

LOCATION STUDIO ™



LOCATION STUDIO QT DEVELOPER'S GUIDE

July 2017



Revision Sheet

Release No.	Date	Revision Description
Rev. 0	7/13/17	Initial Draft

TABLE OF CONTENTS

OVERVIEW	1
Getting Started	1
Install QT Creator & Build the plugin.....	1
Qt Console Plugin Implementation	2
Display Map	2
Side Menu, Title Bar	3
Keyword Search.....	3
Category Search.....	4
Recent List and Add/Remove Recent	5
Favorites List and Add/Remove Favorites	6
POI Details.....	7
Route List	7
navigationRouteController	7
NavigationSession	9
Manuver List	13
Console Example usage	13
QT Cluster Plugin Implementation	14
Display Map	14
Cluster Example usage.....	14
Console-Cluster communications Protocol.....	15
Cluster-Console communications.....	16

OVERVIEW

This document is intended to serve as a reference for Developers as they work on Location Studio Qt Navigation Console & Cluster Plugin. Both Console and Cluster are Custom QML Type QQuickItems registered as plugins with User Interface implemented in QML. The document will fully define the Role of Developer in Location Studio Qt Project.

Getting Started

This section explains how to build and run Location Studio Qt Navigation Console Plugin & Cluster plugins on Ubuntu 16.04.01 LTS x86 64bit system. You can also find this information in the Auto Reference Getting Started Guide in this same docs directory.

<https://git.location.studio/location.studio/autoNavSDK/blob/master/docs/AutoReference%20Getting%20Started%20Guide%20R1.0.pdf>

Install QT Creator & Build the plugin

1. Download and install QT Creator (Qt Creator 4.3.1 for Linux 64-bit) from <https://www.qt.io/download-open-source/>
2. Open Project file of Location Studio Console Plugin in QT creator.
3. In QT > Projects > Manage Kits > Build & Run > Kits
4. Select Auto-Detected kit and create clone of default kit. Rename prefix of kit to LTK-....
5. Select newly created kit in Manual and change device to Desktop, Change QT Version to path already copied at /opt folder. (Path: **/opt/Qt5.8.0/bin**)
6. Build the plugin from QT creator.
7. To install the plugin, Open Terminal/Command Prompt, Go to path_to_project_folder/
make && sudo make install
8. This plugin will be installed in
`/opt/Qt5.8.0/qml/com/locationstudio/qtnavigator/plugin_folder`

Qt Console Plugin Implementation

This plugin supports the features of map, navigation and search services. This plugin exposes the custom qml type **ConsoleView**. Properties like zoom, tilt and follow-me mode can be set from **ConsoleView**. This 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. The plugin has options to view optional layers like satellite view, traffic view, 3D view and POI.

Display Map

The MapKit3d library used for displaying a map on the home screen. The **mLTKMapKit** object of **MapWidget** class is initialized in **MapViewController::init()** and added to main layout in **MainViewController::init()** to display the map. The **MapViewController::SetupMapWidget()** function has direct slots pertaining to MapWidget.

No.	Slot	Description
1	OnMapCreated()	Here the map's zoom, tilt, reference center, avatar position, setnightmode option and map decoration parameters like zoom, follow me and compass button are set.
2	OnLayersCreated()	Here the layer options button is enabled and the zoom and layer button visibility is set.
3	GetPinInfo(const locationtoolkit::Pin*)	Signal SignalPinInfoClicked() is emitted if navigation is started. This signal is connected to mCategorySearchController slot GetPinInfo() where pininfo is updated and displayed with complete information with DetailViewcontroller:show().
4	onCameraUpdated(const locationtoolkit::CameraParameters&)	Camera position is updated to be set for map center.
5	onPolylineClicked(const QList<locationtoolkit::Polyline*>)	If Navigation is not started then make all polylines passive else active/raise the selected polyline.

