

User's Guide

©2014–2015, 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: April 16, 2015

Contents

About This Guide	5
Typographical conventions.....	6
Technical support.....	8
Chapter 1: Overview of the QNX SDK for Apps and Media	9
Chapter 2: A Guided Tour of the Reference Image	11
Audio Demo.....	15
Browser.....	17
Browser Lite.....	21
Camera.....	22
Cordova PPS Demo.....	24
Full Screen HMI.....	26
Hello World.....	27
IP Camera.....	28
Media Player.....	30
Peaks and Valleys.....	32
Photo Viewer.....	34
Settings.....	35
Shutdown.....	37
Video Demo.....	38
Index	41

About This Guide

The *User's Guide* provides a tour of the QNX SDK for Apps and Media platform 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:
General information about the QNX SDK for Apps and Media	Overview of the QNX SDK for Apps and Media (p. 9)
The sample HMI provided with the reference image, and how to interact with the sample apps	A Guided Tour of the Reference Image (p. 11)

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	unsigned short
Environment variables	<i>PATH</i>
File and pathnames	/dev/null
Function names	<i>exit()</i>
Keyboard chords	Ctrl–Alt–Delete
Keyboard input	<code>Username</code>
Keyboard keys	Enter
Program output	<code>login:</code>
Variable names	<i>stdin</i>
Parameters	<i>parm1</i>
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.



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



WARNING: 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).

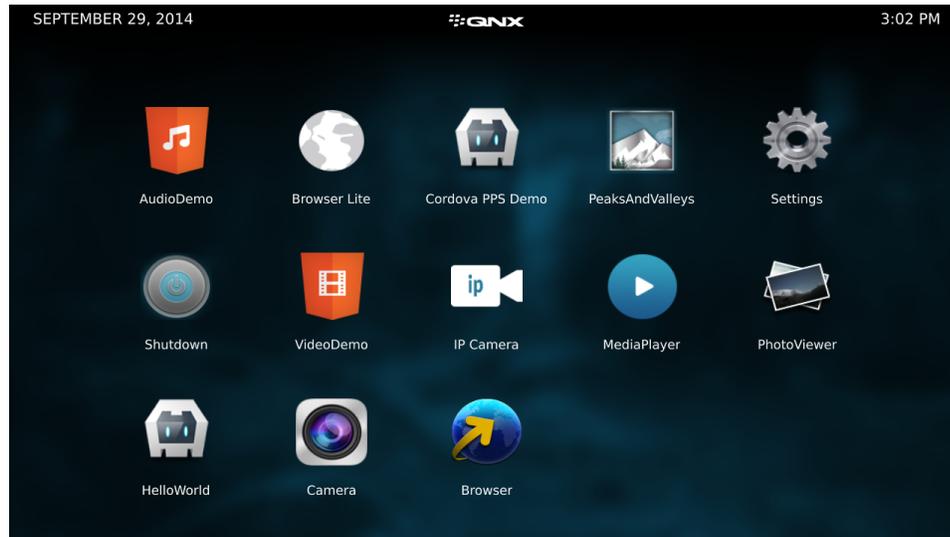
You'll find a wide range of support options, including community forums.

Chapter 1

Overview of the QNX SDK for Apps and Media

The QNX SDK for Apps and Media is a development platform that includes multimedia support, an application management framework, and a chromeless browser powered by the SDK's browser engine. To demonstrate the capabilities of this browser engine, the SDK also includes a web browser application's chrome written entirely in HTML5. These features give you the tools to create a wide variety of native and web-based applications for embedded systems.

This is the sample Home screen:



The SDK includes:

Multimedia components

Support audio and video playback and synchronization of media metadata from connected devices to embedded databases.

Application management framework

Includes tools for installing and uninstalling applications in your embedded system, as well as services for:

- launching applications (**launcher**)
- controlling access to system services (**authman**)

Embedded browser

Provides a mechanism for running web-based apps in your system. Based on WebKit, the browser provides support for HTML5 and related standards and technologies (including CSS3) and for JavaScript and associated standards, such as AJAX, JavaScript Object Notation (JSON), and XML. This browser is embedded in the target image for your board.

