These modules are responsible for receiving the data using various methods. Several receivers can be used simultaneously. Each receiver has its own settings file.

The main receivers are TrackingTCPReceiver (receives data through TCP/IP, TCP protocol) and TrackingUDPReceiver (receives data through TCP/IP, UDP protocol). Depending on the configuration of the geolocation and TCP/IP transmission equipment, either one of these receivers can be used. Each one has its own settings file.

FileReceiver is less commonly used than the others; it may be used to load data from files.

Simulator is used to simulate data reception by Tracking Listener.

```
#Send alarm if no activity for the specified timeout (ms)
activity.alarm.enabled=true
activity.alarm.timeout=43200000
```

This section allows you to configure alarms that will be sent if no data is received for a given period of time. This can be useful to identify unusual activity (e.g. loss of network connection).

The **activity.alarm.enabled** parameter allows you to enable (true) or disable (false) sending alarms.

The **activity.alarm.timeout** parameter is used to define the inactivity period (in milliseconds) after which an alarm will be sent.

Cache

```
#Cache system to load (only one)
cache=jmap.extensions.tracking.listener.cache.FileCache
#cache=jmap.extensions.tracking.listener.cache.MemoryCache
```

This section contains the configuration of the cache system used by Tracking Listener. Tracking Listener uses this cache to store the data received before sending it to the persistence system. If the persistence system is not operating normally, the cache will store the data to prevent it from being lost. In the event data accumulates in the cache, Tracking Listener can send alarms. Refer to the section on alarms for more information.

There are 2 modules. The **FileCache** module stores data on the disk (trackinglistener/cache folder). Should Tracking Listener experience an interruption, the data on the disk will be reread and sent to the persistence system to minimize data loss. If data accumulates in the cache, disk space usage will increase, which is generally not a problem if this occurs for a short period of time. The cache module is recommended. The settings file for this module does not contain any settings.

The **MemoryCache** module stores in-memory data. If Tracking Listener is restarted, cached data will be lost. In addition, if cached data accumulates, the memory can become saturated quickly. When possible, it is preferable to use the **FileCache** module. This module's settings file does not contain any settings.

```
#Send an alarm if cache size becomes higher than specified value.
#Resend alarm if size grows by the specified factor
cache.size.alarm.enabled=true
cache.size.alarm.value=100
cache.size.alarm.value.factor=2
```

This section explains how alarms are sent when data accumulates in the cache.

The **cache.size.alarm.enabled** setting is used to enable (true) or disable (false) sending alarms.

The **cache.size.alarm.value** setting determines the size of cached data required to send an alarm. When the cache reaches this limit, an alarm is sent.

The **cache.size.alarm.value.factor** parameter determines the frequency of subsequent alarms sent until the situation returns to normal. For example, if the value of this factor is 2, a new alarm will be sent each time the size of cached data is doubled. For a value of 100 and a factor of 2, alarms will be sent when the cached data reaches 100, 200, 400, 800, and so forth. This function is used to limit the number of alarms sent.

Decoders

```
#Decoders to load (one or many)
decoder.1=jmap.extensions.tracking.listener.decoder.CypressDecoder
#decoder.2=jmap.extensions.tracking.listener.decoder.BluetreeDecoder
#decoder.3=jmap.extensions.tracking.listener.decoder.CypressDickeyJoh
nDecoder
```

This section describes the modules used to decode messages coming from geolocation equipment. Each decoder is designed to decode the messages of a specific type of equipment. Several decoders can be used simultaneously. New decoders can be developed to support additional equipment types.

CypressDecoder is the most commonly used decoder with Tracking. It is used to decode messages sent by Cypress' Chameleon CTM equipment.

CypressDisckeyJohnDecoder is used to decode messages originating from Cypress equipment connected to Dickey-John application control systems. Using this module, winter maintenance data can be viewed and analyzed with Tracking.

The existing decoders do not have any settings files.

Persistence Handlers

```
#Persistence handler to load (only one)
handler=jmap.extensions.tracking.listener.persistence.JDBCHandler
#handler=jmap.extensions.tracking.listener.persistence.JMapServerHand
ler
#handler=jmap.extensions.tracking.listener.persistence.Simulator
```

This section describes the persistence management modules. Persistence modules are used to store the information Tracking Listener receives. Only one persistence management module can be configured at a time.

The **JDBCHandler** module persists the data directly in a relational database (Tracking database). You can use this module if your network architecture allows Tracking Listener to connect to the Tracking database directly. This module has its own settings file.

The **JMapServerHandler** module delegates data persistence to JMap Server. All the data received will be sent to JMap Server, which handles persistence in the Tracking database. This module has its own settings file.

The **Simulator** module is used for test purposes only and does not handle any persistence.

```
#Persistence thread pool size
threadpool.size=2
```

This setting is used to define how many threads will be used for persistence. This can be used to execute simultaneous persistence tasks when a large volume of data is received. The default value of 2 is appropriate in most situations.

Filters

```
#Filters (zero, one or many)
#filter.0=jmap.extensions.tracking.listener.filter.TrackingDataFilter
SpeedValidity
```

This section contains the configuration of the filter modules. Filters are optional and are used to filter the data received from geolocation devices.

The only filter available is **TrackingDataFilterSpeedValidity**. It allows you to ignore data received if the speed indicated is 0 or lower. Other filters can be added as needed.

SMTP

```
#SMTP parameters for sending alarms using email
smtp.host=mail.k2geospatial.com
smtp.username=someuser
```

```
smtp.password=apassword
smtp.from=tracking@k2geospatial.com
smtp.recipients=manager@k2geospatial.com
```

This section allows you to define settings to send alarms by email. If no settings are defined, no email can be sent.

The **smtp.recipients** parameter is used to define the email addresses of the people who will receive the alarms. Separate the addresses using the " ; " character.

Logs

```
log.level=info
```

Tracking Listener has a log file. This parameter defines the level of information to record in the log. The log files are located in **trackinglistener/logs**.

The available levels are: **all**, **debug**, **info**, **error**, and **fatal**.

Settings files of the modules

jmap.extensions.tracking.listener.receiver.TrackingTCPReceiver.properties file

```
port=3636
```

This file contains a single parameter that defines the port to be used by the **TrackingTCPReceiver** module in order to receive data via the TCP protocol. The port can be different, depending on your needs.

jmap.extensions.tracking.listener.receiver.TrackingUDPReceiver.properties file

```
port=3637
```

This file contains a single parameter that defines the port to be used by the **TrackingUDPReceiver** module in order to receive data via the UDP protocol. The port can be different, depending on your needs.

jmap.extensions.tracking.listener.receiver.Simulator.properties file

```
delay.min=3000
delay.max=10000

#MOBILEID=X,Y,INPUT,DIRECTION|...
Mobile_1=-73.56769676273318,45.494208187827,0,0|-
73.56485608621003,45.492955408051074,1,0|-
73.59709811452376,45.47284507389789,1,0
Mobile_2=-73.66769676273318,45.474208187827,0,0|-
73.46485608621003,45.292955408051074,1,0|-
73.49709811452376,45.67284507389789,0,0
```

This file contains the parameters of the **Simulator** module. This module is used to simulate data reception by Tracking Listener. The simulator reads the data of this file and simulates data reception, with a variable frequency.

The **delay.min** and **delay.max** parameters determine the frequency (in milliseconds) of the data simulated by the simulator. The delay is random and always falls between the minimum delay and maximum delay.

The following lines are used to set the list of mobiles simulated as well as their simulated behaviour. Each line starts with the name of the mobile (the key) and contains series of 4 values separated by " | " characters. Each series contains the x and y coordinates, the value of the entries and the mobile's direction. Once the end of the line is reached for a mobile, the simulator restarts from the beginning.

jmap.extensions.tracking.listener.persistence.JDBCHandler.properties file

```
# Parameters for JDBC persistence
jdbc.driver=org.gjt.mm.mysql.Driver
jdbc.url=jdbc:mysql://localhost:3306/tracking?useCursorFetch=true
jdbc.username=root
jdbc.password=
jdbc.poolsize=1
```

```

```

This file contains the settings of the **JDBCHandler** module. These parameters are used to establish a JDBC connection with a relational database system.