Side Menu, Title Bar

The Titlebar is created in **TitleBarController** and called in **MainViewController::init()**. Height, Width and position for the Titlebar are set from MainViewController. UI customizations are done in **TitleBar.qml**. Side menu button option, Keyword search & settings button option are added in Titlebar and signals are emitted from qml and connected to slots in MainViewController for all the options.

No.	Slot	Description
1	onSettingsButtonClicked()	SettingControllerMenu is created, width, height and parent is set in setup method SettingController.qml is created that has UI.
2	onSearchButtonClicked(const QString &searchString)	Search category is set to 'all' and string is passed to mCategorySearchController->mainBarSearchSDKRequest(parameter, searchString); in CategorySearchController.
3	onMenuClicked()	HamburgerControllerMenu is created, width,height and parent is set and in setup method HamburgerMenu.qml is created that has UI.

Keyword Search

Keyword search can be performed by entering search string from Title bar text field. If user types 3 characters, app redirects to **TitleBar.qml** signal **showSuggestedSearch()** that connects to signal **onSearchButtonClicked()** in MainViewController and there to slot **MainViewController::onSearchButtonClicked()**.

mCategorySearchController Object of **CategorySearchController** class initialized in function **MainViewController::InitializeCategoryController()**, **Fuel.qml** is the UI customization file in which Listview for search result is created and signals are emitted for which slots are in **CategorySearchController**.

No.	Slot	Description
1	<code>onClearAllResentsClicked()</code>	Calls <code>PlaceManager::clearAllRecents()</code> which deletes data in <code>recent.json</code> .
2	<code>onClearAllFavouritesClicked()</code>	Calls <code>PlaceManager::clearAllFavorites()</code> which deletes data in <code>favorites.json</code> .
3	<code>onDeleteSelectiveFav (QString latitude,QString longitude)</code>	Calls <code>PlaceManager::deleteExistingFavConfig()</code> which searches for the matched data and deletes the same from <code>favorites.json</code> . and that particular <code>favorite[index].json</code>
4	<code>on_mapToggleButton_clicked()</code>	For the search result, <code>locationtoolkit::PinParameters</code> are filed and passed to <code>mLTKMapKit->CreatePin(pinpara);</code> and shown on map.
5	<code>onStartNavigationClicked(int)</code>	Index is used to save data in recent list <code>placeManager->addToRecents()</code> , and signal <code>SignalNavigationButtonClicked()</code> is emitted which is connected in <code>mapviewController::OnStartNavigationToDestination()</code>
6	<code>onLoadListDetail(int itemData)</code>	Index is used to save data in recent list <code>placeManager->addToRecents()</code> , and <code>SerachDetailsController::show()</code> is called and related data is passed to <code>mDetailInfoView</code> .
7	<code>loadMoreSearchSDKRequest()</code>	<code>locationtoolkit::SingleSearchRequest::createRequest()</code> is passed with <code>SD_Next</code> parameter and <code>onSearchSuccess()</code> slot more results are loaded

Category Search

Category search can be performed by click on side menu from home screen, Click on 'Near Me' option, Select any category from list or major categories like Theaters, Gasoline are listed in the listview of Hamburger menu along with Favorite and Recent List.

HamburgerMenu.qml has signals that connects to slots in **CategorySearchController::onCategorySearchSelected()** with category parameter code.

Slots in Hamburger Menu emits signal to connect to **CategorySearchController**.

No.	Slot	Description
1	onNearMeButtonClicked()	NearControllerMenu is created, width,height and parent is set and in setup method NearMeController.qml is created that has UI. Signal nearMeItemsClicked() from UI is connected to slot onNearMeItemsClicked() which emits signal CategorySearchSelected() which is connected to slot in category onCategorySearchSelected() with category parameter code.
2	onMoviesButtonClicked()	signal CategorySearchSelected() which is connected to slot in category onCategorySearchSelected() with category parameter code. for movies.
3	onGasStationButtonClicked()	signal CategorySearchSelected() which is connected to slot in category onCategorySearchSelected() with category parameter code for GasStation.
4	onFavouritesButtonClicked()	Signal FavouriteButtonClicked() is emitted with is connected to CategorySearchController::displayExistingFavourites() in MainViewController::onMenuClicked() .
5	onRecentsButtonClicked()	Signal RecentButtonClicked() is emitted with is connected to CategorySearchController::displayExistingRecents() in MainViewController::onMenuClicked() .

