

User's Guide



©2012–2015, QNX Software Systems Limited, a subsidiary of BlackBerry Limited.
All rights reserved.

QNX Software Systems Limited
1001 Farrar Road
Ottawa, Ontario
K2K 0B3
Canada

Voice: +1 613 591-0931
Fax: +1 613 591-3579
Email: info@qnx.com
Web: <http://www.qnx.com/>

QNX, QNX CAR, Momentics, Neutrino, and Aviage are trademarks of BlackBerry Limited, which are registered and/or used in certain jurisdictions, and used under license by QNX Software Systems Limited. All other trademarks belong to their respective owners.

Electronic edition published: Monday, June 29, 2015

Table of Contents

About This Guide	5
Typographical conventions	6
Technical support	8
Chapter 1: Getting Ready	9
Chapter 2: A Guided Tour of the HMI	11
Home	14
Navigation	16
Media Player	17
Play Radio	17
Play Music	18
Play Video	19
Search	20
Media source menu	21
Using voice commands to play media	23
Car Control	24
Audio Control	24
Climate Control	25
Virtual Mechanic	25
Personalization	28
Communication	31
Apps Section	32
Push-to-Talk	36
Chapter 3: Running the HTML5 HMI	37
Chapter 4: Automatic Speech Recognition (ASR)	41
Supported voice commands	43
Using mm-control to process voice commands	50
Chapter 5: Bluetooth: Pairing Devices, Calling, and Messaging	51
Pairing a smartphone	52
Making a phone call	54
Viewing the phone's email	56
Viewing the phone's contacts	57
Viewing the phone's text messages	59
Chapter 6: Network Settings	61

Chapter 7: Rearview Camera	65
Chapter 8: Pandora Radio	67
Chapter 9: Recalibrating the Touchscreen	69
Chapter 10: Software Updates	71
Applying an update from the HMI	72
Applying an update from the command line	74
Chapter 11: Downloading Apps from the QNX App Portal	75

About This Guide

The *User's Guide* provides a tour of the QNX CAR platform's HMI and describes how to use its key features. All application developers should read this guide.

The following table may help you find information quickly:

To find out about:	See:
Installing software and connecting your target	Getting Ready (p. 9)
All the HMI applications	A Guided Tour of the HMI (p. 11)
Switching from the default Qt5 HMI to the HTML5 HMI	Running the HTML5 HMI (p. 37)
Using speech commands	Automatic Speech Recognition (p. 41)
Pairing a smartphone with the car's head unit	Bluetooth: Pairing Devices, Calling, and Messaging (p. 51)
Configuring the Ethernet connection type, accessing a Wi-Fi network, and setting up a Wi-Fi hotspot	Network Settings (p. 61)
Applying software updates from the HMI or command line	Software Updates (p. 71)
New automotive-relevant HTML5 apps that you can try out on your system	Downloading Apps from the QNX App Portal (p. 75)

Typographical conventions

Throughout this manual, we use certain typographical conventions to distinguish technical terms. In general, the conventions we use conform to those found in IEEE POSIX publications.

The following table summarizes our conventions:

Reference	Example
Code examples	<code>if(stream == NULL)</code>
Command options	<code>-lR</code>
Commands	<code>make</code>
Constants	<code>NULL</code>
Data types	<code>unsigned short</code>
Environment variables	<i>PATH</i>
File and pathnames	<code>/dev/null</code>
Function names	<code>exit()</code>
Keyboard chords	Ctrl–Alt–Delete
Keyboard input	<code>Username</code>
Keyboard keys	Enter
Program output	<code>login:</code>
Variable names	<code>stdin</code>
Parameters	<code>parm1</code>
User-interface components	Navigator
Window title	Options

We use an arrow in directions for accessing menu items, like this:

You'll find the Other... menu item under **Perspective** → **Show View**.

We use notes, cautions, and warnings to highlight important messages:



Notes point out something important or useful.



Cautions tell you about commands or procedures that may have unwanted or undesirable side effects.



Warnings tell you about commands or procedures that could be dangerous to your files, your hardware, or even yourself.

Note to Windows users

In our documentation, we typically use a forward slash (/) as a delimiter in pathnames, including those pointing to Windows files. We also generally follow POSIX/UNIX filesystem conventions.

Technical support

Technical assistance is available for all supported products.

To obtain technical support for any QNX product, visit the Support area on our website (www.qnx.com). You'll find a wide range of support options, including community forums.

Chapter 1

Getting Ready

Before you can begin working with the QNX CAR platform, you'll need to install the development software on your host, connect the host to your target, and install the runtime software on the target.

Installation summary

Setting up your system involves performing these steps:

1. Set up your target hardware by connecting the power supply and touchscreen. For example, on the TI OMAP5432 EVM, connect one end of the HDMI cable to the touchscreen, then connect the other end (with the HDMI and USB connectors) to the board's Mini HDMI port and to one of its USB ports.
2. Use an Ethernet cable to connect the target so that it's reachable from your host.
3. Connect your host to the target for serial communications. For example, use a USB-to-Micro-USB cable from your host to the “DEBUG USB” port on the OMAP board.
4. Set up a serial communications tool (e.g., PuTTY) on your host using 11520-8-N-1 settings.
5. Install the QNX CAR image on your target. This involves creating a DOS partition on an SD card, copying the platform's system files to the SD card, inserting the SD card into the target, and then booting the target. You'll find complete instructions on doing this in the “Transfer an Image to Your Target” section of *Building and Customizing Target Images*.

Chapter 2

A Guided Tour of the HMI

The QNX CAR Platform for Infotainment 2.1 delivers mobile apps to the automobile with a powerful yet simple user experience.

The platform contains many apps that provide programming references to help you write new apps for your infotainment system. Some of these reference apps contain functional controls for vehicle settings, which automotive system integrators can use in their own software.

The platform includes two categories of applications:

Core applications

Accessible from the taskbar displayed along the bottom of the HMI, these applications provide controls to configure general subsystems such as the media player or climate controller. The core applications ensure that a set of standard features is available on all cars.

Add-on applications

Accessible through the **Apps Section** screen, these applications extend the capabilities of the platform by accessing topical information such as weather reports, social media feeds, or media repository sites. These apps can be downloaded from an app store or written and installed by customers or a third party. Sometimes, add-on applications are called *sandboxed apps* because they run as standalone processes to protect the rest of the HMI from any software faults.

HMI versions

This release of the product comes with two HMI versions:

- a Qt5 version, written with version 5.2 of the Qt framework
- an HTML5 version, written with the Sencha Touch and JQuery frameworks

Although these HMIs serve primarily as reference UIs, they do contain some active controls for configuring the car's climate, video playback, and other settings. The controls displayed the main HMI screens, the navigation between screens, and the mechanism for launching apps are all identical in the two HMI versions. The newer Qt5 HMI is a software port of the HTML5 HMI, so it matches the appearance and functionality of the HTML5 HMI.

By default, the Qt5 HMI is displayed, but you can reconfigure the system to launch the HTML5 HMI at startup (for details, see [Running the HTML5 HMI](#) (p. 37)).

UI theme

You can change the active UI theme under **Settings** → **Personalization** to give your system a different look and feel. In the Qt5 HMI, apps render their UIs by using controls in the Qt framework; the Personalization application also uses this framework to apply the active UI theme. In the HTML5 HMI, apps such as Media Player and Communication render their UIs by using CSS3. To apply the selected theme, the Personalization application uses **SCSS** to organize the CSS3 properties so it can reuse common styling rules.



The same themes are supported by both HMI versions.

Developers of third-party HTML5 apps may use any combination of SCSS and CSS3 styling, regardless of which HMI version is active.

For more information on the UI frameworks used by the HMI, see “HMI Layer” in the *Architecture Guide*.



The following video gives an overview of the UI mechanisms and key reference apps:

<https://www.youtube.com/watch?v=8Pdx79iybMI>

HMI screens

The following table lists the HMI screens that access the core applications. These screens are displayed by tapping the individual tabs on the taskbar. The rest of this chapter contains brief descriptions of each of these screens.

HMI screen	Description
Home	This screen, which is shown when the HMI starts up, displays a dashboard summarizing the current navigation, weather, recent messages, and any media that's playing.
Navigation	This screen accesses the active navigation engine. By default, the Elektrobit (EB) navigation engine runs when you select this tab. You can install different software and reconfigure your system to access another navigation engine in this screen.
Media Player	Contains the Play Radio , Play Music , Play Video , and Search controls, which allow you to play and search for media from different sources.

HMI screen	Description
Car Control	Displays the Audio Control , Climate Control , Virtual Mechanic , and Personalization controls, which manage settings related to the vehicle's environment, user profile, and audio volume.
Communication	Contains controls for communicating through a Bluetooth device (e.g., a smartphone) paired with the head unit. These controls are: Email Messages , Text Messages , Address Book , and Dial Pad .
Apps Section	Displays add-on apps, including BestParking, The Weather Network, Pandora Radio, and more. Here you'll also find a Settings app with controls for Bluetooth Connectivity, Software Updates, Wi-Fi hotspots, and Wired Network settings.
Push-to-Talk	Located in the lower-left corner of the HMI, this tab launches an automatic speech recognition (ASR) session for processing voice commands.

Home

The **Home** screen appears when the HMI starts up. This screen displays a dashboard summarizing current system activity and basic settings.

The dashboard contains a live view of any navigation in progress as well as any playing media, recent messages, and a live weather feed (hardcoded for Toronto, Ontario, Canada) from [The Weather Network](#). These information feeds are shown in separate view panes.



Taskbar and status bar

In the **Home** screen and all other HMI screens, the taskbar displayed along the bottom allows you to switch screens. The tab for the currently selected screen is always highlighted. The taskbar provides touch-sensitive and voice-activated tabs to navigate to the following screens:

- **Home**
- **Navigation**
- **Media Player**
- **Car Control**
- **Communication**
- **Apps Section**
- **Push-to-Talk**

The HMI also displays a status bar along the top. This bar shows the name of the active user, the time, the temperature inside the vehicle, and the audio volume setting.



Both the Qt5 and HTML5 versions of the HMI provide the same controls for switching screens. The controls shown in the screens appear and behave the same way in both versions.

Navigation

The QNX CAR platform includes the Elektrobit (EB) *street director* navigation software. The **Navigation** screen displays controls that access this software's navigation engine to perform route-based navigation to specific destinations.

Designed for automotive systems, the EB navigation software lets you search for a destination and, with one button tap, calculate the optimal route for reaching that destination and activate the navigation service to monitor your travel progress. As you drive, the map animates the travel progress in real time, while displaying the current street, next turn, and the estimated remaining time and distance to your destination.

The EB *street director* map gives a highly scalable view of your surroundings. You can zoom in (by tapping the + button) to a street-level view and see details of individual roads and buildings or zoom out (by tapping the - button) to a country- or even planet-wide view.



The **Home** screen displays turn and distance information while navigation is active.

In addition to the name-based destination search capability, EB *street director* can keep track of your favorite destinations and recent travels. This capability provides a fast way to look up and select one of your most frequently visited places as your next destination.

This release includes map data for the EB navigation engine. The map data contains location information for navigating to destinations within Canada as well as a World Cartographic Layer for drawing a world map when you zoom out into space.



You must contact Elektrobit to obtain map data for other countries.

Media Player

The **Media Player** screen displays the following controls for discovering and playing audio and video files on different sources: **Play Radio**, **Play Music**, **Play Video**, and **Search**.



You can play music and videos through the touchscreen interface or through voice commands.

Play Radio

The **Play Radio** control lets you select a radio app to play music.