The **jdbc.driver** parameter indicates which class of the JDBC pilot must be used. This class must be in the Java classpath.

The **jdbc.url** parameter is the database connection string. It varies based on the chosen pilot.

The **jdbc.username** and **jdbc.password** parameters contain the database authentication information.

You can activate the password encryption using the password(mot_de_passe) function.

Example: jdbc.password=password(Bonjour!123)

The password written in clear text will automatically be replaced by the password encrypted in the file.

jmap.extensions.tracking.listener.persistence.JMapServerHandler.properties file

```
poolid=99

# Parameters for JMapServer
# connectionmode : CONNECTION_DIRECT or CONNECTION_PROXY

jmapserver.host=localhost
jmapserver.port=7003
jmapserver.httpport=8080
jmapserver.proxypath=/jmap/servlet/jmapproxy
jmapserver.serverid=jmap
jmapserver.connectionmode=CONNECTION_DIRECT
jmapserver.username=administrator
jmapserver.password=
```

This file contains the parameters of the **JMapServerHandler** module. These parameters are used to establish a connection with JMap Server.

The **jmapserver.host** and **jmapserver.port** parameters define the options for a direct connection (if jmapserver.connectionmode=CONNECTION_DIRECT).

The **jmapserver.httpport**, **jmapserver.proxypath** and **jmapserver.serverid** parameters define the options for a proxy connection (if jmapserver.connectionmode=CONNECTION_PROXY).

The **jmapserver.username** and **jmapserver.password** parameters contain the JMap Server authentication information.

You can enable the password encryption using the password(mot_de_passe) function.

Example: jmapserver.password=password(Bonjour!123)

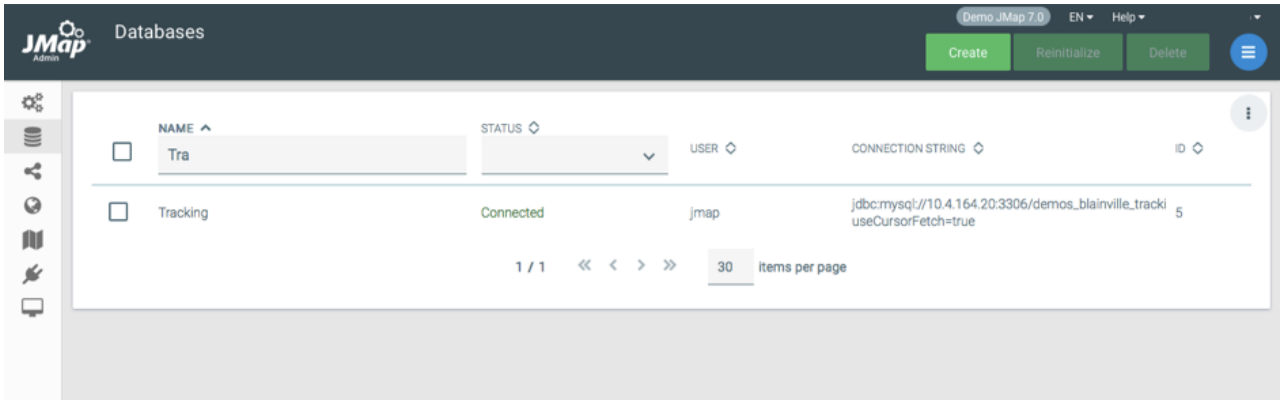
The password written in clear text will automatically be replaced by the password encrypted in the file.

Tracking Database

Tracking requires a database to function. The database is not automatically created. It must be created manually using the **create_tables.sql** script included in the trackinglistener/conf directory. The script contains the SQL queries required to create the database in one of the DBMS supported by tracking (Oracle, MS SQLServer and MySQL).

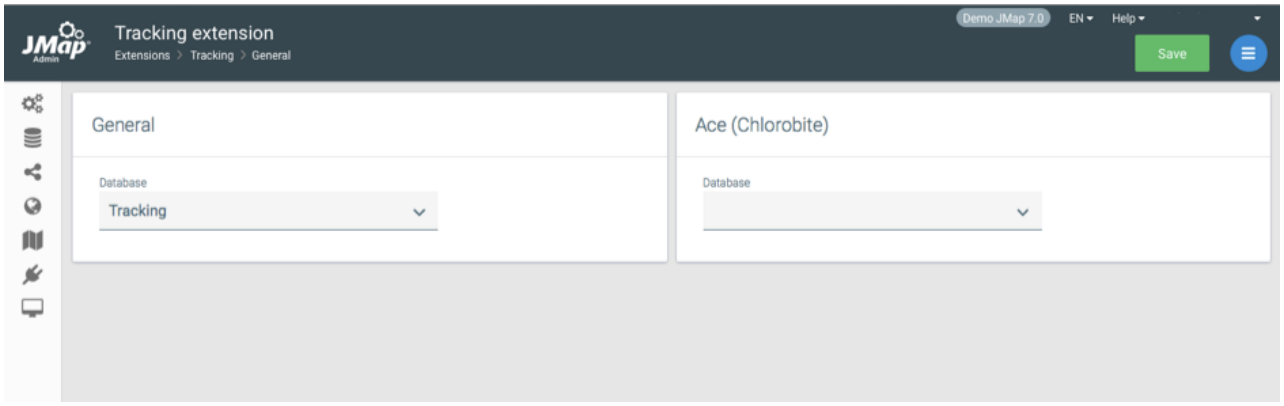
You must create a user account that will be used by Tracking to connect to the database. This account must have read and write permissions in the database.

Once the database is created, you must configure the connection to the database in JMap Admin using the account destined to this purpose. The name of the database in JMap Admin is not important.



Example of Tracking database in JMap Admin

Afterwards, you must indicate to Tracking which database the extension must use (i.e. the database created at the previous step). This parameter can be modified in the **General** section of Tracking's configuration interface.