Recent List and Add/Remove Recent

Recent List is shown Home screen > Hamburger menu > Recent.

CategorySearchController is used to show Recent List.

Slots used for Recent are:

No.	Slot	Description
1	CategorySearchController::displayExistingRecents()	Defined in CategorySearchController. This function updates recents in listview ofCategorySearchController.
2	CategorySearchController::onClearAllResentsClicked()	This slot is defined in CategorySearchController. It will show confirmation alert to user for deleting all Recents. On click Yes from confirmation alert, placeManager::clearAllRecents() method will be called that will clear all entries in local files.
3	PlaceManager::addToRecents()	Defined in PlaceManager. This method is executed when POI details are shown. Used to Add POI to recent. This slot internally calls PlaceManager::writeRecentsConfig to save POI details to file and make entry in recents file.

Favorites List and Add/Remove Favorites

Favorites List is shown Home screen > Hamburger menu > Favorites.

CategorySearchController is used to show Favorites List.

Slots used for Favorites:

No.	Slot	Description
1	CategorySearchController::displayExistingFavourites()	Defined in CategorySearchController. This function updates favorites in listview of CategorySearchController.
2	CategorySearchController::onClearAllFavouritesClicked	This slot is defined in CategorySearchController. It will show a confirmation alert to the user for deleting all favorites. On click Yes from confirmation alert, placeManager::clearAllFavourite() method will be called that will clear all entries in local files.

POI Details

POI Details are shown by clicking on any row from a keyword search or category search list view or tapping on any pin from the map.

mDetailInfoView Object of SearchDetailsController class initialized in function CategorySearchController::setUp().

Height, Width and parent for SearchDetails are set from CategorySearchController and in SearchDetailsController::setup(). **FuelDetails.qml** is created that has UI and signals that are emitted which are connected in SearchDetailsController and CategorySearchController depending on the signals.

POI details button click slots are:

No.	Slot	Description
1	onNavigationButtonClicked(QString, int, QVariant)	Emits signal SignalNavigationButtonClicked() which calls slot in MapViewController::OnStartNavigationToDestination to start fetching routes for navigation.
2	on_callButton_clicked(QString, QString, QString)	Show call screen. CallViewController is used for simulating call screen. CallViewController::setup() will create CALLPOI.qml which will have UI for call screen and signal callEndButton() that would be connected to slot CallViewController::on_call_endButton_clicked() to clear call screen. And end call
3	onMapButtonClicked	Toggle between POI details and Map screen.
4	onFoundCustomFavourite()	Used to add or remove POI location to favorite listplaceManager::addToFavorites, placeManager->removeFromFavorites. And check Favorite.json to show if Item is already added to list then mark it as Favorite.

Route List

navigationRouteController

From POI details screen, On click of navigation button, the Route List on the left side of the home screen and route paths are displayed on the Map. Class navigationRouteController is used to show route list.

mNavigationRoutesController object of navigationRouteController class initialized in function MapViewController::setupNavigationRouteController. In navigationRouteController::setup(), NavigationRouteController.qml is created that has the UI for Route selection.

Signals for selected route index are emitted from navigationRouteController to connect to slots **MapViewController::onStartNavigationOnRoute()** where route polyline, maneuver details and compass are updated and signals for **startNavigationOnRoute()** is emitted which is connected to **NavigationSession::onStartNavigationOnRoute()**.

To fetch route information, the **mNavigationImpl** Object of **NavigationSession** class is used. **NavigationSession** class uses NavKit SDK to get route information as well as Maneuver updates and direction updates. **NavigationSession** emits appropriate signals which call slots of MapViewController and it in turn it calls slots of navigationRouteController to update route list UI.