The QNX CAR platform comes with a default radio app, which serves a UI reference. You can also integrate the [Pandora Radio](#) (p. 67) application with the **Play Radio** control. If Pandora Radio is set up, tapping the **PLAY RADIO** button in the **Media Player** screen or selecting the **Radio** option from the media source menu will make the HMI display a submenu with two options:

- **AM/FM**—tap to view the default radio
- **Pandora**—tap to view the version of Pandora integrated into the media player



The integrated version of Pandora looks and functions like the standalone version.

If Pandora isn't enabled, accessing the **Play Radio** control from the **Media Player** screen or the media source menu takes you directly to a screen that shows the default (reference) radio app:



This app is a simulation on the TI OMAP, TI Jacinto 5, and Freescale i.MX6q boards, which have no radio hardware.

The Radio app has been developed as a Sencha Touch application. The app has two profiles:

- **high**—presents advanced animation and a graphically rich skin
- **mid**—has a more basic look and feel

On the TI OMAP and Freescale i.MX6q boards, the default profile for the Radio app is **high**. On TI Jacinto 5, **mid** is the default profile. You can configure the app to use either **high** or **mid**. The radio profile is defined by a PPS object. For more information about setting the profile type, see the `/pps/qnxcar/system/settings` entry in the *PPS Objects Reference*.

All the controls for the radio application, including the slider, dial, AM, FM, Seek, Scan, and presets are functional. But again, the radio itself is only a simulation on the supported hardware boards.

Play Music

The **Play Music** control shows a Juke Box that lets you visually browse media.

From the **Media Player** screen, tap the **PLAY MUSIC** button to open the Music Player:



The Juke Box database is loaded with several music files that the HMI shows in an active carousel. Using voice or touch controls, you can start, pause, or skip tracks (shuffle and repeat are not available as voice commands). You can use voice commands to play media by the named song, album, or artist. For more information about voice commands, see “[Using voice commands](#) (p. 23)”.

Play Video

The **Play Video** control lets you access the video player. Depending on your target hardware, you may need to copy certain codecs onto your system to enable video playback.

From the **Media Player** screen, tap the **PLAY VIDEO** control to open the video player. This release includes the following video:

“The Making of the QNX Reference Vehicle: Jeep Wrangler”

The **Play**, **Pause**, **Forward**, and **Backward** controls are functional. Note that the **Full Screen** control is functional in the Qt5 HMI but not in the HTML5 HMI.





Video playback isn't enabled on the image for the Freescale i.MX6q SABRE Lite platform. To play videos on this hardware, you must obtain the necessary codecs and copy them onto the target board. The images for the TI OMAP uEVM and TI Jacinto 5 Eco platforms include the codecs necessary for playing videos.

Enabling video playback on the i.MX6q platform

To enable video playback on i.MX6q:

1. Obtain the Freescale video components from the [QNX Download Center](#):

Video components for Freescale silicon

2. Ensure your i.MX6q target board has an active Internet connection.
3. Obtain your target's IP address by running the `ifconfig` command.
4. Mount the `/base` directory as follows:

```
mount -uw /base
```
5. Create the directory `/lib/firmware/vpu/`.
6. Place the VPU codec binary (`vpu_fw_imx6q.bin`) in this newly created directory as follows:

```
scp path_to_downloaded_codec_binary  
root@target_IP_addr:/lib/firmware/vpu/
```

Note that the password is `root`.

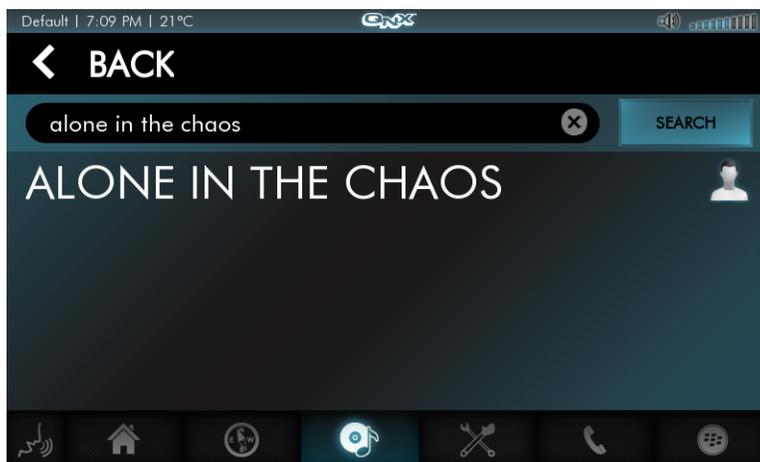
7. Restart your system using the `reboot` command.

Video playback is now enabled.

Search

The **Search** control lets you find songs or videos on a specific media device.

From the **Media Player** screen, tap the **SEARCH** button to access the search feature. The HMI shows a list of search sources, such as iPod, USB drive, or Juke Box. Tap the entry for the device you want to search. The HMI then shows a text field that lets you enter the name of an artist, album, or track.



Media source menu

A menu of media sources is accessible at the top of each of the **Play Radio**, **Play Music**, and **Play Video** screens. This menu makes it easy to switch between playing music and videos from different sources.

The media sources can include:

- Radio
- Apple iPod (if connected)
- USB drive (if connected)
- Digital Media Source (DMS) devices (if DLNA is enabled)
- Juke Box database

In the media source menu, you can:

- Tap **Radio** to access the radio reference UI (or Pandora if it's set up).
- Tap **iPod**, **USB drive**, the entry for any local DMS device, or **Juke Box** to see a list of music media sorted by playlists, artists, albums, songs, and genres.

To return to the previous screen, tap the **Menu** control at the bottom of the screen.

The **Home** screen shows information about the selected song if the Music Player is active.

DLNA support

The media player supports Digital Living Network Alliance (DLNA) technology using the Twonky DMS. When DLNA is enabled on your target, any local DMS devices will appear in the list of available media sources.

All playback controls, including **Skip** and **Seek**, are functional for DLNA devices.

Starting DLNA

DLNA isn't set up to start by default. To start DLNA services during bootup, you must do two steps:

1. Edit `/var/etc/services-enabled` and set the `DLNA` configuration option to `true`:

```
DLNA:true
```

2. Edit `/etc/mm-player.cfg` and add an entry for the DLNA plugin in the `plugins` object. The exact `mm-player` plugins listed in this configuration file depend on your system setup but in general, you should change something like this:

```
"plugins":{
  "mpp-avrcp.so":{
    "mode":"device"
  }
}
```

to:

```
"plugins":{
  "mpp-avrcp.so":{
    "mode":"device"
  },
  "mpp-dlna.so":{
    "mode":"player"
  }
}
```



You must reboot your target for these changes to take effect.

Verifying that DLNA services are running

Use the following commands to verify that the DLNA services are running. The `pidin` utility will return the PID of each service.

```
pidin ar | grep "dmc"
pidin ar | grep "dmr"
```

Tested hardware

The DLNA implementation was tested with the following hardware setup:

Hardware	Description
Target boards	TI OMAP5432 EVM, TI Jacinto 5 ECO, and Freescale i.MX6q SABRE Lite.
Router	Cisco Linksys EA2700 wireless router. The target board was connected to the router by a Category 6 Ethernet cable.

Hardware	Description
Smartphone	A Samsung Galaxy S II (model GT-19100) running Android version 2.3.3 software was used to run the DLNA server software (Twonky DMS). The device was connected by Wi-Fi to the Cisco Linksys wireless router.

For the DMS in this setup, the TwonkyServer Mobile v 2.3 DMS for Android was used.



Currently, the MP3 audio format is the only supported and tested digital media type.

Using voice commands to play media

You can use voice commands to start, pause, and skip tracks and to select a song by artist, album, or title.

Note that the current Automatic Speech Recognition (ASR) implementation has these limitations:

- Supported commands are “Pause”, “Resume”, “Next”, and “Previous”. The “Search” command is *not* supported.
- If you're using `mm-control` as the multimedia player, commands such as “Play David Hasselhoff” or “Play album David Hasselhoff Sings America” will cause the player to create a new tracksession matching the search terms, then start playback.
- The `mm-player` multimedia player doesn't support the “Play” command; if it receives this voice command, it will respond that it doesn't understand the request.

For more information about using ASR, see “[Automatic Speech Recognition](#) (p. 41)” in this guide.

Car Control

The **Car Control** screen displays the following controls for managing the vehicle's environment: **Audio Control**, **Climate Control**, **Virtual Mechanic**, and **Personalization**.



The HMI app displayed in this screen is written with Qt, for both the HTML5 and Qt5 versions of the HMI.

Audio Control

The **Audio Control** feature controls the volume of the music playing in the Music Player. From the **Car Control** screen, tap the **AUDIO CONTROL** button to open a screen with volume controls:



The **Audio Control** feature is the only place in the HMI where you can set the volume—there's no separate app for controlling the volume of audio playback. The **Bass**, **Treble**, and balance controls are active but not functional.

Climate Control

The **Climate Control** feature lets you manage the climate settings in the car.

From the **Car Control** screen, tap the **CLIMATE CONTROL** button to open a screen that lets you control the in-car climate settings:



This reference UI contains the following active controls for both the driver and passenger sides of a vehicle:

- Air distribution mode
- Temperature
- Fan speed
- Heated seats

Additional buttons provide toggle controls for the following options:

- Air recirculation
- Defrost
- Air conditioning
- Synchronizing driver and passenger climate settings (the **ALL** button)

These controls are functional in that they update PPS objects that can be read by a climate control system in a car.

Virtual Mechanic

The **Virtual Mechanic** feature lets you see basic information about the vehicle's fluid levels, brakes, the electrical system, and more.

From the **Car Control** screen, tap the **VIRTUAL MECHANIC** button to view mechanical information about the car:



The initial screen shows a fuel gauge and provides buttons to open dialog boxes that give information about vehicle statistics and the following automotive systems:

- Fluid levels
- Traction system
- Braking system
- Powertrain system
- Electrical system

Fluid Levels

This dialog box demonstrates a UI that shows levels and warning indicators for the following vehicle reservoirs:

- Windscreen washer
- Transmission fluid
- Fuel
- Engine coolant
- Brake fluid

Traction System

The Traction System dialog shows warning indicators for pressure and wear on each tire.

Braking System

In the Braking System dialog, the **ABS BRAKES** toggle control is active but nonfunctional. The dialog also shows visual warning indicators for brake pads and ABS sensors for each tire.

Powertrain System

This dialog indicates the currently selected gear and shows the condition of the following engine and transmission components:

- Engine oil pressure
- Engine oil level
- Engine RPM
- Transmission temperature
- Clutch wear
- Next service date

Electrical System

The Electrical System dialog shows warning indicators for the headlights and rear lights.

Vehicle Statistics

The Vehicle Statistics dialog shows the following test data:

- Departure time
- Trip duration
- Mileage
- Distance traveled
- Average speed



When you tap the **RESET** buttons, the data is reset to zero for each measurement except DEPARTURE TIME—this value is reset to the current system time.

Personalization

The **Personalization** app provides a mockup of customized vehicle and user-profile settings.