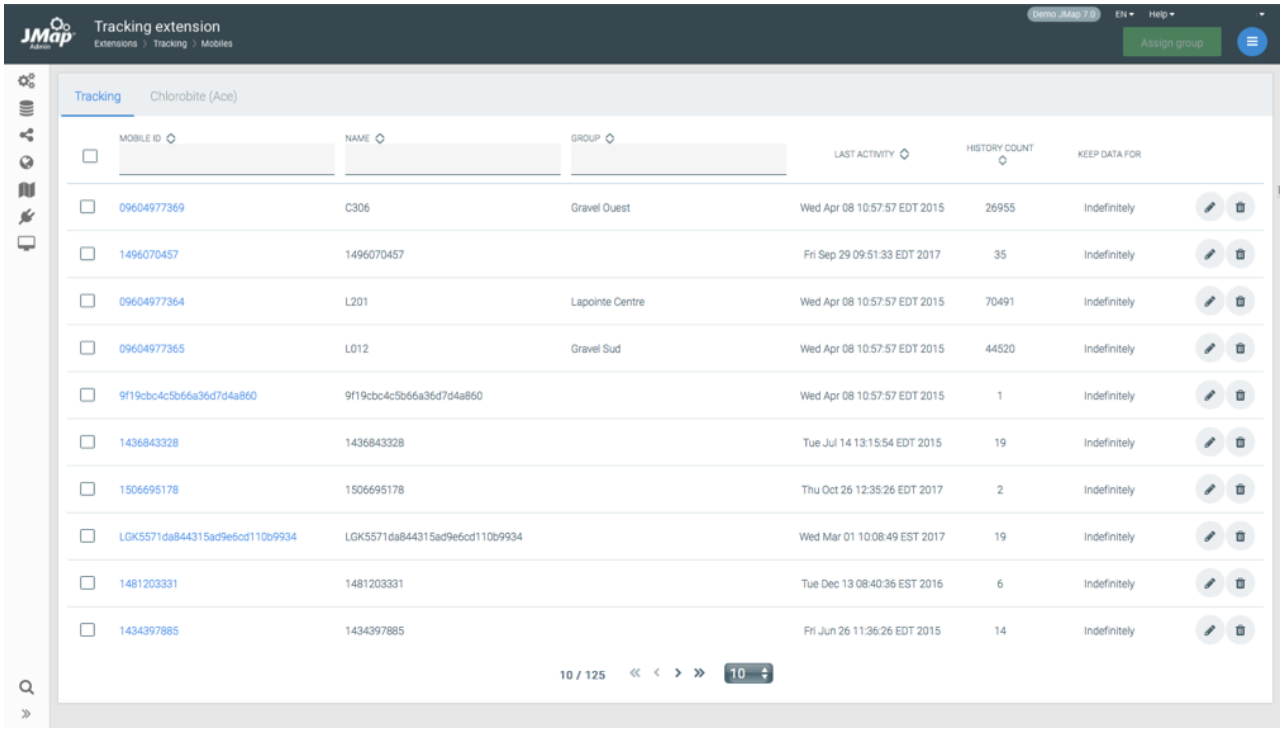


Selecting the Tracking database

Managing Mobile Units and Groups

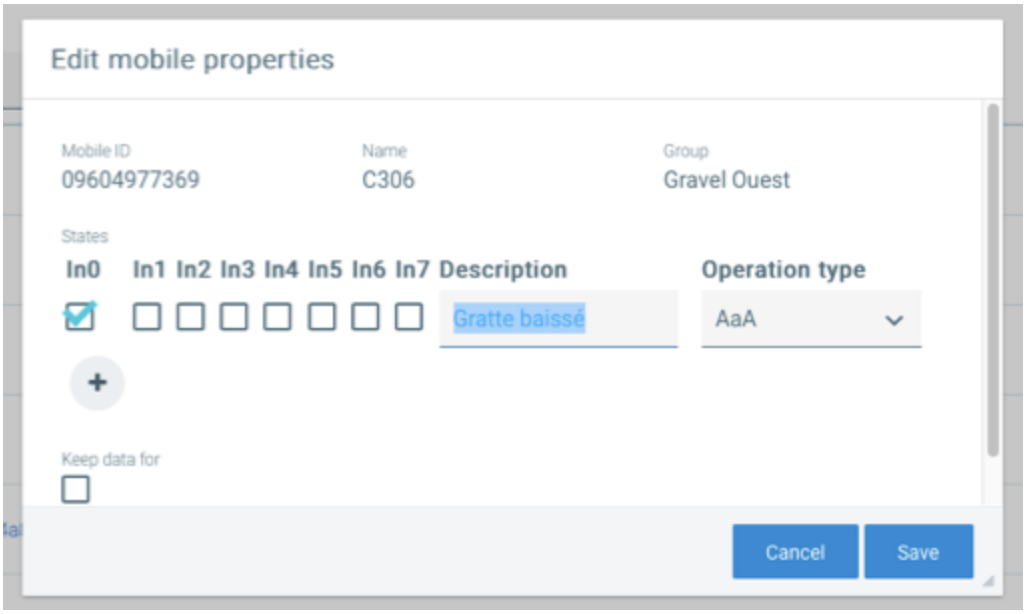
Tracking offers a section for basic management of mobile units. You can access this interface by opening the **Mobile management** section in the configuration of the Tracking extension.

Among other things, you can assign names to mobile units, assign them a group and decide how long the data will be kept. If your Tracking database does not contain any data yet, skip this step and come back to it when data will be available.



List of mobile units

The list displays all mobile units existing in the Tracking database. If any mobile unit data comes from external systems (e.g. AVL Web data service), it will not be stored in the Tracking database and cannot be configured in this interface. To modify the configuration of a mobile unit, click on the icon of the corresponding row.



A mobile unit configuration interface

Edit mobile properties	
Mobile ID	Unique identifier of the mobile unit provided by the tracking system. Cannot be modified.
Name	The name you can assign to the mobile unit. This is the name users will see in JMap applications. This is a required setting, and its default value is the same as the mobile's ID.
Group	You can assign the mobile unit to a group. Afterwards, you can use groups to separate the mobile units into several layers. Refer to the Preparing Data Sources and Layers section for more information. This parameter is optional.
States	You can define states and assign them to a type of operation. States reflect a combination of the I/O values from the tracking equipment. These states can be used to produce reports in Tracking or for special functions. This parameter is optional.
Keep data for	Enable this option if you wish to manage the period during which the data of a mobile unit will be stored in Tracking's database. The oldest data of this mobile unit will be automatically deleted after the period indicated.

Preparing Data Sources and Layers

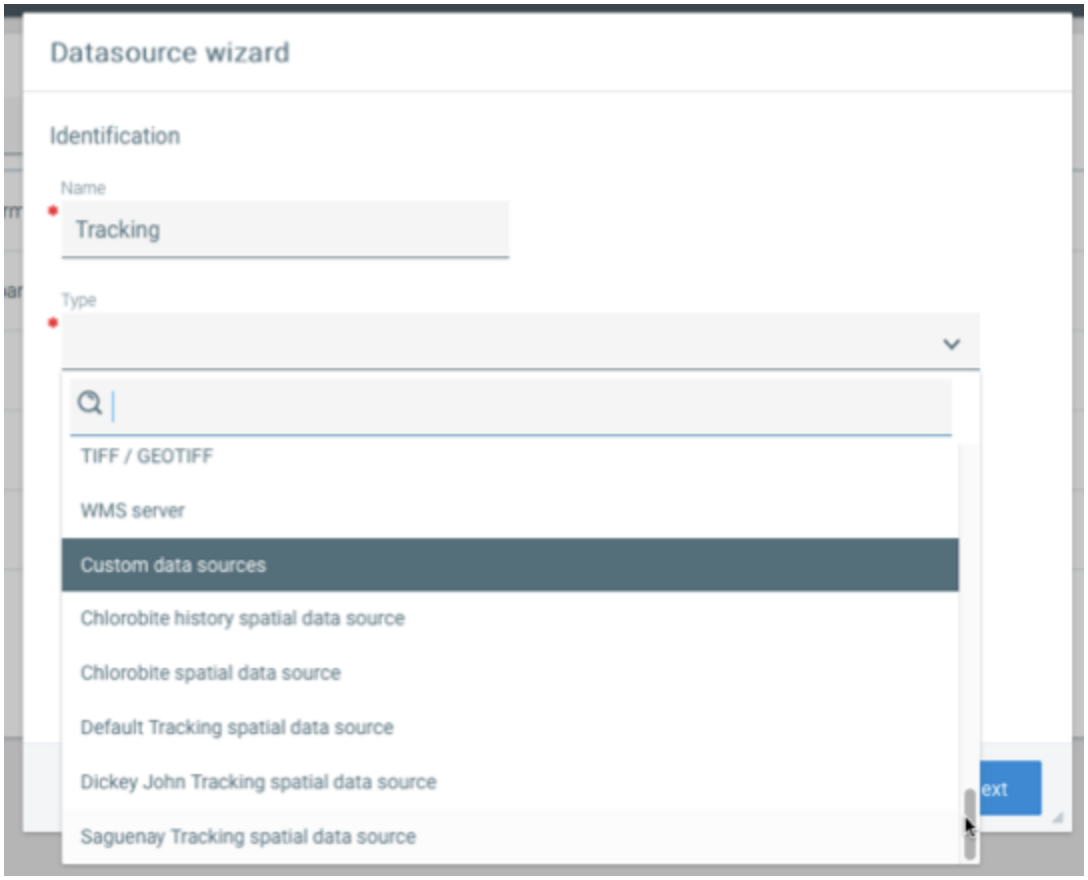
Before using Tracking, it is essential to create a spatial data source for this extension. Tracking uses data source types that have been developed specifically for it. The ones provided with Tracking read the data in the Tracking database, but additional Tracking data sources can be developed to obtain data differently, for example, by querying a Web service via the Internet.

Once you have prepared the Tracking data source as well as layers for Tracking in one or more projects, you will be ready to begin configuring the Tracking extension.