Slots/Functions used in navigationRouteController are:

No.	Slot	Description
1	onRouteFirstClicked() onRouteFirstGoClicked()	Emits signal startNavigationOnRoute() with index which calls slot in MapViewController::onStartNavigationOnRoute() that sets cameraposition, add polyline, set compass and emit signal startNavigationOnRoute() that is connected to NavigationSession::onStartNavigationOnRoute() where RouteInformation is passed to mNavigationImpl::SetActiveRoute for index 0.
2	onRouteSecondClicked() onRouteSecondGoClicked()	Emits signal startNavigationOnRoute() with index which calls slot in MapViewController::onStartNavigationOnRoute() that sets cameraposition, add polyline, set compass and emit signal startNavigationOnRoute() that is connected to NavigationSession::onStartNavigationOnRoute() where RouteInformation is passed to mNavigationImpl::SetActiveRoute for index 1.

3	<pre>onRouteThirdClicked() onRouteThirdGoClicked()</pre>	<p>Emits signal <code>startNavigationOnRoute()</code> with index which calls slot in <code>MapViewController::onStartNavigationOnRoute()</code> that sets cameraposition, add polyline, set compass and emit signal <code>startNavigationOnRoute()</code> that is connected to <code>NavigationSession::onStartNavigationOnRoute()</code> where <code>RouteInformation</code> is passed to <code>mNavigationImpl::SetActiveRoute</code> for index 2.</p>
---	--	--

NavigationSession

NavigationSession class is used to get route list, maneuvers list, direction arrow images, navigation remaining distance, remaining times, next road, current road related updates using **NavKit SDK**.

mNavigationImpl object of this class is initialized in `MapViewController()`.

Slots assigned in `NavigationSession` are:

No.	Slot	Description
1	<pre>OnStartNavigation(LTKContext& ltkcontext, const Place& place, const RouteOptions& routeoption, const Preferences& preference)</pre>	<p>Slot executed from mapviewController to initialize Navigation Object (mNavigationImpl) of NavKit SDK and fetch routes. This object is used to assign slots to session signals and navigation related signals.</p>
2	<pre>OnStopNavigation()</pre>	<p>To stop navigation session and location updates. Executed from mapviewController.</p>
3	<pre>OnUpdateGpsFile(QString gpsfile)</pre>	<p>To update gps file path which will be used for navigation simulation and location updates. Executed from mapviewController.</p>
4	<pre>onStartNavigationOnRoute(int routeIdx)</pre>	<p>Executed from mapviewController class, Navigation object sets specified route as active route for location and navigation</p>
5	<pre>NavigationSession::modifyLocati onServices(bool startService)</pre>	<p>To start/stop location updates.</p>

Slots assigned for session signals are:

No.	Slot	Description
1	<code>OnOffRoute()</code>	Emit when the locations being received by the navigation session are not considered on route to destination.
2	<code>OnRoute()</code>	Emit when the locations being received by the navigation session are considered on route to destination.
3	<code>OnRouteReceived(RouteRequestReason reason, const QVector<QSharedPointer<RouteInformation> >& routes)</code>	Emit when received the list of possible routes for the current navigation session. It emits <code>addPolyLinesForPath(reason, routes)</code> signal which will add polylines to map for received routes in <code>MapViewController</code> .
4	<code>OnRouteRequested(RouteRequestReason reason)</code>	Emit when new route requested
5	<code>OnRouteProgress(qint32 progress)</code>	Emit when the route generation in progress
6	<code>OnRouteError(LTKError error)</code>	Emits when there is any error in getting routes details
7	<code>OnArrivingToDestination(DestinationStreetSide streetSide)</code>	Emit when the current navigation arriving to destination.
8	<code>OnRouteFinish()</code>	Emit when the current navigation session ends. The client will not receive any route updates.
9	<code>OnOffRoutePositionUpdate(qreal headingToRoute)</code>	Emit when the heading to origin off route.