From the **Car Control** screen, tap the **PERSONALIZATION** button to open the Personalization app, which looks like this:



You can customize the settings by:

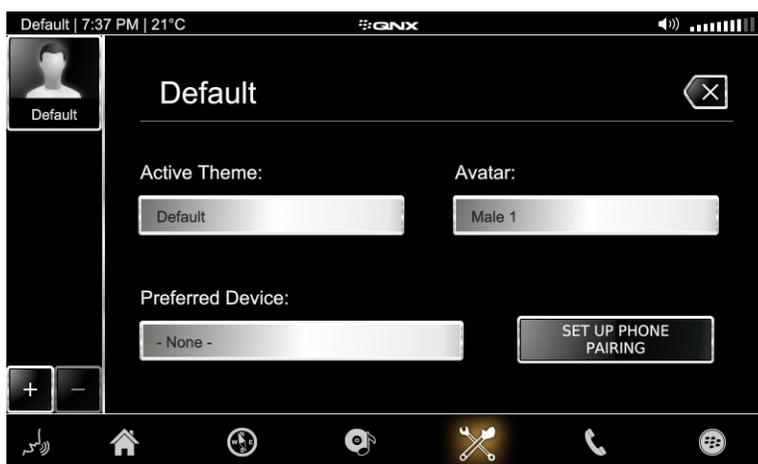
- Changing the profile name—tap the text field at the top, then use the virtual keyboard to enter a new display name for the profile.
- Changing the graphical appearance or *theme* of the UI—tap the **Active Theme** text field and select a new theme. For information on how applying and customizing themes, see “[Changing themes](#) (p. 29)”.
- Selecting an avatar—tap the **Avatar** text field and select an avatar. Two male and two female avatars are available. The thumbnail photo in the top-left corner of the screen changes to a photo of the new avatar.
- Choosing the preferred device—tap the **Preferred Device** text field to display a list of available Bluetooth devices; you can select one as the preferred device. You can return to the main application screen by swiping right from the left border of the device list display.
- Setting up Bluetooth pairing—tap the **SET UP PHONE PAIRING** button to jump to the Bluetooth Connectivity area of the Settings app. In this other app, you can pair the system with a suitable Bluetooth device. This paired device will then appear in the list of Bluetooth devices when you return to the **Personalization** screen and tap the **Preferred Device** text field.

Changing themes

The *theme* changes the graphical appearance of the HMI to give it a certain look and feel. The theme controls the colouring of visual components, image swaps, font sizes and styles, and background patterns.

To customize themes, you can use the PPS theme object to modify the attributes of the Personalization theme packages available on the CAR platform. This object is updated to notify the system whenever a new theme is added or removed. The following themes are currently available:

- **default**



- **midnightblue**



- **titanium**



All styles and image assets related to personalization are defined in files located in `/usr/hmi/common/themes/`.

Two PPS objects exist for themes:

`/pps/qnxcar/themes`

Lets you view the attributes of Personalization theme packages.

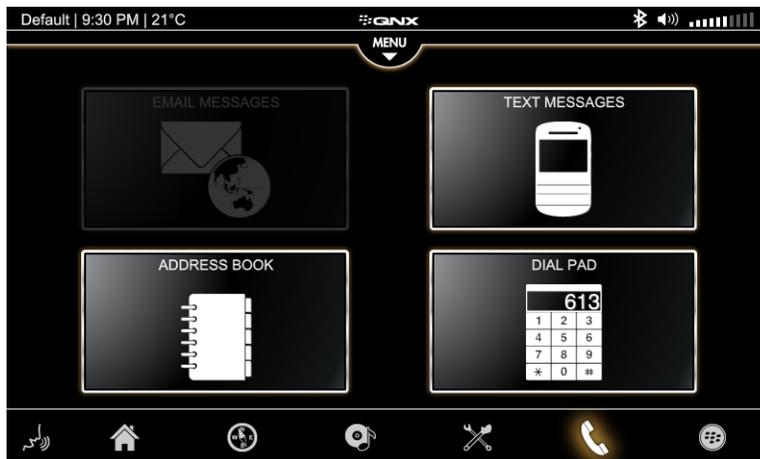
`/pps/qnxcar/profile/theme`

Lets you view or modify the settings for the Personalization **Theme** field. You can set the *theme* attribute to one of the available themes to give the desired appearance for the HMI apps. The *theme* attribute is controlled by the **Theme** field in the Personalization application.

Communication

The **Communication** screen accesses the following features for communicating through a paired Bluetooth device: **Email Messages**, **Text Messages**, **Address Book**, and **Dial Pad**.

All four controls become functional when a Bluetooth device such as a smartphone is paired with the head unit. Note that the device must have an SMS account; otherwise, the controls in the **Communication** screen remain disabled after pairing.



Each control accesses a different Bluetooth service. For details on how to use these communication services, see [Bluetooth: Pairing Devices, Calling, and Messaging](#) (p. 51).

Apps Section

The **Apps Section** screen shows icons for add-on apps, including BestParking, The Weather Network, Pandora Radio, and more. There's also the Settings app, which provides controls for Bluetooth connectivity, software updates, and wired and wireless network settings.

The **ALL** category in the **Apps Section** screen shows all the apps installed on your platform, including any new apps you install from the [QNX App Portal](#) (p. 75). These add-on apps run independently of the HMI, so you can launch HTML5 apps when running the Qt5 version of the HMI, and vice versa.



We have included these third-party apps in the QNX CAR platform's evaluation images. If you build your own target image manually, you won't find these apps on your target. However, you can download them via the **QNX App Portal**. For instructions, see "[Downloading Apps from the QNX App Portal](#) (p. 75)" in this guide.

After installing the QNX CAR platform, you should see icons for the following apps under **ALL**:

API Demos

An app that demonstrates some capabilities of APIs for the Runtime for APK.

BestParking

Compares parking rates so you can find the best deals.

Browser

Web browser app that uses the WebLauncher rendering engine to support HTML5.

Calculator

A calculator app for the Runtime for APK.

eventseeker

Discovers bands, festivals, and events in any city.



You may need to scroll to see the eventseeker app. To scroll, touch the side of the screen and drag your finger up.

Local Search

Uses the *Google Maps* service for POI searches and for Weather.

Pandora

Third-party Internet radio service. You can start a standalone version of this app from the **Apps Section** screen but another version is integrated into the Media Player (see “[Pandora Radio](#) (p. 67)” for more information).



Pandora isn't available in the Qt version of the HMI.

Parkopedia

Provides community-based parking information.

PeaksAndValleys

WebGL demo game that provides a 3D animation of mountainous terrain. The user is given camera control to scroll the image and simulate movement.

QNX App Portal

Allows developers to showcase their HTML5 apps for automakers to evaluate (see “[Downloading Apps from the QNX App Portal](#) (p. 75)” for more information).

Rearview Camera

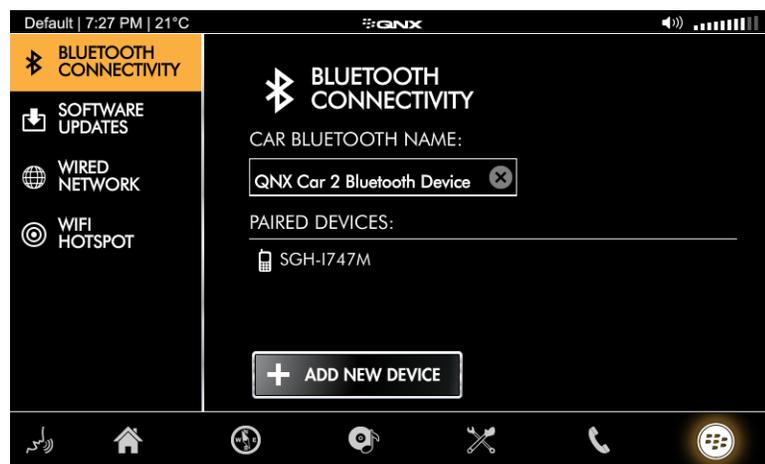
Displays the video feed from a rearview camera (if this is supported by the hardware). Tapping the **Apps Section** button will then return to the previous display.



Settings

Configures device settings for:

- Bluetooth Connectivity—lets you pair or unpair Bluetooth devices. Any device pairing will persist across reboots.
- Software Updates—detects any pending updates by reading a delta file residing on a USB stick. You can start applying the update with one button tap.
- Wired Network—lets you configure your IP address settings.
- WiFi Hotspot—lets you configure your system as a hotspot.



Shutdown

Provides an HMI control for gracefully rebooting the system. Tapping one button invokes the `reboot` script, which backs up the personalization database, shuts down PPS, remounts the user data and DOS filesystems as read-only to flush any changes to disk, and restarts the system.



We recommend always running the `reboot` script—either by using the Shutdown app or by manually invoking `reboot` at the command line—to shut down or restart your system. Disconnecting and reconnecting the power supply can corrupt files and leave your system in an unpredictable state.

Slacker Radio

Internet radio service for listening to custom music, news, sports, and comedy stations for free. Subscribers have on-demand access to specific songs, albums, and playlists.

TunnelTilt

Another WebGL demo game. This app simulates movement through a tilting tunnel.

WeatherNetwork

Links to [The Weather Network](#). The WeatherNetwork app uses geolocation to set the current location. If the correct location can't be determined, the default location of Oakville, Ontario, Canada is used.



Since The Weather Network uses SSL to securely access websites it may need, you must make sure that the system clock on your platform is set during startup, before the HMI loads. Otherwise, the app may fail. For more information, see “Realtime Clock Synchronization” in the *System Services Reference*.

Push-to-Talk

The **Push-to-Talk** screen indicates that the system is actively processing speech commands, which give users a hands-free way to perform a supported set of voice-activated tasks.

Unlike the other main screens in the HMI, the **Push-to-Talk** screen doesn't display a group of controls. Tapping its tab starts an ASR session for processing voice commands.

When you tap the **Push-to-Talk** tab, an overlay graphic appears on the top half of the display, indicating that the system is listening for a voice command. When no command is given after a timeout period, the overlay disappears.



For more information on ASR commands, see “[Automatic Speech Recognition](#) (p. 41)” in this guide.

Chapter 3

Running the HTML5 HMI

You can reconfigure your system to launch the HTML5 HMI instead of the default Qt5 HMI at startup.

To run the HTML5 HMI at startup:

1. In a QNX terminal on the target board, mount the local filesystem with read-write permissions by entering this command:

```
mount -uw /base
```

2. In a text editor, open the System Launch and Monitor (SLM) configuration file (/base/etc/slm-config-all.xml).

- a. Enable the SLM component for the HTML5 HMI by changing this section:

```
<!-- Uncomment the section below to enable automatic startup of the HTML5 HMI and
comment the Qt5 section which follows -->
<!--
  <SLM:component name="hmi">
    <SLM:command>/usr/bin/weblauncher</SLM:command>
    <SLM:args>webkit-launcher
/accounts/1000/appdata/car.navigators/app/native/wwe</SLM:args>
    <SLM:stop stop="signal">SIGTERM</SLM:stop>
    <SLM:envvar>HOME=/accounts/1000/appdata/car.navigators/data</SLM:envvar>
    <SLM:depend>symlink_hmi</SLM:depend>
    <SLM:depend>symlink_base</SLM:depend>
    <SLM:depend>iopkt</SLM:depend>
    <SLM:depend>pps</SLM:depend>
    <SLM:depend>calib-done</SLM:depend>
    <SLM:depend>screen-ready</SLM:depend>
  </SLM:component>
-->
<!-- End uncomment for HTML5 HMI -->
```

to this:

```
<!-- Uncomment the section below to enable automatic startup of the HTML5 HMI and
comment the Qt5 section which follows -->
<!-- -->
  <SLM:component name="hmi">
    <SLM:command>/usr/bin/weblauncher</SLM:command>
    <SLM:args>webkit-launcher
/accounts/1000/appdata/car.navigators/app/native/wwe</SLM:args>
    <SLM:stop stop="signal">SIGTERM</SLM:stop>
    <SLM:envvar>HOME=/accounts/1000/appdata/car.navigators/data</SLM:envvar>
    <SLM:depend>symlink_hmi</SLM:depend>
```

```

    <SLM:depend>symlink_base</SLM:depend>
    <SLM:depend>iopkt</SLM:depend>
    <SLM:depend>pps</SLM:depend>
    <SLM:depend>calib-done</SLM:depend>
    <SLM:depend>screen-ready</SLM:depend>
    </SLM:component>
<!-- -->
<!-- End uncomment for HTML5 HMI -->

