LOCATION STUDIO™

LOCATION STUDIO QT AUTOMOTIVE SUITE API GUIDE

June 2017

COMTECH TELECOMMUNICATIONS CORP.

Revision Sheet

Release No.	Date	Revision Description
Rev. 1	6/13/17	Initial Draft

TABLE OF CONTENTS

GENERAL INFORMATION

Purpose

The purpose of this document is to provide the detailed information about the hardware and software environment settings and configurations needed to successfully load and run the Location Studio Automotive Reference application on the Linux platform for x86 -64 hardware environment.

Linux Hardware and Software Requirements

Name	Vendor, Model and Serial #
Device	VMware player(>=6.0) or x86-64 motherboard
Operating System	Ubuntu 16.04.01 LTS, x86, 64bit
QT Version	5.8.0
Dependency Libraries	libgbm1, libegl1-mesa, libegl1-mesa-drivers, libgles2-mesa, libgles2-mesa-dev, libgl1-mesa-dri, libasound2-dev, alsa-oss, libphonon-dev, libwayland-client0, libwayland-server0, libsqlite3-dev, libfreetype6-dev, libgstreamer1.0-dev, libwayland-dev, libwayland-egl1-mesa*, libgstreamer1.0-0, libssl-dev, libarchive-dev, libyaml-dev, libfontconfig1-dev, libdbus-1-dev, libdbus-1-3, libdbus-glib-1-dev

LocationStudio Console and Cluster Plugins

Platform Support

Linux X11 with Qt 5.8 Automotive

LocationStudio Console Plugin

This LocationStudio console plugin supports the features of mapTile, navigation and search services. This plugin exposes the custom qml type **`ConsoleView`**. Properties like zoom, tilt and followme mode can be set from **`ConsoleView`**. The plugin also offers feature like category search, single search, and local device store option for recently searched places and it can be added to a favorite list. While in navigation mode, the application offers a detailed maneuver list with total remaining time and distance along with direction and street name details for different route options. Another feature during navigation is the current Speed limit. While navigating if fuel level reduces to certain low level, alert is shown to redirect to nearest fuel station and after again re-routed to original destination. Plugin has options to view optional layers like satellite view, traffic view, 3D view and POI.

Properties

Parameters	Description
`width`	Sets width of application
`height`	Sets height of application
`zoom`	Sets default zoom level for the map
`tilt`	Sets the angle for the map
`isFollowMe`	Sets the map to follow and rotate as device moves
`workFolder`	Absolute path to map cache and resource directories. The default path for this is `/opt/locationstudio/res/console`.
`token`	API Key created on your Location Studio developer account under the ‘Applications’ tab. It is required for authentication.

Example Usage

```
import com.locationstudio.qtnavigator.console 1.0
…….

ConsoleView {
      width: 1024
      height: 768
      zoom: 17
      tilt: 45
      isFollowMe: true
      workFolder: "/opt/locationstudio/res/console"
      token: “<YOUR_APIKEY_HERE>”
}
```

LocationStudio Cluster Plugin

The LocationStudio cluster plugin is an application that has the features of mapTile integrated with navigation services. This plugin exposes the custom qml type **`ClusterView`**. Properties like zoom, tilt and followme mode can be set from clusterview. Apart from the map, while in navigation mode, the plugin offers maneuver details with remaining time and distance along with direction and street name details for given route. Also display the current Speed limit.

Parameters	Description
`workfolder`	Absolute path to map cache and resource directories. The default path for this is `/opt/locationstudio/res/cluster`
`zoom`	Sets default zoom level for the map
`tilt`	Sets the angle for the map
`isFollowMe`	Sets the map to follow and rotate as device moves
`avatarPosition.x`	Sets avatar latitude
`avatarPosition.y`	Sets avatar longitude
`avatarHeading`	Sets avatar heading
`token`	API Key created on your Location Studio developer account under the ‘Applications’ tab. It is required for authentication.

Example Usage

```
import com.locationstudio.qtnavigator.cluster 1.0

…….

ClusterView {
      id: map2
      workFolder: "/opt/locationstudio/res/cluster"
      zoom: 17.0
      tilt: 15.0
      avatarPosition.x: -118.25
      avatarPosition.y: 34.05
      avatarHeading: 0.0
      isFollowMe: true
      token: <provided by comtech>
}
```

Console-Cluster Communication Protocol

D-Bus is used as the interface for communication from Console to Cluster. The following are the APIs. Any module can register for these interfaces to be notified of that information.

onManeuverUpdated

```
Q_SCRIPTABLE QString onManeuverUpdated(const QString& maneuverIcon,
const QString& sreetName, const QString& distance, const QString&
trafficWarning);
```

Description: Notified when there is update for current Maneuver during navigation.

updateRemainingManeuverDistance

```
Q_SCRIPTABLE QString updateRemainingManeuverDistance(const QString&
distance);
```

Description: Notified when there is update for current Maneuver distance during navigation.

updateGpsPosition

```
Q_SCRIPTABLE QString updateGpsPosition(const QString& lat, const
QString& lon, const QString& heading);
```

Description: Notified when there is update in gps position.

setNavigationMode

```
Q_SCRIPTABLE QString setNavigationMode(bool navMode);
```

Description: This notifies about an active navigation session.

onUpdateSpeedLimit

```
Q_SCRIPTABLE bool onUpdateSpeedLimit(const QString& speedLimit)
```

Description: This notifies about speed limit during navigation

updatePolyline

```
Q_SCRIPTABLE  bool updatePolyline (const QString &polyline)
```

Description: This notifies about polyline update during navigation.

clearPolyline

```
Q_SCRIPTABLE  bool clearPolyline ()
```

Description*:* This notifies to clear polyline while navigation end.

updateManeuverArrow

```
Q_SCRIPTABLE  bool updateManeuverArrow (const QString &pts)
```

Description*:* Notified when there is update for maneuver arrow on polyline during navigation.