Slots assigned for navigation update signals:

No.	Slot	Description
1	<code>OnTripRemainingTime(quint32 time)</code>	Emit when trip remaining time updated. It calls appropriate slot in <code>MapViewController</code> to update total trip time label in <code>ManeuverDetailController</code> & <code>ManeuverListController</code>
2	<code>OnTripRemainingDelay(quint32 time)</code>	Emit when trip remaining delay updated
3	<code>OnTripRemainingDistance(const qreal& distance)</code>	Emit when trip remaining distance updated. It calls appropriate slot in <code>MapViewController</code> to update distance in <code>ManeuverDetailController</code> & <code>ManeuverListController</code> .
4	<code>OnCurrentRoadName(const QString& primaryName, const QString& secondaryName)</code>	Emit when the name of current road updated. Used to update current road name label in <code>ManeuverDetailController</code> & <code>ManeuverListController</code> .
5	<code>OnManeuverExitNumber(const QString& exitNumber)</code>	Emit when the exit number of current maneuver updated.
6	<code>OnNextRoadName(const QString& primaryName, const QString& secondaryName)</code>	Emit when the name of next road updated. Used to update next road name label in <code>ManeuverDetailController</code> .
7	<code>OnManeuverType(const QString& Type)</code>	Emit when the type of current updated.
8	<code>OnManeuverRemainingTime(quint32 time)</code>	Emit when remaining time of maneuver updated. Used to update current maneuver remaining time label in <code>ManeuverDetailController</code> .
9	<code>OnManeuverRemainingDelay(quint32 delay)</code>	Emit when the remaining time delay of current maneuver updated.

10	<code>OnManeuverRemainingDistance(const qreal& distance)</code>	Emit when the remaining distance of the current maneuver updated. Used to update current maneuver remaining distance label in <code>ManeuverDetailController</code> .
11	<code>OnManeuverImageId(const QString& imageId)</code>	Emit when maneuver image ID returned. Use <code>MapUtils</code> for getting bitmap.
12	<code>OnStackTurnImageTTF(const QString& stackImageTTF)</code>	Emit when the stack maneuver of turn image text returned. Used to update direction image in <code>ManeuverDetailController</code> .
13	<code>OnPositionUpdated(const Coordinates& coordinates, qint32 speed, qint32 heading)</code>	Emit when position on route has been changed.
14	<code>OnManeuverPoint(const Coordinates& point)</code>	Emit when the turn point coordinate of maneuver changed.
15	<code>OnUpdateManeuverList(const ManeuverList& maneuvers)</code>	Emit when updating upcoming maneuver list when maneuver changed.

Slots assigned for Traffic and announcement signal are:

No.	Slot	Description
1	<code>OnTrafficChanged(const TrafficInformation& trafficInfo)</code>	Emit when traffic information is updated
2	<code>OnDisableTrafficAlerted()</code>	Emit when disable the traffic alert.
3	<code>OnSpeedLimit(const SpeedLimitInformation& speedInfo)</code>	Emit when Speed Limit Updated.
4	<code>OnLaneInformation(const locationtoolkit::LaneInformation& laneInfo)</code>	Emit when Lane Information Updated.
5	<code>OnTripTrafficColor(char color)</code>	Emit when Traffic color updated.
6	<code>OnRoadSign(const locationtoolkit::RoadSign& roadSign)</code>	Emit when Road Sign updated.

7	OnAnnounce (Announcement *announcement)	Provides the announcement related information output stream to get audio data to be played audio text for TTS support
---	---	---

Maneuver List

While onNavigation > click Menu button on top right corner, Complete Maneuver List of current navigation route is displayed with Destination name, time remaining and distance remaining on header and direction & placeName list.

mManeuversListController object of ManeuversListController class initialized in function MapViewController::setupManeuverListSignalAndSlots(). In ManeuversListController::setup(), ManeuverListView.qml is created that has UI for Maneuver List.