```

b. Disable the SLM component for the Qt5 HMI by changing this section:

```

<!-- Comment the section below to disable automatic startup of the Qt5 HMI -->
<!-- -->
    <SLM:component name="hmi">
        <SLM:command>/qtcars/bin/qnxcar2</SLM:command>
        <SLM:stop stop="signal">SIGTERM</SLM:stop>
        <SLM:envvar>HOME=/accounts/1000/appdata/car.navigators/data</SLM:envvar>
        <SLM:envvar>QONX_PHYSICAL_SCREEN_SIZE=150,90</SLM:envvar>
        <SLM:envvar>QONXCAR2_ASSETS_DIR=/qtcars/share/qnxcar2</SLM:envvar>
        <SLM:stdout>/var/log/qnxcar2-qt.stdout</SLM:stdout>
        <SLM:stderr>/var/log/qnxcar2-qt.stderr</SLM:stderr>
    <SLM:depend>symlink_hmi</SLM:depend>
    <SLM:depend>symlink_base</SLM:depend>
    <SLM:depend>iopkt</SLM:depend>
    <SLM:depend>pps</SLM:depend>
    <SLM:depend>calib-done</SLM:depend>
    <SLM:depend>screen-ready</SLM:depend>
    </SLM:component>
<!-- -->
<!-- End comment for Qt5 HMI -->

```

to this:

```

<!-- Comment the section below to disable automatic startup of the Qt5 HMI -->
<!--
    <SLM:component name="hmi">
        <SLM:command>/qtcars/bin/qnxcar2</SLM:command>
        <SLM:stop stop="signal">SIGTERM</SLM:stop>
        <SLM:envvar>HOME=/accounts/1000/appdata/car.navigators/data</SLM:envvar>
        <SLM:envvar>QONX_PHYSICAL_SCREEN_SIZE=150,90</SLM:envvar>
        <SLM:envvar>QONXCAR2_ASSETS_DIR=/qtcars/share/qnxcar2</SLM:envvar>
        <SLM:stdout>/var/log/qnxcar2-qt.stdout</SLM:stdout>
        <SLM:stderr>/var/log/qnxcar2-qt.stderr</SLM:stderr>
    <SLM:depend>symlink_hmi</SLM:depend>
    <SLM:depend>symlink_base</SLM:depend>
    <SLM:depend>iopkt</SLM:depend>
    <SLM:depend>pps</SLM:depend>
    <SLM:depend>calib-done</SLM:depend>
    <SLM:depend>screen-ready</SLM:depend>
    </SLM:component>
-->
<!-- End comment for Qt5 HMI -->

```

3. Save the SLM configuration file and return to the QNX terminal.

When you're ready to run the HTML5 HMI, reboot the target board by issuing the `reboot` command.

The system reboots and launches the HTML5 HMI, which is identical in appearance and functionality to the Qt5 HMI.



To run the Qt5 HMI again, restore the SLM configuration file to its original state (with the HTML5 HMI component disabled and the Qt5 HMI component enabled), and then reboot again.

Chapter 4

Automatic Speech Recognition (ASR)

The platform includes an ASR subsystem that provides speech-recognition and text-to-speech services to other system components and third-party applications.

To start an ASR session, tap the **Push-to-talk** tab on the taskbar, then wait for the audible cue before you say a command. Shorter commands have lower success rates.

For a listing of commands you can use, see “[Supported voice commands](#) (p. 43)”.

For more information about using ASR for different tasks, see the task-specific pages (e.g., “[Media Player](#) (p. 17)”). For more information about the ASR modules, see the following:

- “Automatic speech recognition” in the *Architecture Guide*
- *Automatic Speech Recognition Developer's Guide*

ASR grammars

You can modify the grammars specified in the `/etc/asr-car.cfg` file to define keys (synonyms) for the supported speech commands. The grammars reside in the `car-control.cfg` files that are listed in the `localized-assets` section of `asr-car.cfg` for each module. For example, the grammar for the `car-media` module is located at `$(locale-dir)/car-media/car-control.cfg`.

Recognition latency

Several factors affect the latencies of voice-command recognition:

- **End of Speech (EOS) detection**—Too much ambient noise might prevent the ASR service from detecting EOS. In this case, the service uses the `max_utterance_seconds` setting to limit the audio capture. You can change this setting in the `/etc/asr-car.cfg` file.

To be certain that this is an issue, connect to the target and run `sloginfo -w` to determine when the audio capture completes. The time from completion of audio capture to response is the offboard recognition latency.

You can change EOS detection parameters in the `asr-car.cfg` file. To find the relevant settings, search for `eos`. Try to tune the EOS detector to perform better for your environment. You can adjust the `eos-*` values in steps of 50. Slay and restart `io-asr-generic` after each change, and then test how quickly the ASR service detects EOS. Note also that EOS detection is poor if the signal is not very dynamic. Before you start driving, make sure that the microphone has a fresh battery and that it's pointed in the right direction.

ASR server congestion

Server usage might be higher than usual. Run `sloginfo -w` to determine if the latency is on the recognition server.

Text-to-speech (TTS) latency

A slow TTS response can affect the perceived responsiveness of the system. The latency of the message that announces what is being done, or an unrecognized command, might appear to a user to be associated with a voice-recognition issue. In this case, the output of `sloginfo -w` should give you a good sense of the TTS latency as well. The service will log the message to be spoken before sending the request to the ASR service.

Determining unrecognized commands

If you say a command that the system doesn't recognize, there are a number of ways you can get more information about how the command was interpreted. You can:

- see the interpreted command on the screen
- examine the system log. To examine the system log and search for a particular command, run the following:

```
sloginfo -w | grep utterance
```

- examine the `/pps/services/asr/control` object to see what ASR understood and what intents it extracted from the command. For example, the command “Switch to media player” results in the following update to the PPS object:

```
@control
result:json:{"confidence":925,
             "recognizer":"io-asr-nlal",
             "status":"result_ok",
             "type":"intent",
             "action":"launch",
             "utterance":"Switch to media player",
             "intents":[{"field":"application","value":"media player"}]}
speech::handled
state::idle
strobe::on
```

Supported voice commands

The ASR service supports a set of keywords that you can use to initiate handsfree requests. In the sections that follow:

- Elements in square brackets are optional and can be used only once.
- Elements separated by pipe characters (“|”) represent choices. You can use one item from the set.

Before each command, you can use the following optional words as a preamble:

`[could | can | will | would] [you] please`

For example, “Could you please...”, “Will you...”, “Please...”, “Can you...”, and so on.

In the following sections, the supported commands are grouped according to commonly used tasks. Some of these tasks are handled locally by the system, whereas in other cases the commands are sent to offboard services. The following table lists the tasks and whether they are handled locally or offboard:

Task	Handling
Launch or terminate an application (p. 44)	Local
Get weather information (p. 44)	Offboard
Get the current time or date (p. 45)	Local
Find a nearby point of interest (POI) (p. 46)	Offboard
Get directions to an address (p. 46)	Offboard
Show a map of a city (p. 47)	Offboard
Search for media tracks or internet POIs (p. 47)	Media search is local for tracks in the Juke Box; otherwise, it's offboard. Internet POI search is offboard.
Make a phone call (p. 48)	Local
Play music by song title, artist, or album (p. 48)	Media playback is local for tracks in the Juke Box; otherwise, it's offboard.
Control the media player (p. 48)	Local
Get help (p. 48)	Local
Stop using the ASR service (p. 48)	Local

Launch or terminate an application

(<launch-cmd> | <terminate-cmd>) [<app-preamble>] <app-name> [application]

<launch-cmd> is one of:

- “open”
- “switch to”
- “go to”
- “launch”

<terminate-cmd> is one of:

- “close”
- “terminate”
- “kill”

<app-preamble> is optional and is one of:

- “the application” or “my application”
- “the app” or “my app”
- “the”
- “my”

<app-name> is the name of an HMI tab, a built-in app, or one of the apps in the HMI's [Apps Section](#) (p. 32). The ASR service's application list is built dynamically from the data in the `/pps/services/app-launcher` object. New applications that are added can't be launched by ASR until the system is rebooted.

You can optionally use the word “application” after the name of the application.

For example: “Please go to the video player”, “Launch radio”, “Switch to Home”, “Please go to the app BestParking”, “Would you launch my calculator application”, “Please close the BestParking application”, “Would you please terminate Pandora”.

Get weather information

[<acquire-cmd> | <inquire-cmd>] <conditions> [<location-preamble>] [<location>]

<acquire-cmd> is optional and is one of:

- “get”
- “what's”, “what is”, or “what are”
- “check what's”, “check what is”, or “check what are”
- “give”, “tell”, or “show”
- “give me”, “tell me”, or “show me”

optionally followed by “the”.

<inquire-cmd> is optional and is one of:

- “I need to know”, “I need to find”, or “I need to get”
- “I must know”, “I must find”, or “I must get”
- “I have to know”, “I have to find”, or “I have to get”

optionally followed by “the”.

`<conditions>` is one of:

- “weather”
- “temperature”
- “conditions” or “conditions are”

`<location-preamble>` is optional and is one of:

- “for”
- “in” or “over in”
- “like”, “like for”, “like in” or “like over in”

`<location>` is the optional location name. If the location isn't given, the weather conditions for the current location are retrieved.

For example: “What's the weather like in Maui”, “Get the weather in Sydney”, “What's the temperature like over in Vancouver”, “Tell me the conditions for Tokyo”, “I need to know the conditions in Portland, Oregon”, “I have to find the temperature in Calgary”.

Get the current time or date

`[<acquire-cmd>] [<time-date-preamble>] <time-date> [please]`

`<acquire-cmd>` is optional and is one of:

- “get”
- “what's”, “what is”, or “what are”
- “check what's”, “check what is”, or “check what are”
- “give”, “tell”, or “show”
- “give me”, “tell me”, or “show me”

optionally followed by an article (“a”, “an”, or “the”).

`<time-date-preamble>` is optional and is one of:

- “current”
- “today's”

`<time-date>` is one of:

- “time”, “time is”, “time is it”, “time it is”
- “time now”, “time is now”, “time is it now”, “time it is now”

- “date”, “date is”, “date is it”, “date it is”
- “day”, “day is”, “day is it”, “day it is”
- “date today”, “date is today”, “date is it today”, “date it is today”
- “day today”, “day is today”, “day is it today”, “day it is today”

You can optionally use the word “please” after the command.

For example: “Please get today's date”, “What's the time now please”, “Give me today's date”, “What day is it”, “Tell me what time it is now please”.

Find a nearby point of interest (POI)

A POI search using the navigation engine can take two forms:

```
( find | locate) <proximity-modifier> <destination>
```

or:

```
<navigate-cmd> <proximity-modifier> <destination>
```

<navigate-cmd> is one of:

- “get”
- “what's”, “what is”, or “what are”
- “check what's”, “check what is”, or “check what are”
- “give”, “tell”, or “show”
- “give me”, “tell me”, or “show me”

followed by the word “directions” and the optional word “to”.

<proximity-modifier> is optional and is one of:

- “nearby” or “a nearby”
- “nearest” or “the nearest”

Note that the modifiers “closest” and “close by” aren't supported.

<destination> is the name of a POI.

Note that POI search is relative to your current location, so searching for points of interest from other cities or countries may not yield any results.

For example: “Locate the CN Tower”, “Find a nearby post office”, “Please show me directions to a nearby Starbucks”, “Tell me directions to the BrookStreet Hotel”.

Get directions to an address

```
( navigate to ) <nav-address>
```

<nav-address> is the destination address, consisting of a numeric-only building number and a street name (optionally including a direction). Note that unit numbers and letters aren't supported (e.g., instead of “221 B Baker Street”, say “221 Baker Street”).

