Manual Técnico RastrackServices

Rastrack S.A.S.

Jorge Andrés Espinosa Pineda

Ingeniero de Sistemas

Jorgt.dna@gmail.com

Tabla de contenido

[1. Introducción 2](#_Toc9798549)

[2. Estructura del Aplicativo 2](#_Toc9798550)

[3. Configuration.properties 3](#_Toc9798551)

[a. Servicio Global FTP 4](#_Toc9798552)

[b. Servicio Data Base 5](#_Toc9798553)

[c. Servicio WS Widtech 5](#_Toc9798554)

[d. Servicio WS SkyWave 6](#_Toc9798555)

[e. Servicio WS InReach 7](#_Toc9798556)

[4. RastrackServices.rh 8](#_Toc9798557)

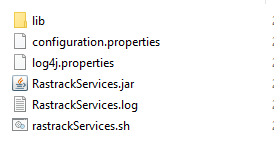
# Introducción

El siguiente documento contiene la información de cómo funciona el aplicativo RastrackServices.jar, donde se explica la configuración y su funcionamiento para poder ejecutarlo.

El aplicativo tiene 5 servicios configurados que permiten traer la información de las localizaciones de los vehículos que se están monitoreando por el aplicativo CRTApp.

# Estructura del Aplicativo

El aplicativo contiene la siguiente estructura de archivos:



**Lib:** La carpeta lib es donde se encuentran todas las librerías que usa el aplicativo, donde se encuentra el driver para la conexión hacia el motor de base de datos.

**Configuration.properties**: Es el archivo de configuración del aplicativo, que es fundamental para que el aplicativo funcione, aquí se encuentra toda la configuración de los servicios, base de datos y del servicio principal.

**Log4j**.**properties:** Es el archivo de configuración de log del aplicativo, que es importante para genera el log de aplicativo y ver las trazas de todo lo que escribe el aplicativo. En el siguiente link se puede encontrar información de como configurar este archivo según las necesidades. <https://logging.apache.org/log4j/2.x/>

**RastrackServices**.**jar:** Es el ejecutable del aplicativo, quien es el que inicia el procesamiento de los servicios.

**RastrackServices**.**log:** Es el log que registra todo el proceso del aplicativo.

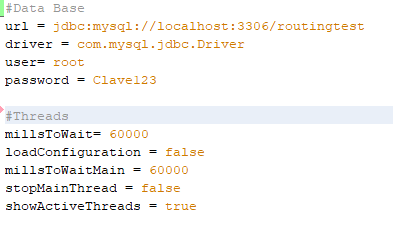
**rastrackServices**.**sh:** Es el Shell para iniciar el aplicativo.

# Configuration.properties

El aplicativo RastrackServices.jar tiene 5 servicios configurados que son servicios los que actualmente están implementados para traer la información de las coordenadas y localizaciones de los vehículos que tienen el servicio de monitoreo con la entidad. Y contiene un hilo principal que controla y carga la configuración para los 5 servicios configurados.

La configuración del aplicativo RastrackServices.jar se encuentra en el archivo llamada “configuration.properties”, donde están todos los parámetros que necesita el aplicativo para ejecutarse, y así mismo, poder iniciar los 5 Servicios de monitoreo.

En el archivo “configuration.properties” contiene las siguiente parametrización:



**url**: Es el parámetro de cadena de conexión de la base de datos configurada para persistir la información

**driver**: Es el parámetro donde se aplica el driver del motor de la base de datos a procesar, este driver puede ser cambiado colocando la librería en la carpeta lib, y cambia el nombre de la clase correspondiente al motor que se desea configurar.

**User**: Es el usuario con que se autentica a la base de datos

**Password**: Es la contraseña para autenticar a la base de datos.

**millsToWait:** Son los milisegundos que se configura a los hilos de cada servicio para que tome el tiempo de espera, y continúen procesando.

**loadConfiguration:** True indica que el aplicativo necesita cargar la configuración nuevamente, esto significa que algún parámetro haya cambiado. False es lo contrario y se mantiene ejecutándose con la parametrización inicial.