Chapter 2

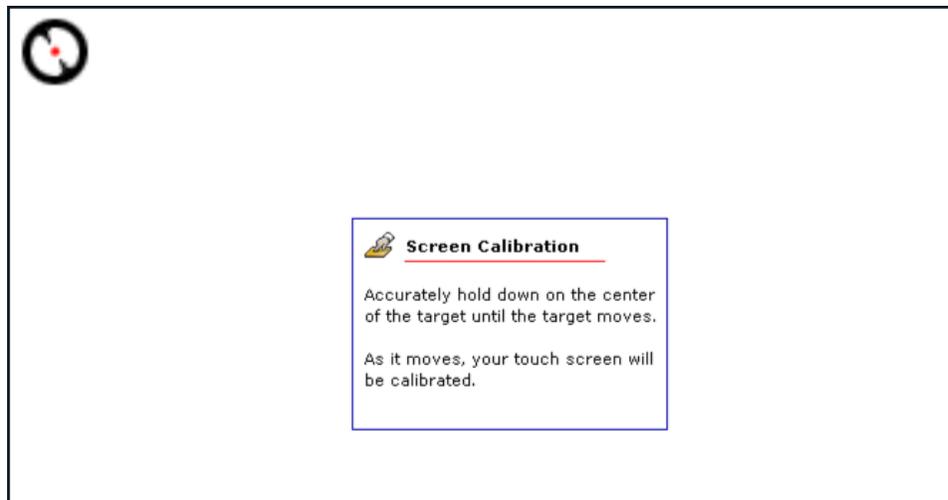
A Guided Tour of the Reference Image

The reference image for the QNX SDK for Apps and Media includes a sample HMI that displays the installed apps.

The HMI can run apps created with Qt or HTML5 (with Cordova plugins). The reference image contains sample apps written with these two technologies.

First time booting your image on a target

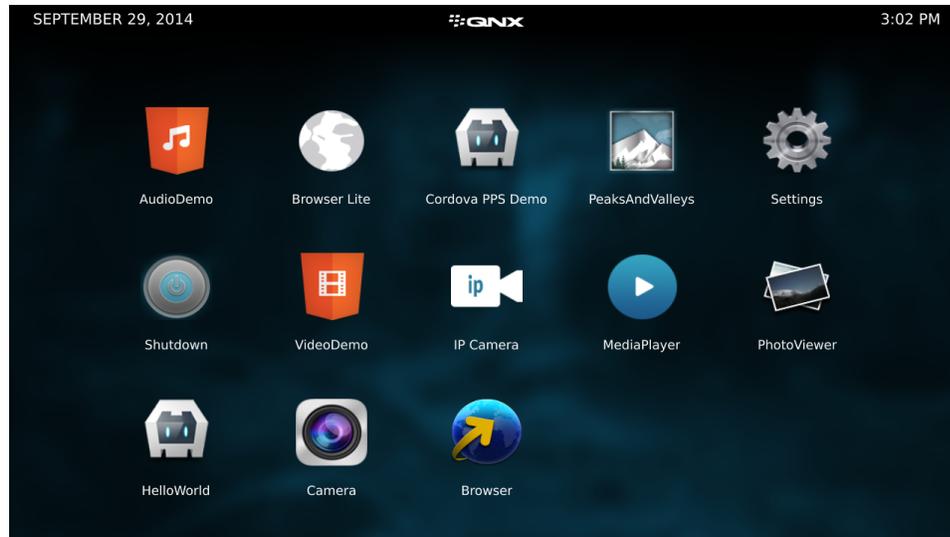
When you first boot your target, it might be necessary to calibrate the screen. Touch the dot in the upper-left corner and continue to touch the dot where it appears on the display until the calibration process is complete.



After the target board boots, the sample HMI appears. The HMI allows you to launch apps from the touchscreen.

Home screen

The HMI provided with your reference image is built using Qt. This HMI is referred to as the *Home screen*. Your reference image includes fully-functional sample apps written with Qt version 5.3.1 and HTML5. At the top of the Home screen is a status bar showing the QNX logo and the date and time. Below the status bar, the Home screen displays the icons of the installed apps.