For example: “Navigate to 691 Somerset Street West”.

Show a map of a city

[<acquire-cmd>] map [<preposition>] <location>

<acquire-cmd> is optional and is one of:

- “get”
- “what's”, “what is”, or “what are”
- “check what's”, “check what is”, or “check what are”
- “give”, “tell”, or “show”
- “give me”, “tell me”, or “show me”

optionally followed by an article (“a”, “an”, or “the”).

<preposition> is optional and is one of:

- “of”
- “on”
- “to”
- “in”

For example: “Show me a map of Toronto”, “Please get a map of Rio de Janeiro”.

Search for media tracks or internet POIs

A media or internet POI search can take two forms:

(search) [<search-type>] [for] <search-term>

or:

<search-request> <search-term>

<search-type> is optional and is one of:

- “media”, “music”, or “songs”
- “Bing”, “Google”, “the internet”, or “the web”

<search-request> is one of:

- “get information” or “get information on”
- “give me information on”, “tell me about”, or “show me information on”
- “I'm interested in”, or “I am interested in”

Note that POI search is relative to your current location, so searching for points of interest from other cities or countries may not yield any results.

For example: “Search media for Weezer”, “Get information on the parliament buildings”, “Search the internet for bowling”, “Search music for Synchronicity”.

Make a phone call

(dial | call) <contact>

<contact> is one of:

- the phone number to dial
- a contact name in your phonebook

For example: “Dial 555-1212”, “Call Emily”.

Play music by song title, artist, or album

(play) [<media-category>] <media-name>

<media-category> is optional and is one of:

- “song” or “track”
- “the artist”, “songs by”, or “tracks by”
- “the album”, “songs from the album”, or “tracks from the album”

For example: “Play Bamboleo”, “Play tracks by Johnny Cash”, “Play songs from the album Thriller”.

Control the media player

To control the media player, use one of the following commands:

- “previous track” or “previous song”
- “next track” or “next song”
- “pause playback”, “pause media”, or “pause music”
- “resume playback”, “resume media”, or “resume music”

Get help

To get help with the ASR service, use one of the following commands:

- “help”
- “what can I say”
- “I need help”
- “help me”

Stop using the ASR service

To stop the ASR service, use one of the following commands:

- “quit”
- “cancel”
- “exit”

- “end”

Using mm-control to process voice commands

To support voice commands in legacy applications that use `mm-control` to perform media operations, your QNX CAR image must load a nondefault version of the **CarMedia** module that works with `mm-control`.

There are different versions of the **CarMedia** conversation module (or plugin) for each of the `mm-player` and `mm-control` media services shipped with the platform. By default, the ASR subsystem is configured to use the version that works with `mm-player`; this setup works with both the Qt5 and HTML5 HMIs. However, if you want to support media-related voice commands in applications that depend on `mm-control`, you must use the plugin version designed for this alternative service.

*To use the `mm-control` version of the **CarMedia** conversation module:*

1. From a QNX terminal on the target board, open the ASR configuration file (`/etc/asr-car.cfg`) in an editor.
2. In the `load-modules` section, disable the conversation module for `mm-player` by changing:

```
# add module to control media playback and perform
# media queries using the mm-player back-end.
dll = $(dlldir)/asr-mm-player-car-media-conversation.so
```

to:

```
# add module to control media playback and perform
# media queries using the mm-player back-end.
#dll = $(dlldir)/asr-mm-player-car-media-conversation.so
```

3. Enable the conversation module for `mm-control` by changing:

```
# add module to control media playback and perform
# media queries using the mm-control back-end.
#dll = $(dlldir)/asr-mm-control-car-media-conversation.so
```

to:

```
# add module to control media playback and perform
# media queries using the mm-control back-end.
dll = $(dlldir)/asr-mm-control-car-media-conversation.so
```

4. Save the ASR configuration file and return to the QNX terminal.
5. Reboot the target board by issuing the `reboot` command.

The system reboots and relaunches the HMI. Now, when you run media apps and issue voice commands, `mm-control` will process them.

To reenabling the loading of the plugin version that works with `mm-player`, restore the ASR configuration file to its original state and then reboot again.

Chapter 5

Bluetooth: Pairing Devices, Calling, and Messaging

The QNX CAR platform supports two-way Bluetooth discovery for device pairing.

Overview

The **Bluetooth Connectivity** control in the Settings app lets you pair the vehicle's Bluetooth system with another Bluetooth device, such as a smartphone. When the device is paired, you can use the **Communication** screen to access the smartphone's email or text messages, view contact information, and make phone calls.



Bluetooth is not supported on the i.MX6q platform.

Bluetooth phones supported

The following table lists the phones that were tested for this release and gives the level of Bluetooth functionality supported for each. Phones not listed here were not tested.

Phone	HFP	PBAP	PAN	A2DP/AVRCP	SPP	MAP
BlackBerry Bold 9700 NOTE: Tested on QNX CAR 2.0 only.	Yes	Yes	Yes	No	Yes	Yes
BlackBerry Bold 9900	Yes	Yes	Yes	No	Yes	Yes
BlackBerry Z10	Yes	Yes	Yes	Yes	Yes	Yes
Samsung Galaxy S2, S3, & S4 NOTE: Support for A2DP/AVRCP may be limited on the Galaxy S2, depending on the version of the phone's OS.	Yes	Yes	Yes	Yes	Yes	Text messaging only; email not supported on phone.
iPhone 4S (iOS 5)	Yes	Yes	Yes	Yes	Yes	No (not supported on phone).
iPhone 5S (iOS 6)	Yes	Yes	Yes	Yes	Yes	Yes

Pairing a smartphone

You can initiate a Bluetooth pairing request from the QNX CAR platform or a smartphone.

To pair a smartphone with the car's Bluetooth system:

1. Say the command “**Launch Settings**”.



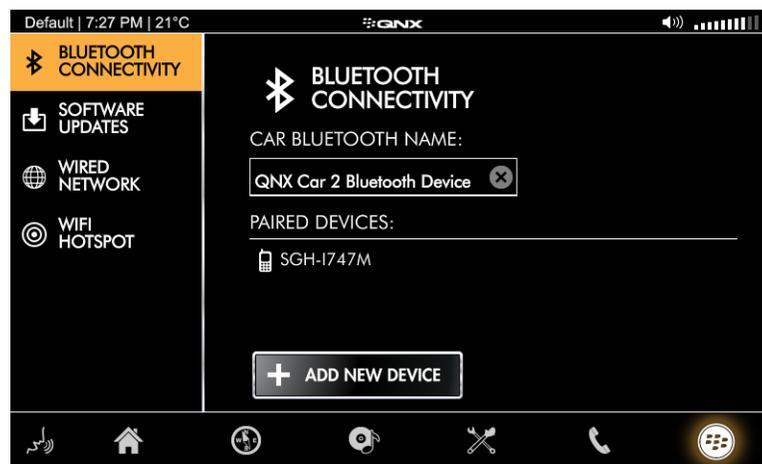
For information about how to use the ASR service to issue voice commands, see “[Automatic Speech Recognition](#) (p. 41)”.

Instead of using ASR, you can access the Settings app via touchscreen commands:

On the taskbar, tap **Apps Section** → **ALL** → **Settings**.

2. In the **Settings** app display, tap **Bluetooth Connectivity**.

The HMI displays the following:



3. Tap **ADD NEW DEVICE**.
4. Initiate the pairing request.

You can initiate pairing from either the HMI or the smartphone.

- **To pair from the HMI:**
 - a. Ensure the Bluetooth device you want to pair with has its “discoverable” setting enabled.
 - b. On the **ADD NEW DEVICE** screen, tap **SEARCH** to find new devices.
All compatible devices found will be listed in the HMI.
 - c. In the **Found Devices** list, tap the entry for the device you want to pair with.

The first time you connect to a device, the system sends the device a six-digit passkey and then displays this passkey in the **Accept Passkey** dialog box. Make sure the passkey shown in the pairing request on your device matches what's shown in the HMI. If so, confirm the matching at both ends.

For all connection attempts, the **Bluetooth Connectivity** feature shows a message telling you the device is connected.

- **To pair from the phone:**

- a. On the **ADD NEW DEVICE** screen, tap **LISTEN** to listen for new devices.

As the Bluetooth system listens for pairing requests, the HMI shows the time remaining for receiving requests.

- b. On the phone, search for Bluetooth devices and find the QNX CAR system.
- c. Send the pairing request from the phone.

The first time you connect from a device, the HMI displays the six-digit passkey sent from the device in the **Access Passkey** dialog box. Make sure the passkey shown in the pairing request in the HMI matches what's shown on the device. If so, confirm the matching at both ends.

For all connection attempts, you must confirm the pairing request in the **Bluetooth Connectivity** feature. The device and the QNX CAR Bluetooth system are then fully connected.

All Bluetooth communication services—**Email Messages**, **Text Messages**, **Address Book**, and **Dial Pad**—are enabled and the relevant data is synchronized from the device when it's paired. For example, the senders, subject lines, and contents of emails stored on the device are uploaded and shown in the HMI.

Making a phone call

You can make a call on a paired smartphone from the HMI. The audio is routed through the device to the QNX CAR system. The dialing feature uses the Hands Free Profile (HFP).

Phone calls can be initiated by dialing (tapping) a number or by using voice commands.

To make a call on a Bluetooth device, do one of the following:

- If you're using ASR, speak the phone number or the name of the contact you want to call (e.g. "Dial 613 591 0931" or "Call Angelina").
- If you're not using ASR, do the following:
 1. Tap the **Dial Pad** button in the **Communication** screen. Or from the menu at the top of this screen, tap the item named **Dial Pad**.

You'll see the dial pad:



2. Dial the phone number by tapping its digits on the dial pad, then tap **CALL**.

The call is sent out from the paired device.

When the recipient answers, the output audio is routed through the device to the vehicle's speakers. The input audio is first captured with the same hardware that processes voice commands and is then sent out through the paired device.



You can terminate the call at any time by tapping the **Disconnect** button.

Receiving or blocking calls

You can also receive or block calls from the HMI by using the **Accept** and **Decline** controls. To monitor incoming calls, you must be using the Dial Pad application. Incoming call dialogs will pop up only if the Dial Pad is the current, active application.

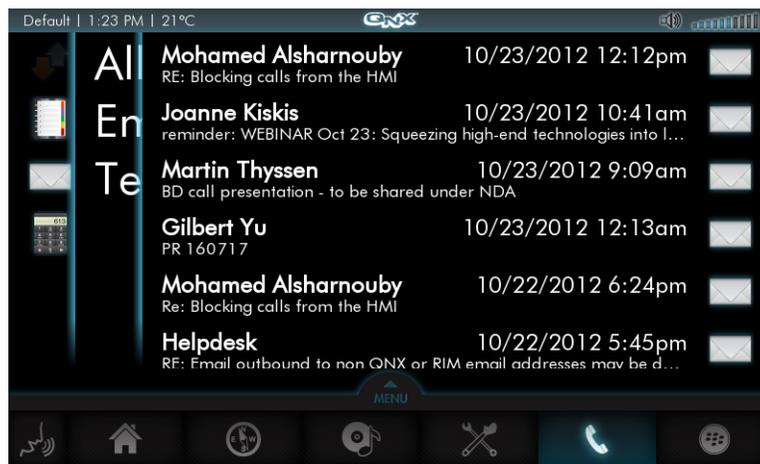
Viewing the phone's email

You can view email messages stored on a paired Bluetooth device in the HMI. The email feature uses the Message Access Profile (MAP).

To view the paired phone's email:

1. Tap the **Email Messages** button in the **Communication** screen. Or from the menu at the top of this screen, tap **Messages** → **Email Inbox**.