**milllsToWaitMain:** Son los milisegundos que se establece para que el hilo principal espere para que siga procesando.

**showActiveThreads:** Imprime en el log los hilos que actualmente se están ejecutando.

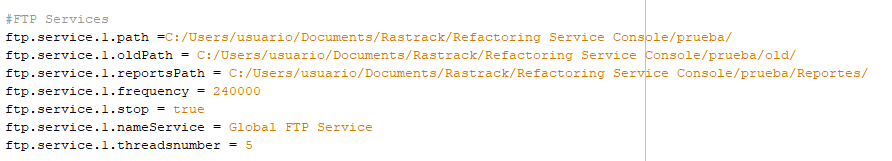
Estos 5 servicios se encuentran configurados de la siguiente forma:

## Servicio Global FTP

El servicio FTP consiste en lectura de un sistema de archivo que tiene una estructura XML, donde está almacenada la información del vehículo que está siendo monitoreando. La localización de los vehículos son reportados por el proveedor GLOBAL, y ellos son los que copian los archivos al sistema de archivo, que es el insumo del servicio para que los archivos empiecen a ser procesados.

Los archivos procesados terminan en un formato JSON con todas las variables que se encontró en el archivo XML, estos quedan almacenados en la tabla global\_unittype\_reports.

El servicio tiene la siguiente configuración:



Las propiedades declaradas para el FTP, empieza con [ftp.service](ftp://ftp.service), siempre y cuando que haya más de dos servicios configurados, estos se enumeran para que tome la configuración específica de cada uno de los servicios. En este caso se configuró un servicio para FTP con las propiedades siguientes:

**Path:** Es el directorio donde se guardan los archivos a procesar, y que no han sido procesados todavía por el servicio.

**oldPath:** Es el directorio donde se guardan los archivos que ya han sido procesados, una vez procesados, el servicio los mueve a esta carpeta indicando que el servicio ya ha sido procesado.

**reportsPath:** Es el diretorio donde se guardan los reportes del procesamiento de los archivos de los vehículos que tiene configurado en la base de datos, la columna CREATE\_TEXT\_FILE de la tabla tbvehicle en true, de lo contrario no colocará el archivo procesado en esa carpeta.

**Frecuency:** Parámetro donde se establece los milisegundos que el servicio va a esperar para volver a ejecutarse.

**Stop:** Es el parámetro que indica true que el servicio no se va a ejecutar, false indica que el servicio si se ejecutará.

**nameService:** Es el nombre que se le asigna al servicio para ser identificado en el log.

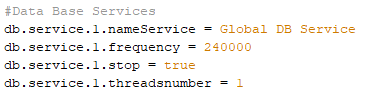
**Threadsnumber:** Es el parámetro que establece cuantos hilos se configuran para el procesamiento de los archivos.

## Servicio Data Base

El servicio Data Base consiste en la lectura de formato JSON que fue almacenado en la tabla global\_unittype\_reports y los convierte en un registro en la tabla vehiclelocation para cada vehículo reportado.

La configuración inicia con db.service, y si existe configurado más de un servicio, se comienzan a enumerar.

El servicio tiene la siguiente configuración



**nameService:** Es el nombre que se le asigna al servicio para ser identificado en el log.

**Frecuency:** Parámetro donde se establece los milisegundos que el servicio va a esperar para volver a ejecutarse.

**Stop:** Es el parámetro que indica true que el servicio no se va a ejecutar, false indica que el servicio si se ejecutará.

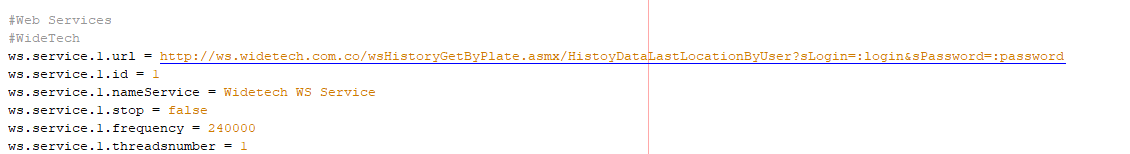
**Threadsnumber:** Es el parámetro que establece cuantos hilos se configuran para el procesamiento de los registros.

## Servicio WS Widtech

El servicio WS Widtech consiste en un consumo de un servicio web, definido en el archivo de configuración, donde extrae la información de la localización de los vehículos que se encuentran registrados allí. Esta información es reportada por el proveedor de Widtech. También el servicio web requiere autenticación por lo que tiene consultar a la base de datos del usuario y contraseña y autenticarse al servicio.

La configuración inicia con ws.service, y si existe configurado más de un servicio, se comienzan a enumerar. En este caso el servicio está asociado al número 1.

El servicio tiene la siguiente configuración:



**url:** Servicio Web que el servicio consume para traer la información de los vehículos que se encuentran registrados allí.

**Id:** Id del servicio que se encuentra registrado en la base de datos, para poder traer los parámetros de configuración del servicio.