The following apps are not available in VMware images:

- Camera
- IP Camera
- Video Demo

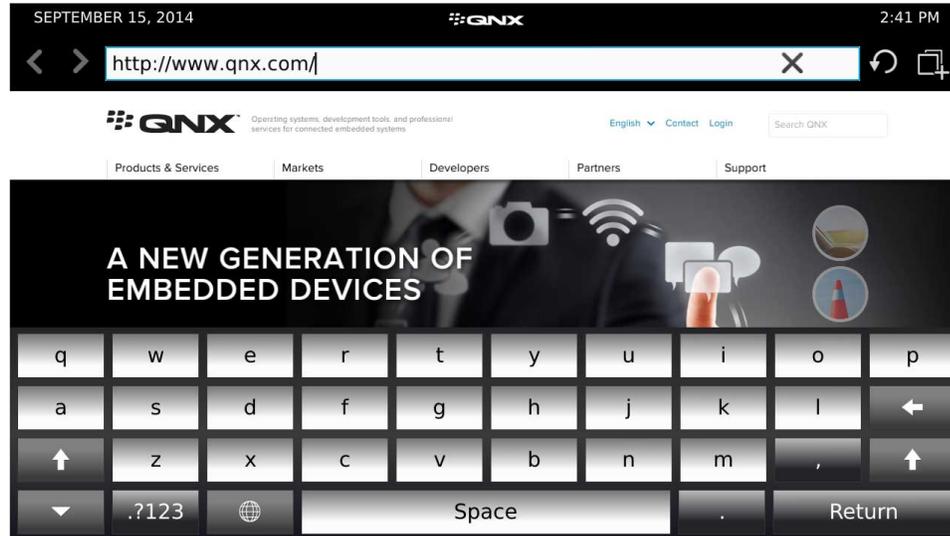
You can tap any icon to start an app. Apps run inside browser chrome that includes the *status bar* from the Home screen and a *control bar* along the bottom. You can interact with the app using its user interface. From the control bar, you can:

- tap **CLOSE APPLICATION** to stop the app and return to the Home screen
- tap the speaker buttons to increase or decrease the volume



Software keyboard

Included with the reference image is a software keyboard. If you have a USB keyboard connected to your target, it's recognized when the image starts; otherwise, a software keyboard appears when you tap a field that requires keyboard input. For example, if you tap the URL box in the Browser Lite application, a keyboard appears on the screen. When you tap an area that doesn't require keyboard input, the keyboard disappears from the screen.



Reference apps

The reference apps installed on the image are as follows:

Reference Application	Description
Audio Demo (p. 15)	An HTML5 app that plays audio files
Browser (p. 17)	A fully functional web browser created using HTML5 and JavaScript
Browser Lite (p. 21)	A simple web browser written with HTML5 and Cordova plugins
Camera (p. 22)	An HTML5 app that demonstrates connecting to a video camera on a target board
Cordova PPS Demo (p. 24)	An HTML5 app that demonstrates how to use the PPS service in an HTML5 application
Full Screen HMI (p. 26)	A simple HMI that shows how to develop a monolithic HMI without using packaged apps
Hello World (p. 27)	An HTML5 app that shows how to display text
IP Camera (p. 28)	A Qt app that displays a video feed supplied by an RTP/IP-based camera
Media Player (p. 30)	A Qt app that demonstrates browsing and playing audio and video content stored in a local directory
Peaks and Valleys (p. 32)	An HTML5 app that uses the WebGL API to show a 3D terrain

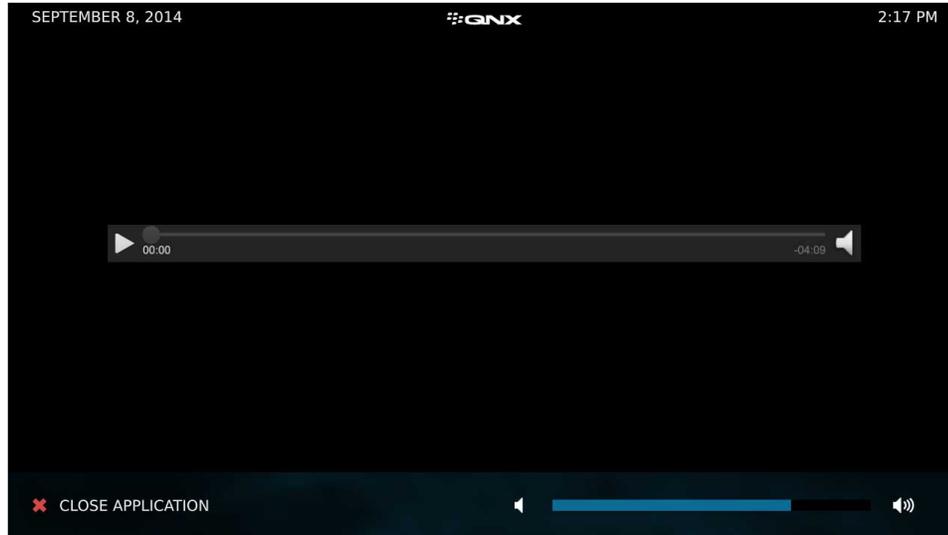
Reference Application	Description
Photo Viewer (p. 34)	A Qt app that displays picture files stored locally
Settings (p. 35)	A Qt app that provides controls to configure the system
Shutdown (p. 37)	An HTML5 app that shows how to reboot the system
Video Demo (p. 38)	An HTML5 app that plays a short video