Slots for mManeuversListController are connected in MapViewController::setupManeuverListSignalAndSlots():

No.	Slot	Description
1	OnTripRemainingTime (quint32 time)	Emit when trip remaining time updated. It calls appropriate slot in MapViewController to update total trip time label in ManeuverDetailController & ManeuverListController
2	OnTripRemainingDistance (qreal)	Emit when trip remaining distance updated. It calls appropriate slot in MapViewController to update distance in ManeuverDetailController & ManeuverListController.

Console 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>"
}
```


QT Cluster Plugin Implementation

This plugin supports features of a map integrated with navigation services. This plugin exposes the custom qml type **ClusterView**. Properties like zoom, tilt and follow me 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 displays the current Speed limit.

Display Map

MapKit3d library used for displaying map on home screen. **mLTKMapKit** object of **MapViewWidget** class is initialized in **MapView::init()** to display map. Slot pertaining to map is in the **init()** method.

No.	Slot	Description
1	onMapReady()	Here map's zoom, tilt, reference center, avatar position, setnightmode option and map decoration parameters like zoom, follow me and compass button are set.

Slots for low fuel alert:

No	Slot	Description
1	onLowFuelAlert()	Signal lowFuelLimit is emitted from ConfirmationBox.qml that is connected to mapview. Here showFuelAlert API is exposed on Dbus.

Cluster 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 communications Protocol

D-Bus is used as the interface for communication from Console to Cluster. Any module that registers for these interfaces will be notified of that information.

No.	Slots	Description
1	<pre>Q_SCRIPTABLE QString onManeuverUpdated(const QString& maneuverIcon, const QString& streetName, const QString& distance, const QString& trafficWarning);</pre>	Notified when there is update for current Maneuver during navigation.
2	<pre>Q_SCRIPTABLE QString updateRemainingManeuverDistance(co nst QString& distance);</pre>	Notified when there is update for current Maneuver distance during navigation.
3	<pre>Q_SCRIPTABLE QString updateGpsPosition(const QString& lat, const QString& lon, const QString& heading);</pre>	Notified when there is update in gps position.
4	<pre>Q_SCRIPTABLE QString setNavigationMode(bool navMode); Q_SCRIPTABLE bool onUpdateSpeedLimit(const QString& speedLimit)</pre>	<p>This notifies about an active navigation session.</p> <p>This notifies about speed limit during navigation</p>
5	<pre>Q_SCRIPTABLE bool updatePolyline (const QString &polyline)</pre>	This notifies about polyline update during navigation.
6	<pre>Q_SCRIPTABLE bool clearPolyline ()</pre>	This notifies to clear polyline while navigation end.

7	<code>Q_SCRIPTABLE bool updateManeuverArrow (const QString &pts)</code>	Notified when there is update for maneuver arrow on polyline during navigation.
----------	---	---

Cluster-Console communications

D-Bus is used as the interface for communication from the Cluster to Console. Any module that registers for these interfaces will be notified of that information.

1. When Fuel Alert is notified, **mAlert** object with option to navigate to nearest Fuel station is shown on console created in **MapViewController**.
2. Slot **onAlertBoxValueClicked ()** calls method **MapViewController::fuelAlertAcceptClicked ()** which emits **fuelSearchRequest (parameter)** that is connected to **CategorySearchController::onFuelSearchRequest (QVariant info)** where search request is raised for nearest fuel station.
3. When navigation to fuel station ends, navigationSession emits **RouteFinish ()** signal that is connected in **mapviewController::OnRouteFinish ()**, where **ConfirmBoxController** is called and user is asked to continue with previous destination. If user agrees, then **MapViewController::startNavigationWithNavKit ()** is called.

Following is the slot in Console:

No.	Slots	Description
1	<code>Q_SCRIPTABLE QString showFuelAlert(bool Mode)</code>	This notifies that fuel level is low.

For help and any questions, please contact support@location.studio.