**nameService:** Es el nombre que se le asigna al servicio para ser identificado en el log.

**Frecuency:** Parámetro donde se establece los milisegundos que el servicio va a esperar para volver a ejecutarse.

**Stop:** Es el parámetro que indica true que el servicio no se va a ejecutar, false indica que el servicio si se ejecutará.

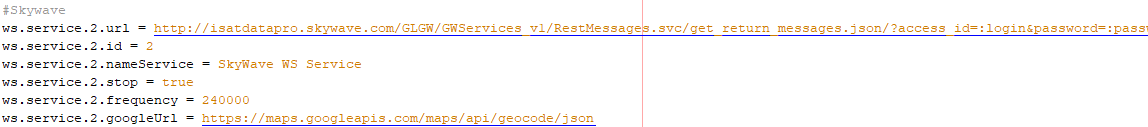
**Threadsnumber:** Es el parámetro que establece cuantos hilos se configuran para el procesamiento de los registros.

## Servicio WS SkyWave

El servicio WS Skywave consiste en un consumo de un servicio web, definido en el archivo de configuración, donde extrae la información de la localización de los vehículos que se encuentran registrados allí. Esta información es reportada por el proveedor de Skywave. También el servicio web requiere autenticación por lo que tiene consultar a la base de datos del usuario y contraseña y autenticarse al servicio.

La configuración inicia con ws.service, y si existe configurado más de un servicio, se comienzan a enumerar, en este caso está asociado al número 2.

El servicio tiene la siguiente configuración:



**url:** Servicio Web que el servicio consume para traer la información de los vehículos que se encuentran registrados allí.

**Id:** Id del servicio que se encuentra registrado en la base de datos, para poder traer los parámetros de configuración del servicio.

**nameService:** Es el nombre que se le asigna al servicio para ser identificado en el log.

**Frecuency:** Parámetro donde se establece los milisegundos que el servicio va a esperar para volver a ejecutarse.

**Stop:** Es el parámetro que indica true que el servicio no se va a ejecutar, false indica que el servicio si se ejecutará.

**Threadsnumber:** Es el parámetro que establece cuantos hilos se configuran para el procesamiento de los registros.

## Servicio WS InReach

El servicio WS InReach consiste en un consumo de un servicio web, definido en el archivo de configuración, donde extrae la información de la localización de los vehículos que se encuentran registrados allí. Esta información es reportada por el proveedor de InReach. También el servicio web requiere autenticación por lo que tiene consultar a la base de datos del usuario y contraseña y autenticarse al servicio.

La configuración inicia con ws.service, y si existe configurado más de un servicio, se comienzan a enumerar, en este caso está asociado al número 3.

El servicio tiene la siguiente configuración:



**url:** Servicio Web que el servicio consume para traer la información de los vehículos que se encuentran registrados allí.

**Id:** Id del servicio que se encuentra registrado en la base de datos, para poder traer los parámetros de configuración del servicio.

**nameService:** Es el nombre que se le asigna al servicio para ser identificado en el log.

**Frecuency:** Parámetro donde se establece los milisegundos que el servicio va a esperar para volver a ejecutarse.

**Stop:** Es el parámetro que indica true que el servicio no se va a ejecutar, false indica que el servicio si se ejecutará.

**Threadsnumber:** Es el parámetro que establece cuantos hilos se configuran para el procesamiento de los registros.

# RastrackServices.rh

Este archivo contiene el script de ejecución del servicio RastrackServices.jar. El servicio siempre se debe iniciar con este script, para así controlar que sólo sea el único aplicativo en ejecución.

Este servicio tiene tres operaciones:

**Start:** Parámetro para iniciar el aplicativo con los parámetros establecidos en el archivo configuration,properties. El comando para iniciarlo es el siguiente

Sudo sh RastrackServices.sh start

**Stop:** Parámetro para detener el aplicativo, y en el cual el programa comienza a detener todos los hilos que se estaba ejecutando en este momento. En el log se puede ver como finaliza la aplicación. El comando para detenerlo es el siguiente:

Sudo sh RastrackServices.sh stop

**Load:** Parámetro que sirve para recargar el archivo configuration.properties siempre y cuando este haya sido modificado, para así no detener el aplicativo por complete y siga operando, sin embargo el aplicativo tiene que detener todos los hilos que se están ejecutando, para así recargar toda la configuración y volver a iniciar. El comando para cargar la configuración es el siguiente:

Sudo sh RastrackServices.sh load

# Repositorio

Todo el proyecto se encuentra subido en un servidor git, la dirección es la siguiente:

https://git-codecommit.us-east-2.amazonaws.com/v1/repos/RastrackServices