Audio Demo

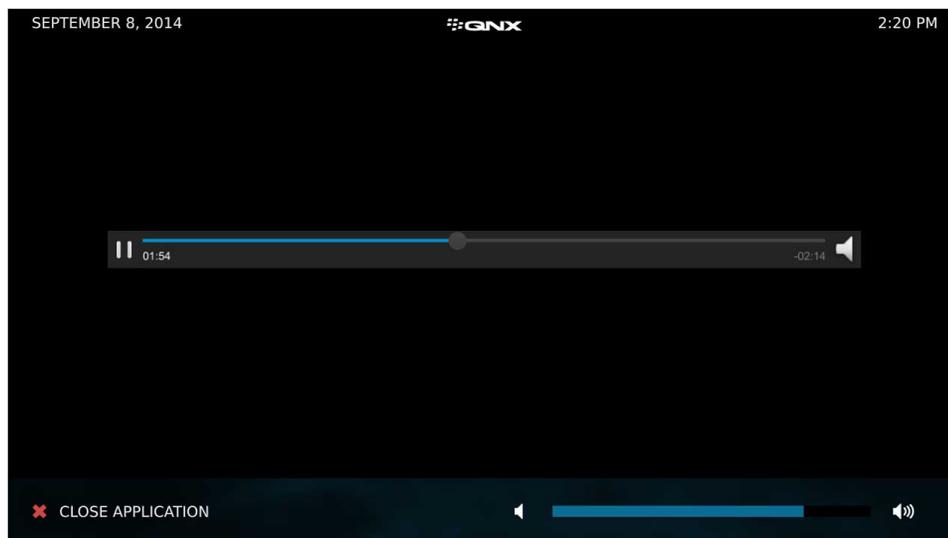
The Audio Demo application is a reference HTML5 sample that shows how to play an audio file.

The app demonstrates audio playback through HTML5 controls, by using the HTML5 `<audio>` element to play a provided MP3 file.

To start the application, tap **AudioDemo** from the Home screen. After the app starts, a screen appears with a **Play** button and a progress bar.



To play the audio, tap the **Play** button. This button then changes to a **Pause** button as playback begins. You can toggle between the pause and play functions and tap the volume control near the bottom to adjust the volume. If you tap the **Speaker** button in the progress bar, the progress bar becomes hidden.



Considerations:

Connect the target board to speakers so that you can hear the audio.

Location on the reference image:

`/apps/AudioDemo.testDev_AudioDemo__82a252b9`

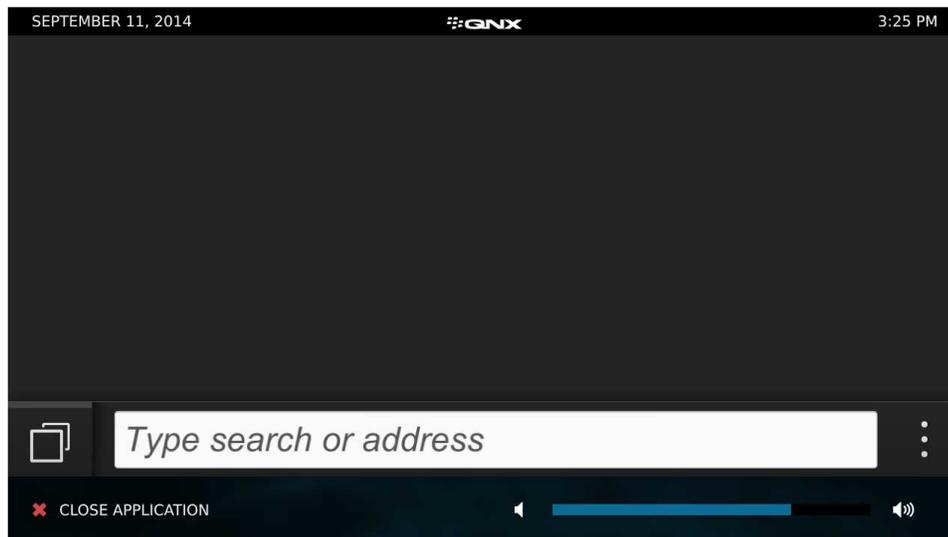
Browser

The Browser app is a fully functional web browser built using HTML5 and JavaScript.