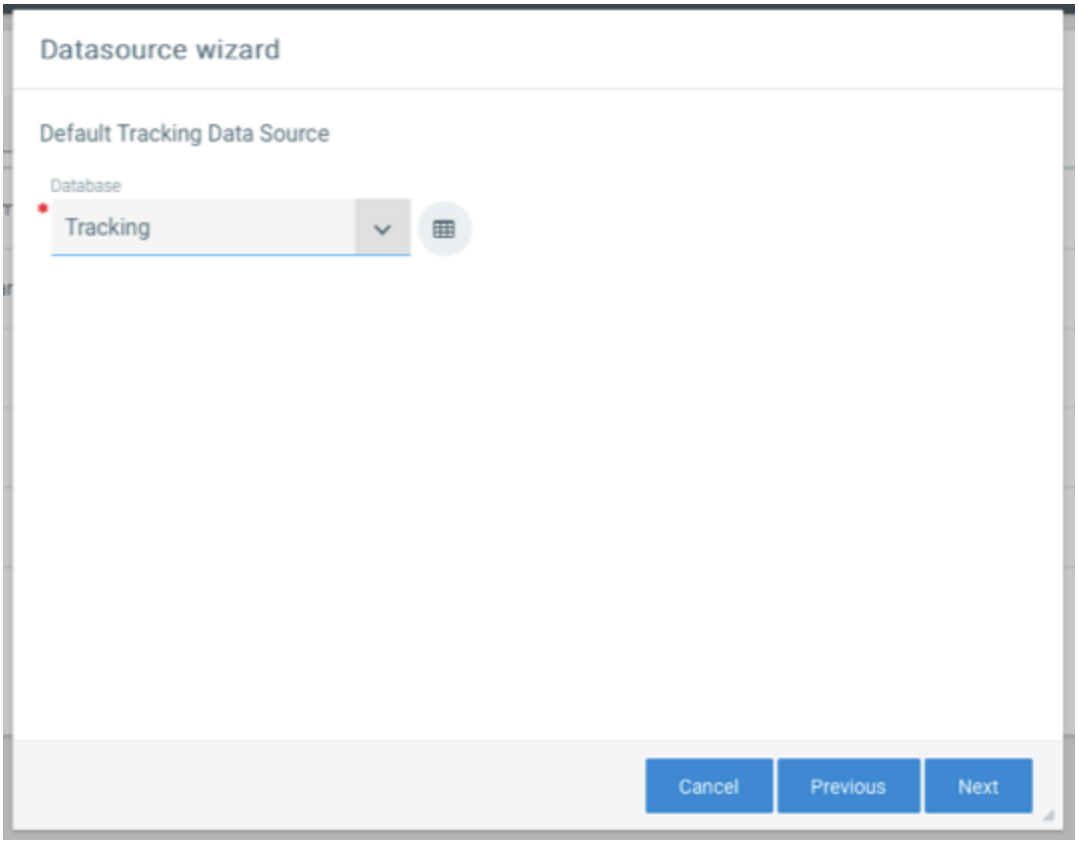
Creating a spatial data source for Tracking

Follow the steps below to create a spatial data source for Tracking.

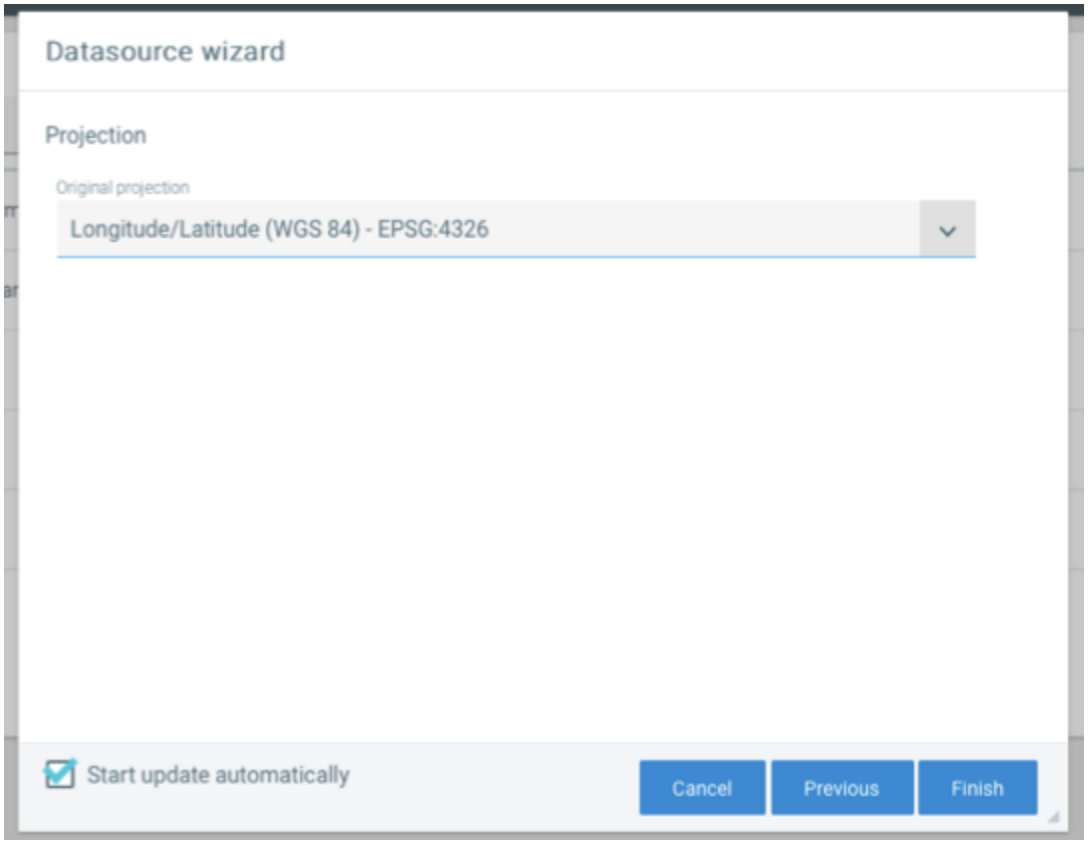
1. **Create** a spatial data source. The name you give the data source is not important.
2. At the following step, select the **Default Tracking spatial data** source type. Other types of data sources can exist for Tracking, such as one for **Dickey-John equipment** to manage application data for winter maintenance. Their configuration is similar.



3. On the next step, select the Tracking database in the list.



4. Afterwards, select the Longitude/Latitude (WGS 84) projection to indicate to JMap that GPS data is used. In some specific cases, the projection may be different.



5. Click on **Finish**.

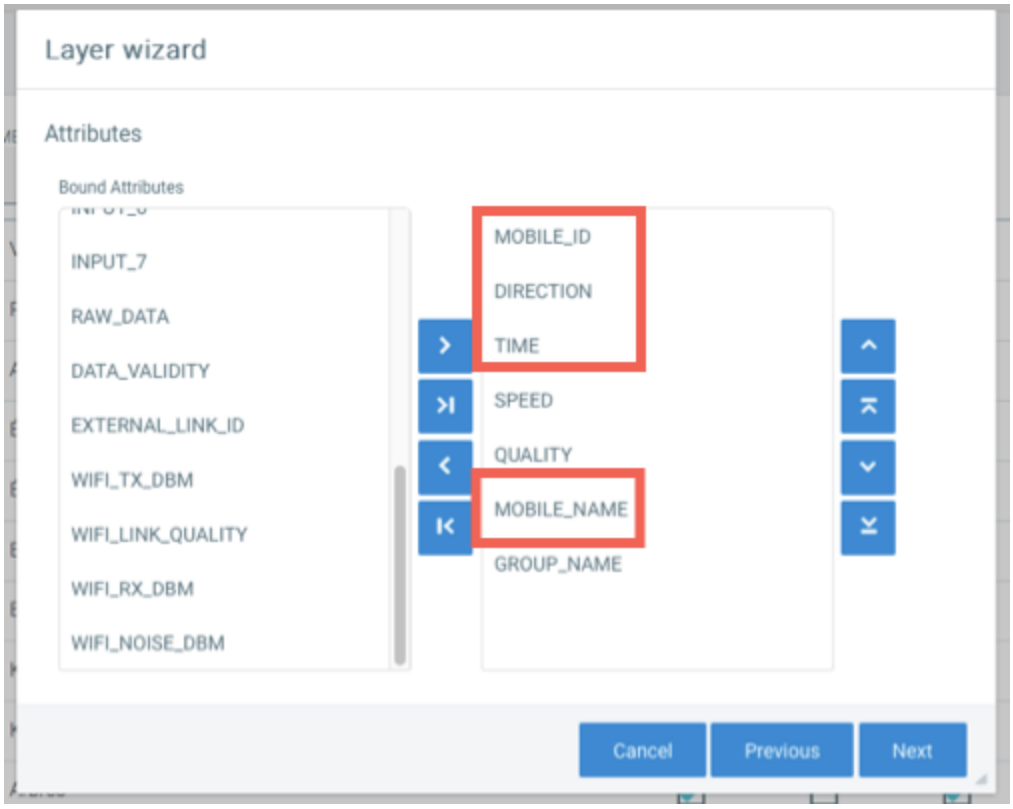
Creating layers for Tracking in a project

To view the mobile units managed by Tracking in a project, you must create at least 2 layers for Tracking: the mobile units layer and the history layer.