You'll see the list of emails from your device's Inbox:



You can scroll through the email messages by swiping up or down in the display area that lists those messages.

2. Tap the subject line of an email to view its contents.

You can return to the list of emails by swiping right from the left border of the display area that shows the email contents.

Viewing the phone's contacts

You can view the contact information stored on a paired Bluetooth device in the HMI. The address book feature uses the Phone Book Access Profile (PBAP).

To view the paired device's contact information:

1. Tap the **Address Book** button in the **Communication** screen. Or from the menu at the top of this screen, tap the item named **Address Book**.

You'll see the list of contacts stored on your device:



You can scroll through the address book entries by swiping up or down in the display area that lists those entries.

2. Tap the name of a contact to see their details.

The HMI displays a thumbnail photo (if available) and all other contact details (including phone numbers, email address, and street address) entered for the contact you selected:



You can return to the list of contacts by swiping right from the left border of the display area that shows the contact details.

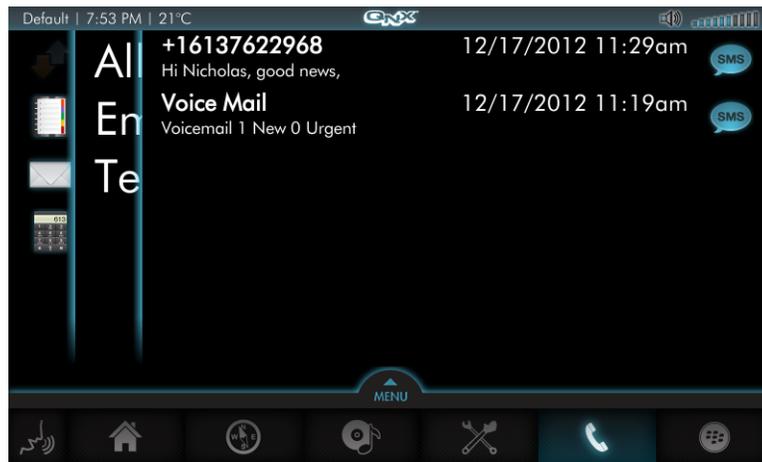
Viewing the phone's text messages

You can view text messages stored on a paired Bluetooth device in the HMI. The text message feature uses the Message Access Profile (MAP).

To view the paired phone's text messages:

1. Tap the **Text Messages** button in the **Communication** screen. Or from the menu at the top of this screen, tap **Messages** → **Text Messages**.

You'll see the list of text messages stored on your device:



You can scroll through the messages by swiping up or down in the display area that lists the text messages.

2. Tap the subject line of a text message to view its contents.

You can return to the list of text messages by swiping right from the left border of the display area that shows the text message contents.

Chapter 6

Network Settings

Accessible from the Settings app, the network configuration controls allow you to select the preferred type of Ethernet connection, access a Wi-Fi network, and set up a Wi-Fi hotspot in the vehicle.

The QNX CAR platform supports wired and wireless connections to portable devices such as smartphones and tablets. You can configure:

- a wired, in-vehicle network to connect different devices to the target platform
- a Wi-Fi hotspot for connecting devices to the head unit so they can access the Internet

You can also search for Wi-Fi hotspots accessible from your current location and attempt to connect to one of them.

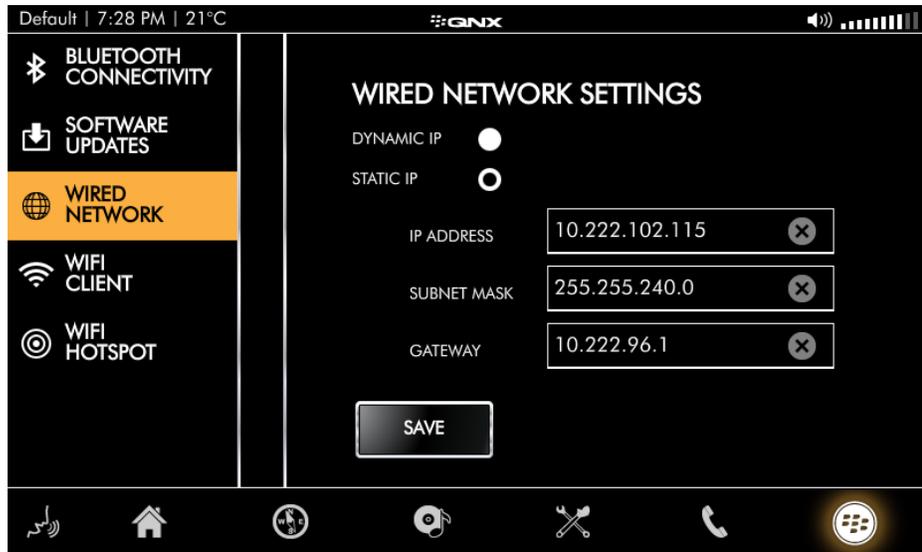
The network settings features use the `net_pps` and `wpa_pps` utilities, which are explained in the *System Services Reference*, as well as several networking-related PPS objects, which are described in the *PPS Objects Reference*.

To configure wired or Wi-Fi settings, tap **Settings** (found in the **Apps Section** screen under **ALL**), then select one of **WIRED NETWORK**, **WIFI CLIENT**, or **WIFI HOTSPOT**.

Configuring a wired network

On the **WIRED NETWORK** screen, choose **Configure**, then set and save the following information, as you would for any other wired network:

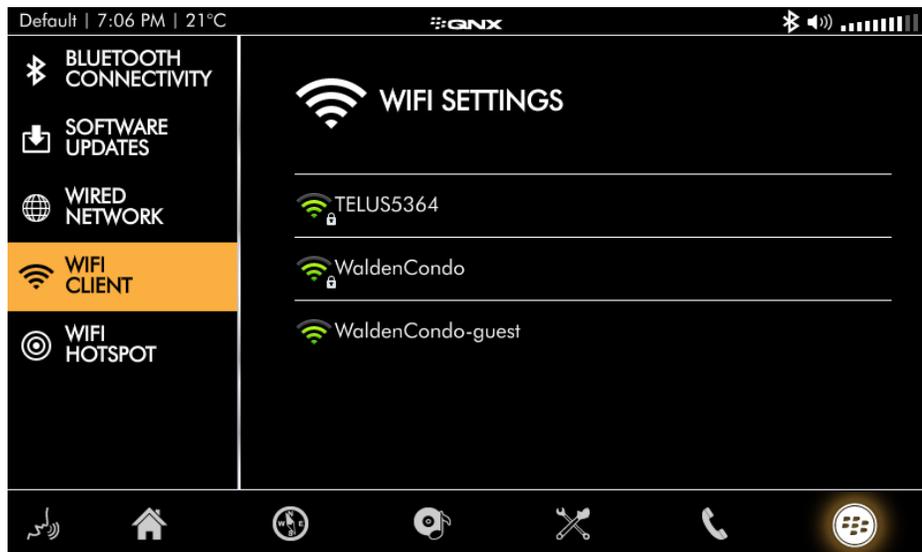
- type of IP address you want to use: static or dynamic (DHCP)
- IP address
- subnet mask
- gateway



When these parameters are set, external devices can connect to the QNX CAR platform over a standard Ethernet connection.

Connecting to a Wi-Fi network

Tap **WIFI CLIENT** to display a list of accessible Wi-Fi networks. Then tap a list entry to attempt to connect to that network.

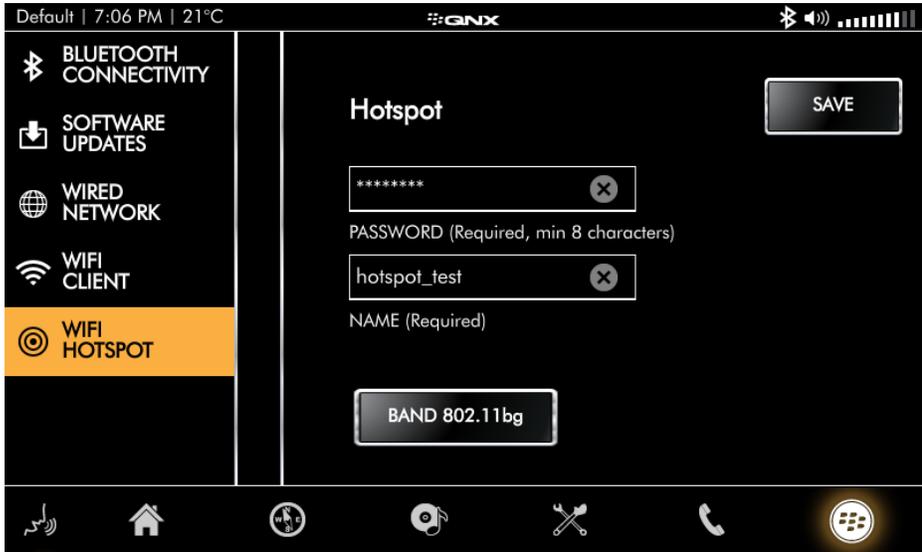


Setting up a Wi-Fi hotspot

Tap **WIFI HOTSPOT** and make sure the **ON** button is highlighted. Then choose **Configure** and enter the following information:

- hotspot name
- password that you want users to enter to access the hotspot

Remember to tap **SAVE** when you're done.



Chapter 7

Rearview Camera

All hardware boards supported by this release can capture video and display a video feed from a camera. If your vehicle has a rearview camera, the camera's video feed replaces the HMI when you put the vehicle in reverse.

The video feed fills the entire display, so you won't even see the status bar at the top or the taskbar at the bottom:



The HMI returns to its previous display when you shift out of reverse.

You can also display this video feed by accessing the Rearview Camera app in the **Apps Section** screen. In this case, the status bar and taskbar remain but the rest of the HMI shows the camera's video feed.

Camera detection

For the supported board types, when the board boots up, the `rearview-camera` service is started. If it detects a video signal, the service displays the video on the screen during startup. If it doesn't detect a signal, the service exits. When the HMI starts up, it takes control of the `rearview-camera` service by using it to start and stop the display of video from the signal detected during bootup. The HMI starts and stops the video rendering based on the activity of the vehicle operator or any apps that interact with the `rearview-camera` service through PPS.



For information about how to connect a camera to your board, as well as about any configuration you may have to do, see the *BSP User Guide* for your board.

Chapter 8

Pandora Radio

The Pandora Radio app is distributed with the QNX CAR platform. You can access a standalone version of the app (under the **ALL** category of the **Apps Section** screen) or an integrated version (through the Media Player).

Pandora Radio is an Internet radio service that users can personalize by creating radio stations that stream content customized to their own tastes.



The service is available only in the United States, Australia, and New Zealand.

To work with Pandora Radio, you must first create a Pandora Developer account on the Pandora website (<https://developer.pandora.com/>). At the Pandora site, click **Visit the sign up page**, then complete and submit the application form to request access to the API.

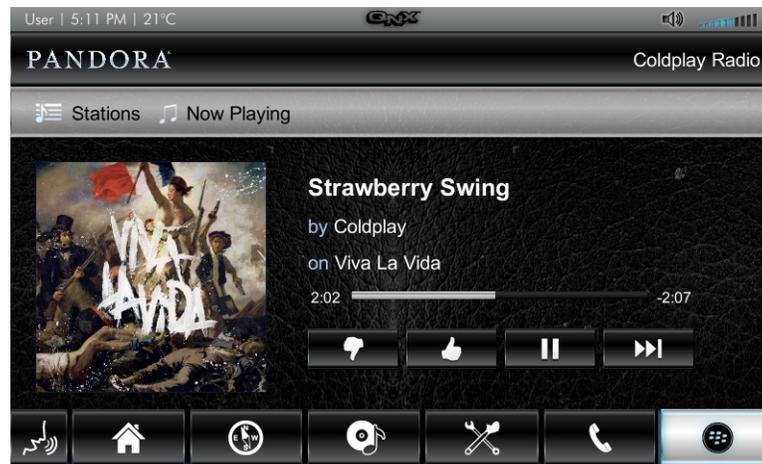
Pandora provides two sets of usernames and passwords. Using one set of these credentials, you can log in from the Pandora app:

1. On the taskbar, tap **Apps** → **ALL** → **Pandora**.
2. If you're accessing Pandora Radio for the first time since starting your system, you'll be prompted for authentication credentials:



3. On the login page, type your credentials and then tap **Submit**.

After a successful login, the Pandora application is activated and appears on the screen:



You can enter the same credentials to access the version of Pandora that's integrated into Media Player. For more information, see "[Play Radio](#) (p. 17)".