Each pair of layers will be used to manage a set of mobile units (e.g. police vehicles). You must decide how the mobile units will be organized and how many layers you want to create.

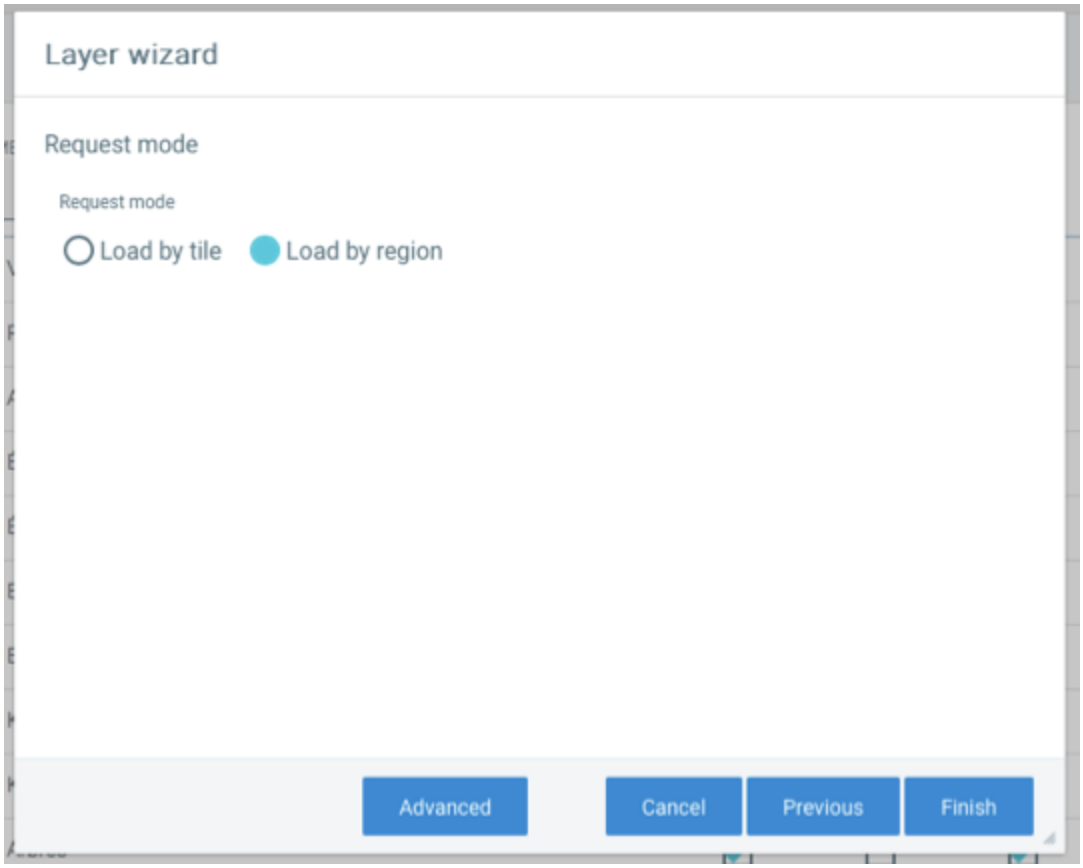
Follow these steps:

1. **Create** a layer (named Vehicles, for instance) of mobile units whose spatial data source is the Tracking data source created above.
2. At the bound attributes stage, you must, at a minimum, select the following attributes: MOBILE_ID, DIRECTION, TIME, and MOBILE_NAME.

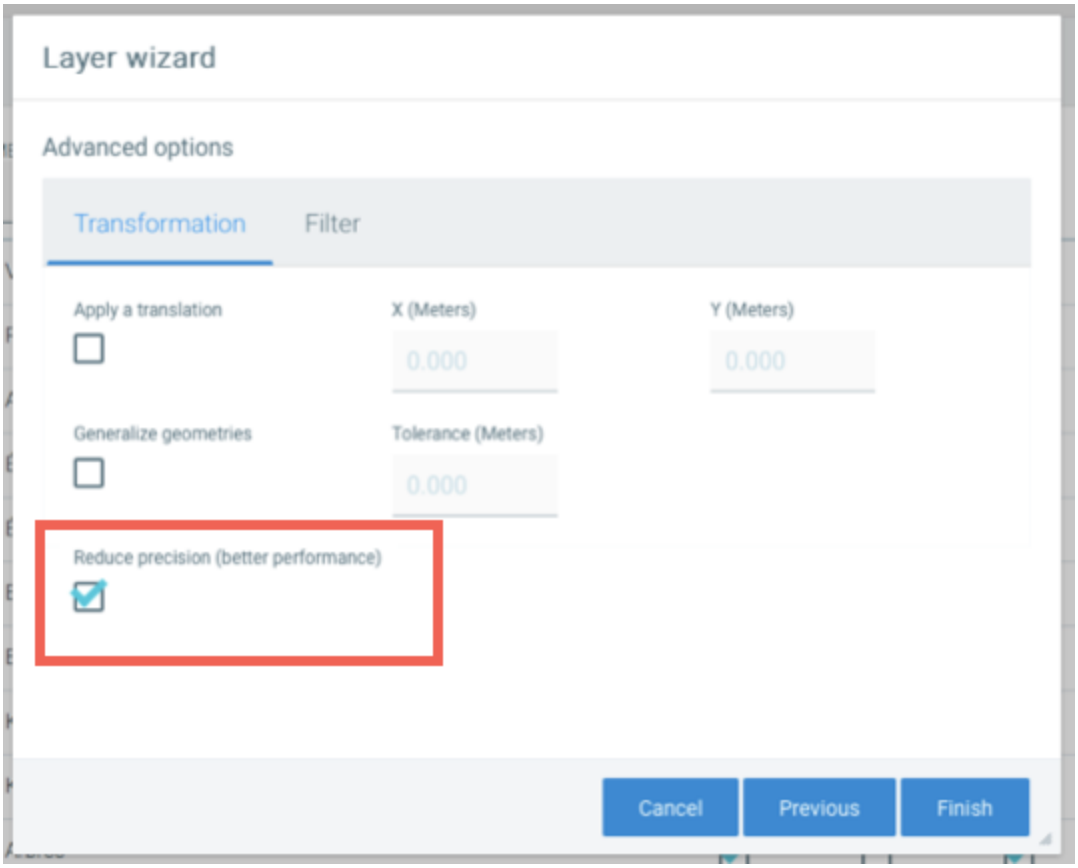


These attributes are required in order for Tracking to function. You can also select any other attribute you want to use in JMap.

3. At the request mode step, you must select **Load by region**.



4. In the **Advanced** section, disable the **Reduce precision** option.



5. Also, in the **Advanced** section, you can filter with SQL conditions to specify the content of the Tracking layer. This would allow you to have several Tracking layers with various mobile units. To do this, you can use the groups assigned to the mobile units, which are defined in the configuration of the mobile units. Refer to Managing Mobile Units and Groups for more information. This step is optional and you can easily come back to it later.
6. Click on **Finish**.