Since Pandora uses SSL to securely access websites it may need, you must make sure that the system clock on your platform is set during startup, before the HMI loads. Otherwise, Pandora may fail. For more information, see "Realtime Clock Synchronization" in the *System Services Reference*.

Theming in Pandora

Theming in Pandora is provided by the theming API. All styles and image assets defined in the API are located under `/usr/hmi/common/themes/`. The theme API `app name` used for the external Pandora application is `PandoraExternal`.

The Pandora application changes skins when the user switches between the HMI's `default`, `midnightblue`, and `titanium` themes.

Chapter 9

Recalibrating the Touchscreen

You can recalibrate your target's touchscreen at any time.

To recalibrate the touchscreen:

1. Remove the following file:

```
/var/etc/system/config/calib.localhost
```

2. Run the `reboot` script.

The system reboots and the calibration utility (`calib-touch`) runs, asking you to touch the center of a “target” multiple times and then to accept the new calibration settings.



For more information about the `calib-touch` utility, see its entry in the *OS Utilities Reference*.

Chapter 10

Software Updates

The QNX CAR platform supports software updates by using Red Bend's vRapid Mobile® FOTA software (v8.0.1.29). You can apply updates from the HMI or the command line.

To apply an update, you'll need a software update package that includes, at a minimum, a *delta file*. The delta file describes the filesystem changes needed to upgrade your current system to a new version. You can obtain a delta file from your system provider or generate your own. The *System Services Reference* has instructions on generating a delta file.

If you update your system through the HMI, your update package must also include a *manifest file*, which you must write. The *System Services Reference* contains a sample manifest file.

Applying an update from the HMI

You can use the Settings app to initiate updates with one button press.



To update your system from the HMI, you must have a valid *delta file* as well as a *manifest file*. The *System Services Reference* has instructions on generating a delta file and an explanation of the structure and contents of the manifest file (which you must write), including a sample manifest file.

To apply an update from the HMI with the Settings app:

1. Copy the manifest file into the root directory of a USB mass-storage device formatted as DOS/FAT32.

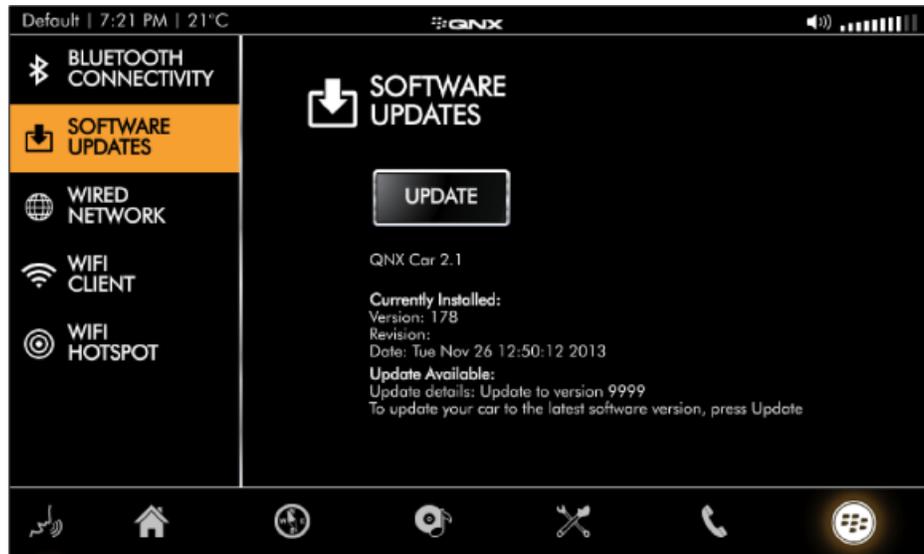
The filename must be of the format *myname.manifest*, where *myname* is a substring of your choosing. This should be the only manifest file stored on the device.

2. Copy the delta file into the root directory of the same storage device.

The delta file can have any name but it must match what's specified in the `path` key of the manifest file. To ensure the name accurately reflects the delta file's purpose, we recommend following the filename syntax of `QNXCAR2-myname-from_version-to_version.mld`; this syntax is explained in the instructions for generating a delta file.

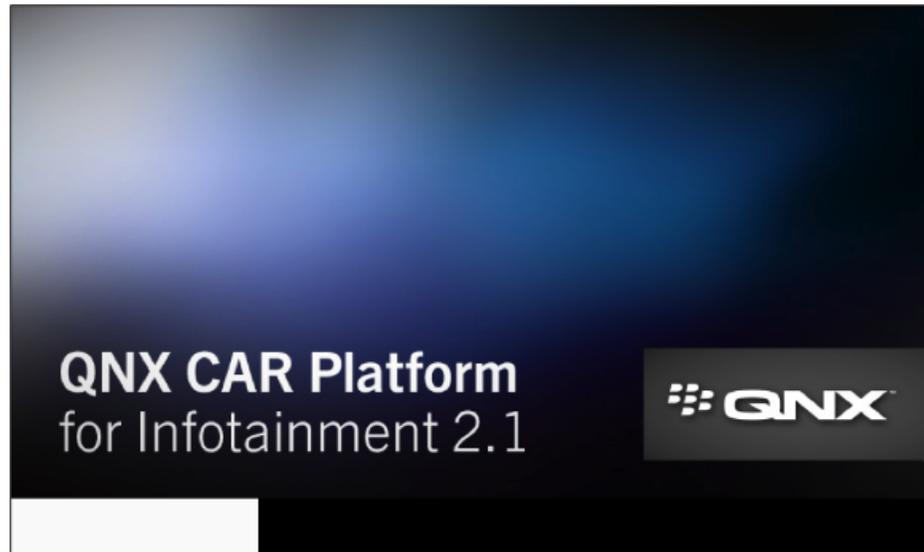
3. Plug the storage device into a USB port on the target board.
4. From the HMI Home screen, select **Apps Section** → **ALL** → **Settings**.
5. Once the Settings app has launched, select the **SOFTWARE UPDATES** tab along the left-hand side of the screen.

The Software Updates app reports the version of the currently installed software. When the app detects a manifest file on an attached USB device, it displays information about the update and enables the **Update** button:



6. Press **Update** to start the update.

The system copies the manifest file and the delta file from the USB device to `/var/swud`. Next, the software update daemon, `swud`, uses `downsize` to terminate all nonessential processes and then applies the update, displaying the progress in a progress bar on the screen:



The system then reboots. If the update is successful, the system will reboot back into the HMI. If the update is interrupted by, say, a board reset, the system will attempt to reboot again and to reapply the update.

Applying an update from the command line

You can use the `reboot` command to initiate updates from a QNX Neutrino terminal.



To update your system from the command line, you need only a valid delta file. You don't have to supply a manifest file because the `reboot` script creates one to use in the update. The *System Services Reference* has instructions on generating a delta file.

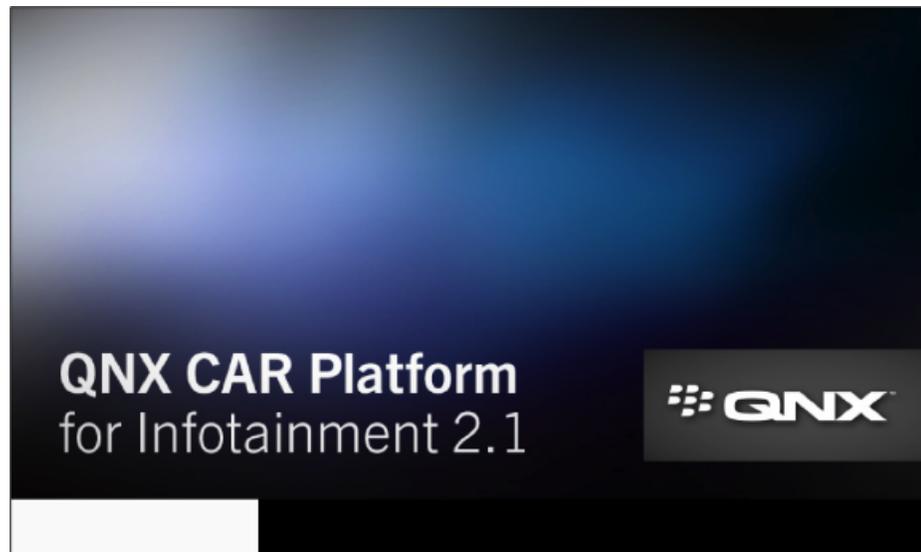
To apply an update from the command line:

1. Copy the delta file from an external device (e.g., USB, NFS) into `/var/swud/mydelta.mld`.

The filename must be `mydelta.mld` when updating from the command line.

2. In a QNX Neutrino terminal, enter the command `reboot -U`.

The system generates the manifest file in `/var/swud`. Next, the partial-shutdown utility, `downsize`, terminates all nonessential processes and then applies the update, displaying the progress in a progress bar on the screen:



The system then reboots. If the update is successful, the system will reboot back into the HMI. If the update is interrupted by, say, a board reset, the system will attempt to reboot again and to reapply the update.

Chapter 11

Downloading Apps from the QNX App Portal

The *QNX App Portal* lets developers showcase their automotive-relevant apps for you to download and evaluate on your target.



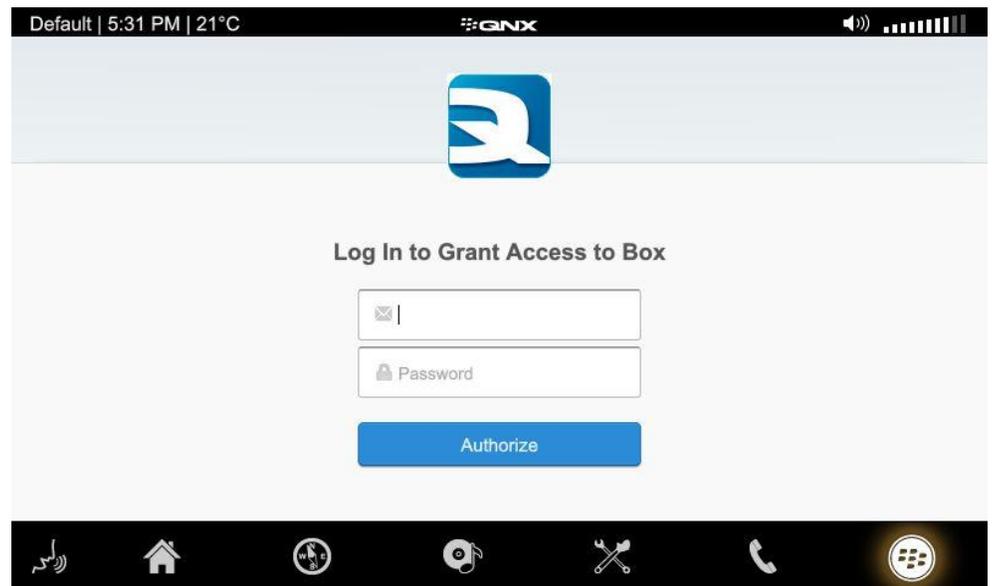
As a customer of the QNX CAR Platform for Infotainment, you'll be invited to collaborate on the QNX App Portal via email. Before you can use the QNX App Portal client on your target, you must accept the email invitation and set up a [Box](#) account (if you don't already have one). If you'd like to access the QNX App Portal but haven't received an email invitation, please contact your sales representative.