Important

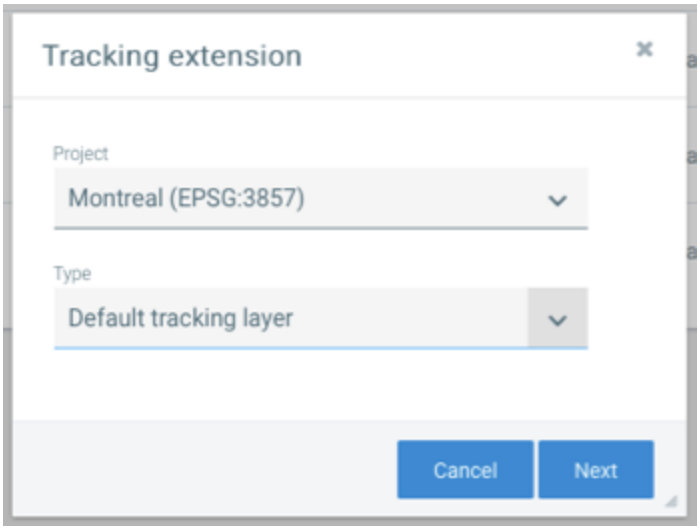
You must also create a second layer (named Vehicles - History, for instance), which will be identical to the previous one, but it will be used to display the history (previous positions) of the layer's mobile units.

Repeat the 6 previous steps for the history layer.

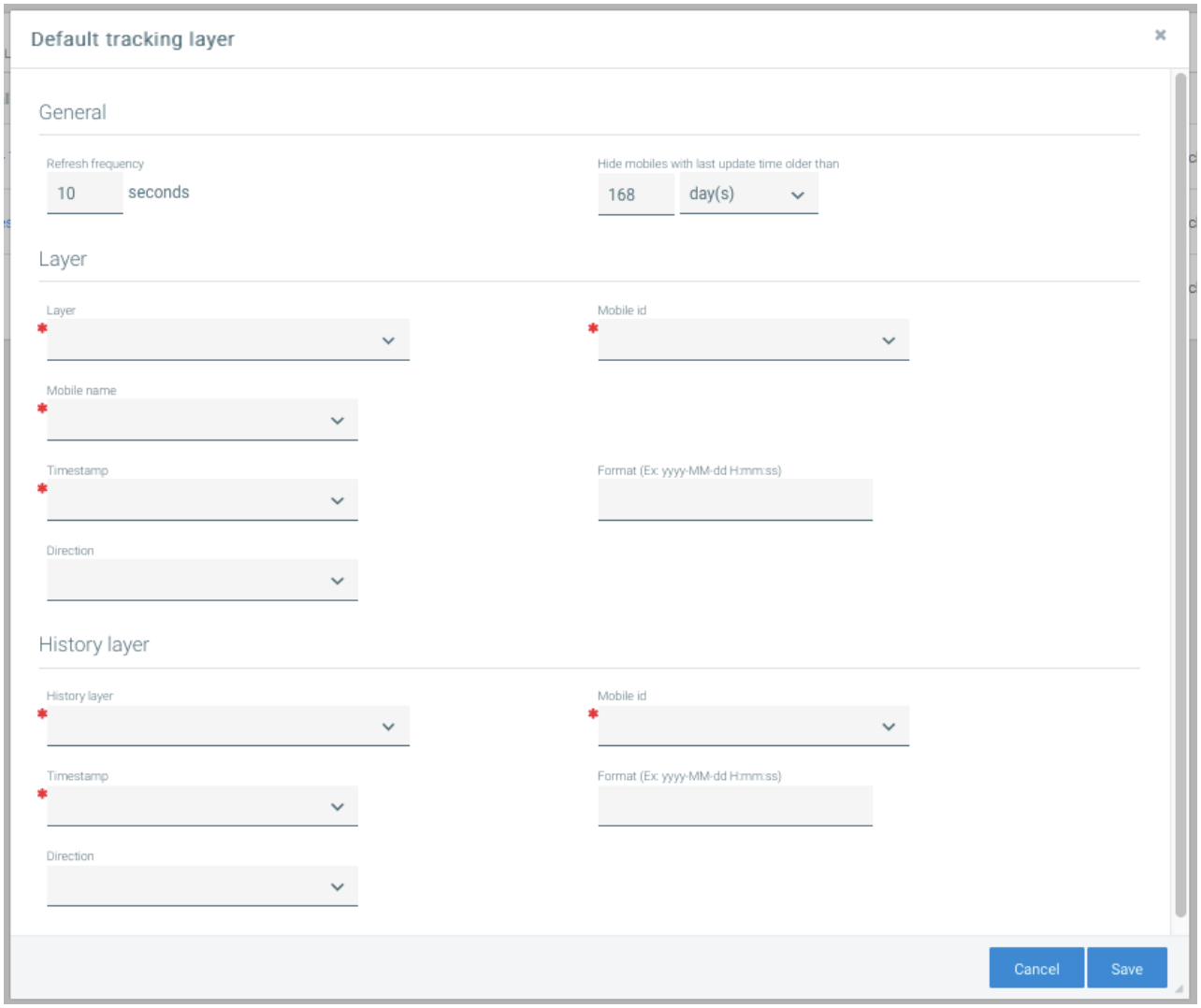
Configuring the Layers Managed by Tracking

Once the spatial data sources and various project layers have been prepared, the Tracking layers must be configured in the Tracking extension's configuration. These are Tracking layers that will be displayed in Tracking's graphical interface in JMap applications.

1. Open the **Tracking layers** section in the Tracking extension configuration.
2. Click on **Create** to create a Tracking layer configuration.
3. Select the project for which you wish to create a Tracking configuration as well as the type of configuration (normally Default tracking layer). Click on **Next**.



4. In this interface, you must enter several parameters to configure the Tracking layer.



The following table describes each parameter. When you are done, click on **Save**.

General	
Refresh frequency	Determines the period of time (in seconds) after which the layer will be automatically refreshed in JMap applications.
Hide mobiles with last update time older than	Allows you to hide, in JMap's applications, the mobiles of this layer that have not had any activity since the time specified. This setting can be modified personally by each Tracking user. Enter 0 to disable this function for the layer.
Layer	

Layer	Select the mobile units layer that must be handled by Tracking for this configuration. This layer must have been created previously (refer to the Preparing Data Sources and Layers section).
Mobile id	Among the attributes bound to the layer, select the one containing the unique identifier for the layer's elements.
Mobile name	Among the attributes bound to the layer, select the one containing the name of the mobile unit. This name will be displayed to users in JMap's applications.
Timestamp	Among the attributes bound to the layer, select the one containing the date and time of the mobile unit's last trip. This must be a Date or Timestamp attribute.
Format	Indicate the data format used by the values of the attribute containing the date/time. This format may vary depending on the DBMS used. MySQL: yyyy-MM-dd H:mm:ss SQLServer: yyyyMMdd HH:mm:ss Oracle: dd-MMM-yyyy h:mm:ss.S a
Direction	Among the attributes bound to the layer, select the one containing the direction of the mobile unit. The direction can be represented on the map by a rotation of the symbol based on the value of this attribute.
History layer	
History layer	Select the history layer that must be handled by Tracking for this configuration. This layer must have been created previously (refer to the Preparing Data Sources and Layers section).
Mobile id	Among the attributes bound to the history layer, select the one containing the unique identifier for the layer elements.
Timestamp	Among the attributes bound to the history layer, select the one containing the date and time of the mobile unit's last trip. The attribute type must be Date or Timestamp.
Format	Indicate the data format used by the values of the attribute containing the date/time. This format can vary based on the DBMS used. MySQL: yyyy-MM-dd H:mm:ss SQLServer: yyyyMMdd HH:mm:ss Oracle: dd-MMM-yyyy h:mm:ss.S a