To run the QNX App Portal client on your target:

1. Under the **ALL** category on the **Apps Section** screen, tap the QNX App Portal icon:

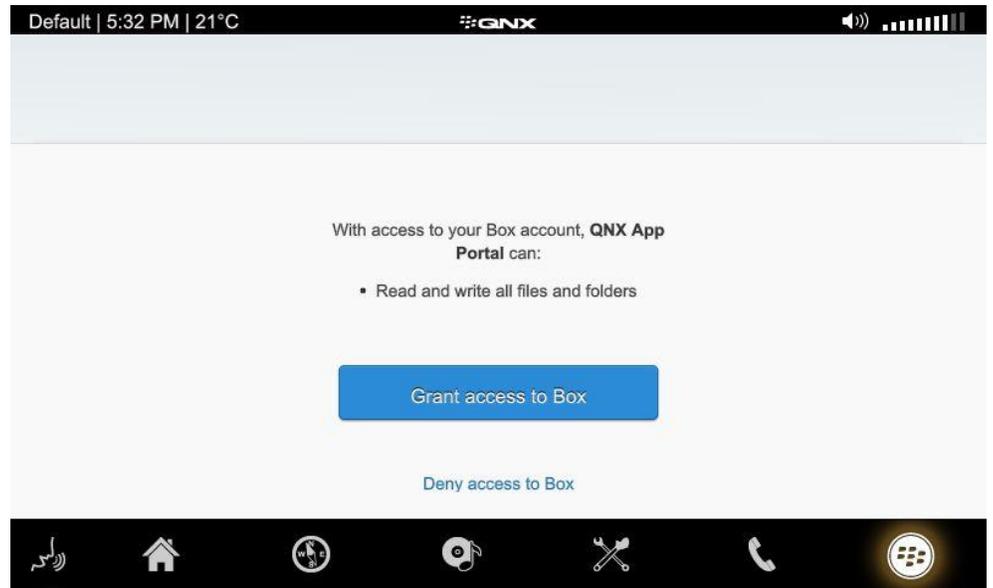


You'll see the Box login page:



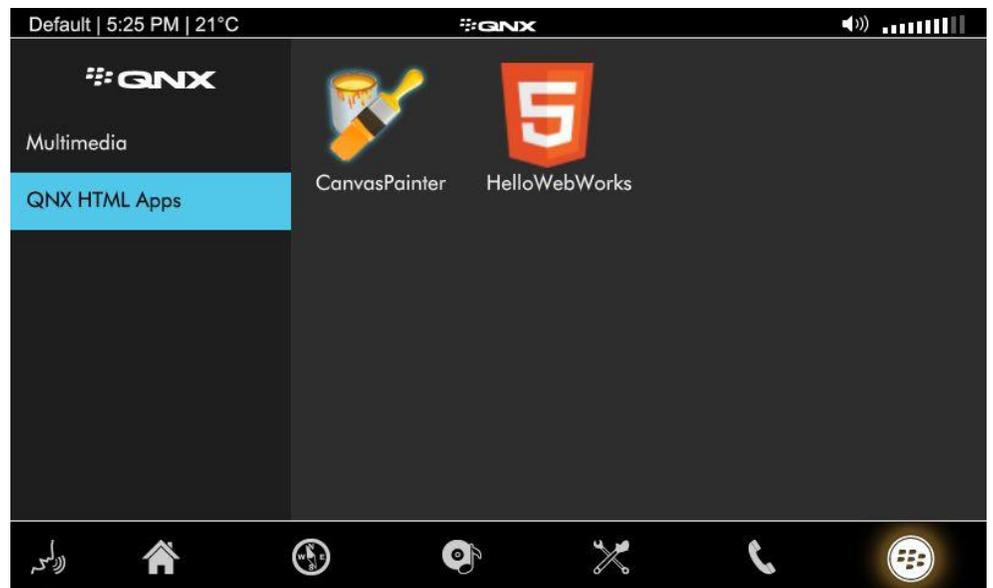
2. Enter the username and password for your Box account, then tap **Authorize**.

The QNX App Portal app then prompts you for permission to access your Box account's files and folders:

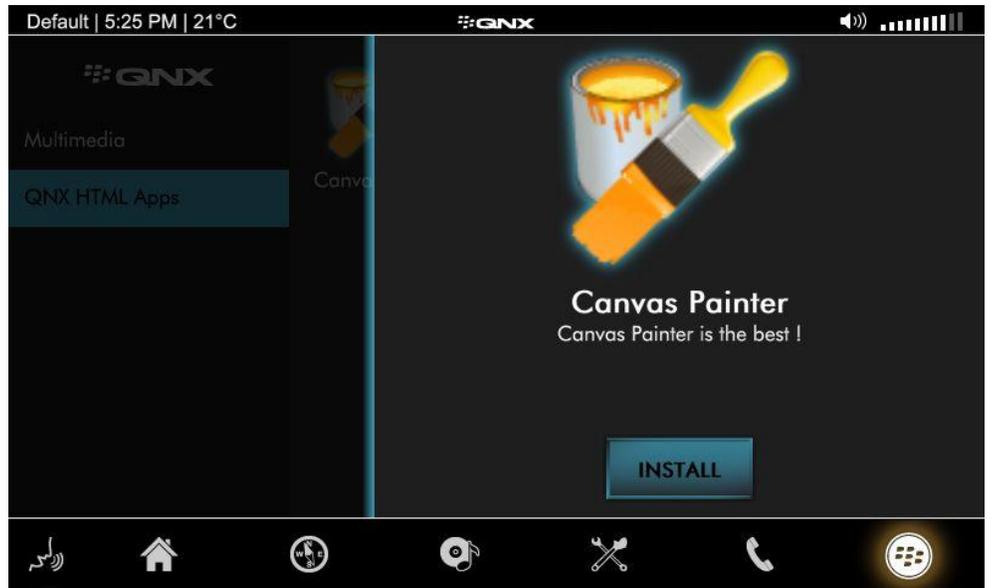


3. Tap **Grant access to Box**.

You'll then see the apps available for you to download and install:



4. Select the app you want to download, then tap **INSTALL**:



You should then see the app's licensing information on your screen.

5. To begin the installation, tap **Accept**.



You can also use the QNX App Portal client app to *uninstall* any apps you've downloaded and installed on your target.

Under the **ALL** category in the **Apps Section** screen you'll find the new app's icon—tap it to run it.

Index

/var/etc/system/config/calib.localhost 69

A

add-on applications 11
 API Demos 32
 apps 36
 Push-to-Talk 36
 Apps Section screen 32
 ASR 23, 36, 41
 recognition latency 41
 starting a session 36, 41
 using to play media 23
 asr-car.cfg 41
 AUDIO CONTROL button 24
 automatic speech recognition, See ASR

B

BestParking 32
 Bluetooth 51, 53, 54, 56, 57, 59
 control in Settings app 51
 Hands Free Profile (HFP) 54
 Message Access Profile (MAP) 56, 59
 pairing a smartphone 53
 Phone Book Access Profile (PBAP) 57
 phones supported 51
 Bluetooth Connectivity control 52
 Browser 32

C

Calculator 32
 camera 65
 detection at boot 65
 Car Control app 24, 25, 28
 Audio Control feature 24
 Climate Control feature 25
 Personalization feature 28
 Virtual Mechanic feature 25
 Car Control screen 24
 Changing HMI themes 29
 CLIMATE CONTROL button 25
 Communication screen 31
 configuration 61
 network for QNX CAR platform 61
 contacts stored on paired phone, viewing 57
 core applications 11
 CSS3 12

D

delta file required for updating software 71
 detection 65
 camera at boot time 65

DHCP 61
 configuring for QNX CAR platform 61
 DLNA 21

E

email stored on paired phone, viewing 56
 End of Speech (EOS) 41
 Ethernet 61
 configuring for QNX CAR platform 61
 eventseeker 32

H

Hasselhoff, David 23
 HMI 11, 12, 14, 20, 37
 applications 11
 changing default version 37
 initial screen 12
 media search feature 20
 status bar 14
 taskbar 14
 UI frameworks 12
 versions 11
 Home screen 14
 HTML5 11, 15
 HMI version 11, 15
 HTML5 HMI 37
 launching at startup 37

I

installation summary 9
 io-asr-generic 41

J

JQuery 11
 Juke Box 19

L

legacy media applications 50
 enabling media voice commands 50
 Local Search 32
 in Apps Section 32

M

manifest file required for updating software 71
 Media Player screen 17
 media source menu 21
 mm-control 50
 enabling media voice commands 50

N

- navigation 16
 - Elektrobit (EB) street director engine 16
- Navigation screen 16
- network 61
 - configuring for QNX CAR platform 61

P

- pairing a smartphone 53
- Pandora 32
- Pandora Radio 17, 33, 67
 - integrated version 17
 - overview 67
 - standalone version 33
- Parkopedia 32
- PeaksAndValleys 32
- PERSONALIZATION BUTTON 28
- phone calls on a paired phone 54
- Play Music control 18
- Play Radio control 17
- Play Video control 19
- Push-to-Talk screen 36

Q

- QNX App Portal 32, 75
 - Box account required for 75
 - client application 75
 - in Apps Section 32
- Qt framework 11, 12
 - used in applying UI theme 12
- Qt5 11, 15
 - HMI version 11, 15
- Qt5 HMI 32
 - support for HTML5 applications 32

R

- Rearview camera 65
 - overview 65
- Rearview Camera 32
 - in Apps Section 22
- rearview-camera service 65
- reboot script 34
- Running the HTML5 HMI 37

S

- screens 14, 16, 17, 24, 31, 32, 36
 - Apps Section 32
 - Car Control 24
 - Communication 31
 - Home 14
 - Media Player 17
 - Navigation 16
 - Push-to-Talk 36
- SCSS 12
- Search control 20
- Sencha Touch 11

- Settings 32, 34
 - Bluetooth Connectivity 34
 - in Apps Section 32
 - Software Updates 34
 - WiFi Hotspot 34
 - Wired Network 34
- Shutdown 32, 34
 - in Apps Section 32
- Slacker Radio 32
- SLM configuration file 37
- Software Updates 71
- Speech-To-Text (STT), See ASR
- static IP address 61
- supported media sources 21
- supported voice commands 43

T

- Technical support 8
- text messages stored on paired phone, viewing 59
- Text-to-speech (TTS) latency 42
- touchscreen 69
 - recalibrating 69
- TunnelTilt 32
- Typographical conventions 6

U

- UI overview video 12
- UI theme 12
 - changing 12
 - rendering 12
- updating your QNX CAR platform 72, 74
 - from the command line 74
 - from the HMI 72

V

- video 19, 20
 - codecs (Freescale i.MX6q) 20
 - Jeep Wrangler 19
- VIRTUAL MECHANIC button 25
- voice commands 43, 44, 45, 46, 47, 48
 - control the media player 48
 - find a nearby POI 46
 - get directions to an address 46
 - get help 48
 - get the current time or date 45
 - get weather information 44
 - launch an application 44
 - list of 43
 - make a phone call 48
 - play music 48
 - search for media tracks or internet POIs 47
 - show a map of a city 47
 - stop using ASR 48
 - terminate an application 44
- voice commands for playing media 23

W

WeatherNetwork 32
default location, changing 32

Wi-Fi 62
configuring a hotspot 62
connecting to a network 62