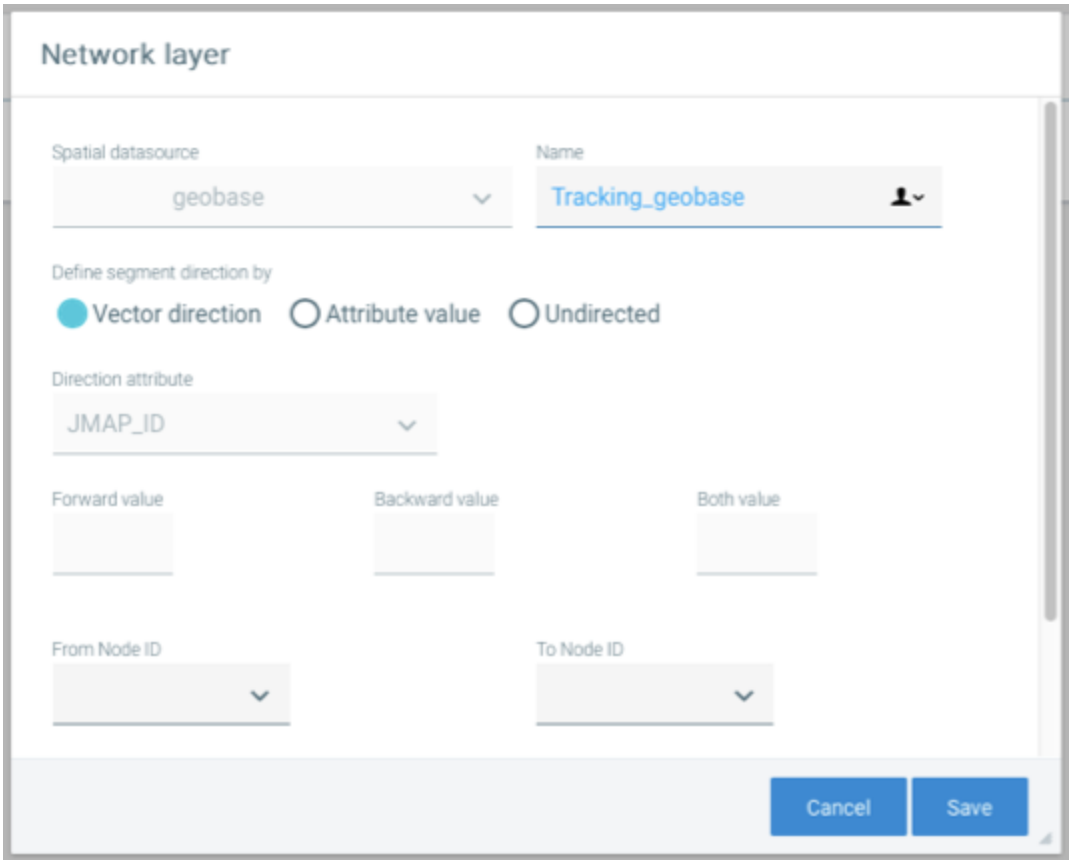
Direction	Among the attributes bound to the history layer, select the one containing the direction of the mobile unit. The direction can be represented on the map by a rotation of the symbol based on the value of this attribute.

Configuring Road Networks for Analysis

Layers can be configured for linear networks (road networks, sidewalk networks, etc.). These will be used for the analyses performed by Tracking. These networks are used to bind the data of mobile units in certain analyses and to produce maps that are easy to interpret.

Networks may or may not be directed. Directed networks increase the reliability of analysis results. In addition, networks can be single or double (one segment for each traffic direction).

1. Open the **Network layer** section in the configuration of the Tracking extension.
2. Click on **Create** to create a network layer configuration.



Interface for configuring a road network for analysis

Network layer parameters are described in the following table.

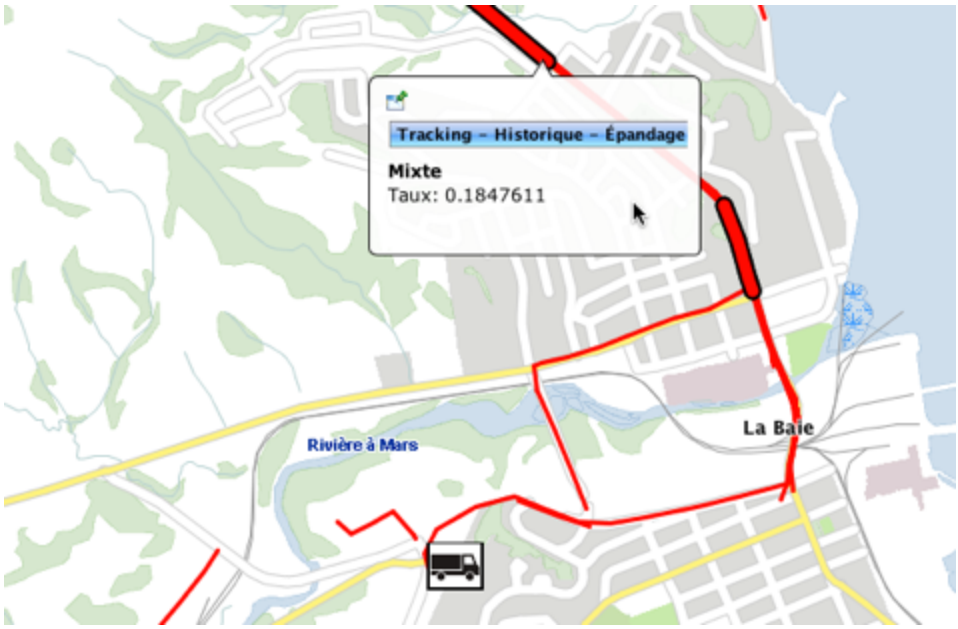
Network properties	
Spatial data source	Each network must be based on a spatial data source configured in JMap Admin. Select the data source. Once the network has been created, you can no longer change this parameter.

Name	Name you wish to give the network.
Define segment direction by	If the network has a direction, you must tell Tracking how to interpret the direction. In some analyses, Tracking will make better decisions if the network is directed. **Vector direction**: Tracking uses the vector direction (digitizing direction) to know the direction of network segments. This works well if the network is doubled (one segment per traffic direction). **Attribute value**: Tracking uses the value of an attribute combined with the vector direction (digitizing direction) to know the direction of the network segments. The values of the attribute used indicate the relationship between the vector direction and the traffic direction. This works well with single or double networks. **Undirected**: The network is undirected and Tracking will ignore traffic directions during its analyses.
Direction attribute	If you selected the **Attribute value** option, you must select the attribute containing the values that must be used to interpret the traffic directions.
Forward value	If you selected the **Attribute value** option, enter the attribute value indicating that the traffic direction is the same as the vector direction.
Backward value	If you selected the **Attribute value** option, enter the attribute value indicating that the traffic direction is contrary to the vector direction.
Both value	If you selected the **Attribute value** option, enter the attribute value indicating that the traffic is in both directions, with no connection to the vector direction.
From Node ID	If your network data contains topology, you can indicate the attribute containing the identifier of the node from which the segments originate. Using a network with topology improves the results of certain analyses.
To Node ID	If your network data contains topology, you can indicate the attribute containing the identifier of the destination node for the segments.
One Way	Select the attribute that indicates if the network segment is a one way. Some network data sources (not doubled) use such an attribute. If the value is 1 or true, it is a one way that matches the direction of the vector. Otherwise, it is not a one way.
Street Name	Select the attribute containing the street name of the network segments. Tracking uses this information to record the streets where the mobile units have traveled. If this is not applicable, do not select anything.

Unique ID	Select the attribute containing the unique identifier of the network segments. Tracking uses this information to record the streets where the mobile units have traveled. If this is not applicable, do not select anything.

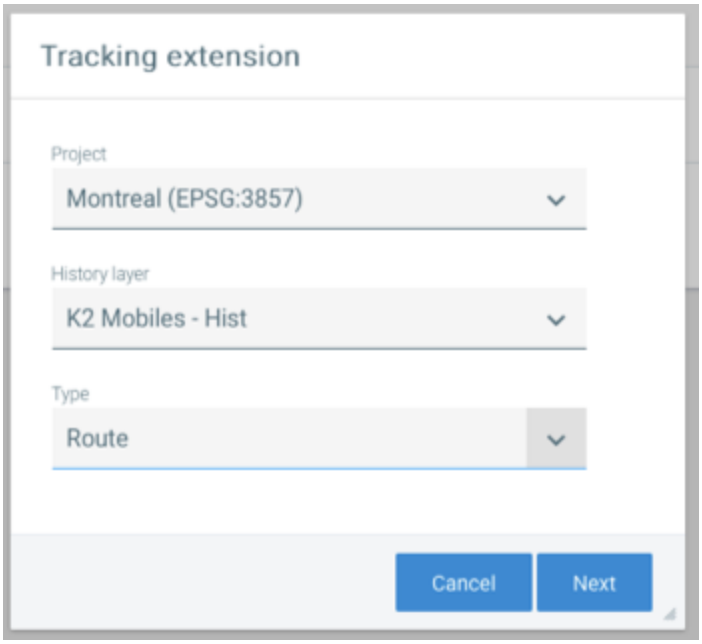
Configuring Types of Analyses

Tracking's analyses allow you to analyze the data of mobile units (history) and to produce thematic maps that are easy to interpret.



Spreading analysis example

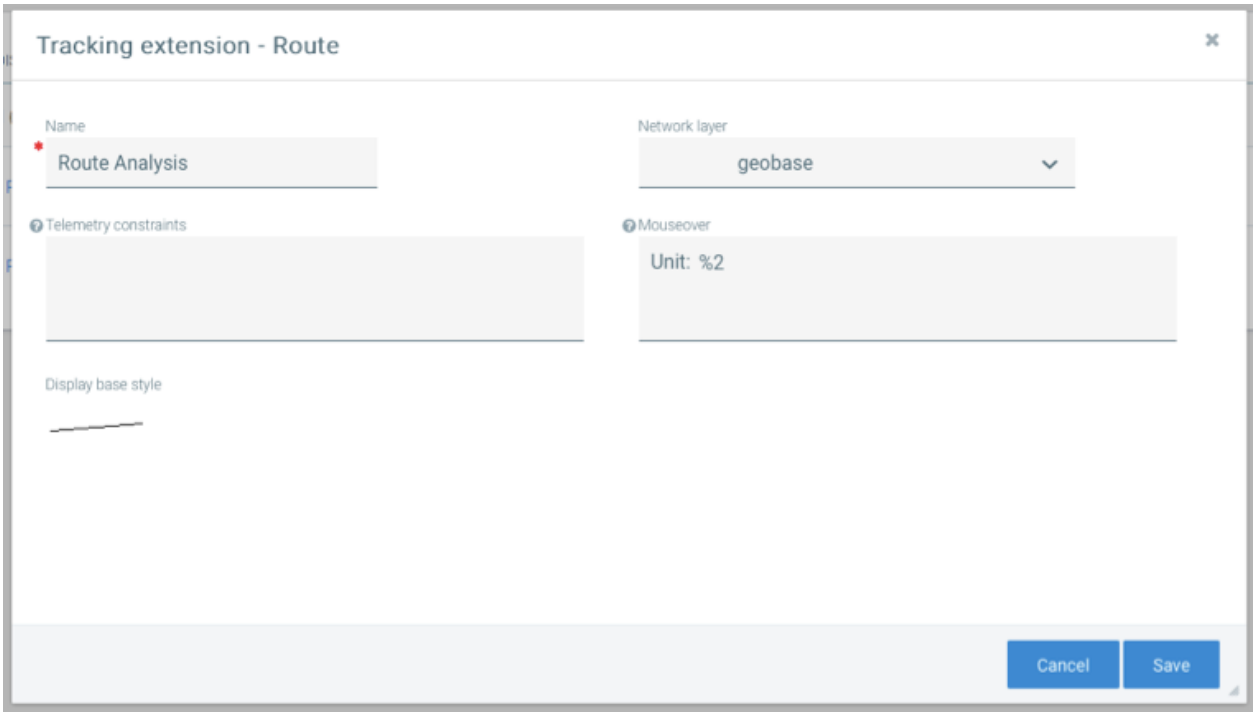
1. Open the **Analysis layers** section in the Tracking extension configuration.
2. Click on **Create** to create an analysis layer configuration.



Interface for creating an analysis layer

Analysis layer properties	
Project	Select the project in which this analysis will be created. Only projects that have Tracking layers are available.
History layer	Select the Tracking history layer that will be used to create the analysis.
Type	Select the type of analysis to configure. Several types of analyses are predefined, and each type has its own configurations. Other types of analyses may be added in the future. For more information, refer to Types of analyses.

3. Click on **Next**.



Interface for creating an analysis layer

Properties of the analysis layer	
Name	Name of the analysis layer. Users can see this name.
Telemetry constraints	Allows you to define conditions the data must meet to be considered by the analysis. For instance, in the case of a snow removal application involving vehicles equipped with sensors, there could be an analysis that only considers the data for which the vehicle's plough is lowered (on the ground). Other data (raised plough) would be ignored. The following syntax allows you to define the conditions. <ATTRIBUTE INDEX> <OPERATOR> <VALUE> The operators supported are: = != < <= > >= The attribute index refers to the attribute position in the list of attributes bound to the history layer that is used for the analysis. You can combine several conditions by separating them with a " ; " and they will be processed with a logical OR. Example: 3='true',9=1

Mouseover	The result of the analysis is a layer displayed on the map. The layer elements can include a mouseover bubble to provide additional information to the user. This parameter allows you to define the contents of the mouseover bubble. The values available are values calculated during the analysis, and they are generally limited to this information. However, some types of analyses can offer more values: %0 - ATTRIB_PASSAGES: Total number of passages on a segment of the network, by all of the mobile units. %1 - ATTRIB_LAST_PASSAGE: Date and time of the last passage on a segment of the network. %2 - ATTRIB_MOBILES: The list of mobile units that have passed on a network segment. %3 - ATTRIB_LAST_NAME: Name of the last mobile unit that passed on a segment of the network.
Network layer	The network layer to be used to analyze and represent the results. Some types of analyses do not require any network layer.
Styles	The base style to display the results of the analysis. If the analysis supports a thematic, several styles can be configured, along with the thematic's classes. These styles must previously have been configured as style templates.

Types of analyses

Types of analyses	
Route	This type of analysis presents the route taken by the mobile units. The segments covered are colored according to the style configured for the analysis. Telemetry constraints can be configured to filter the data analyzed. The route covered is based on the network selected for the analysis. The contents of the mouseover bubble can be configured.
Passage count / Time elapsed since last passage	This type of analysis presents the route segments covered with various styles, depending on the number of times the mobile units have passed over the segments. A thematic must be configured to define the styles for each value of the number of passages.

	This analysis also provides the time elapsed since the last passage of a mobile unit on each segment.
Ace	This is a specialized analysis for the application data of Ace control systems.
Dickey-John	This is a specialized analysis for the application data of Dickey-John control systems. The result shows the route segments based on the type of material applied and excess doses. A thematic must be configured to define the style for each case. It is also possible to know the application rates for each segment.
Multi Analysis	This type of analysis allows you to combine several analyses in one to perform them simultaneously.

Configuring Geofencing

The following parameters must be defined to ensure geofencing functions properly in Tracking. Geofence creation is controlled by permissions. Refer to Managing Permissions for more information.

Geofencing properties	
Validate geofences every	This parameter determines the frequency with which Tracking verifies data to monitor geofencing rules. This period should not be too long (longer delays before alarms are sent) or too short (Tracking work overload). The default value is 10 seconds, and this should be appropriate in most cases.
Alarms 'From' email	When Tracking sends alarms, emails can be sent automatically. In this field, enter the email address from which the emails will be sent.

Managing Permissions

Only one type of permission is available in this version of Tracking. Other types may be added in future versions.

Tracking permissions are managed by project. You must select the project for which you wish to modify Tracking's permissions. To grant permissions to users or user groups, you must start by adding them to the list.

Tracking permissions	
Manage geofences	Allows you to create, modify or delete geofences. **Note**: Geofences created by a user in a project can also be managed by all other Tracking users who have this permission for the same project.