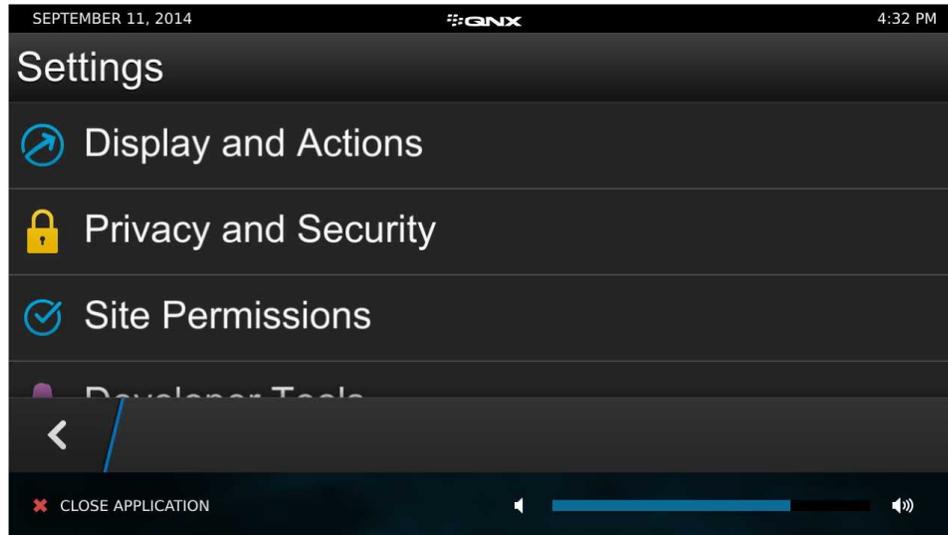
The app contains portions of the WebKit Open Source Project and the JavaScriptCore Project. The browser itself has been extended with numerous capabilities to provide full-featured web browser capabilities. By default, the browser supports 720p resolutions. You can reuse this app in your image to provide these browser capabilities on your target:

- URL bookmarking
- tracking of the last visited sites and creating tiles with links to those sites
- progress checking of file downloads
- configuration of settings such as display, privacy, and site permissions
- creation of new tabs
- checking of browsing history

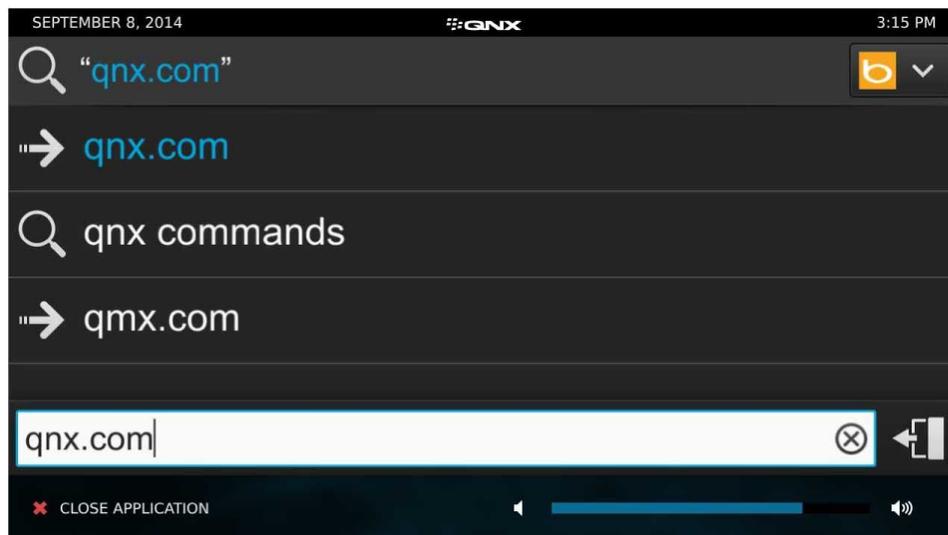
To start Browser, tap **Browser** on the Home screen.



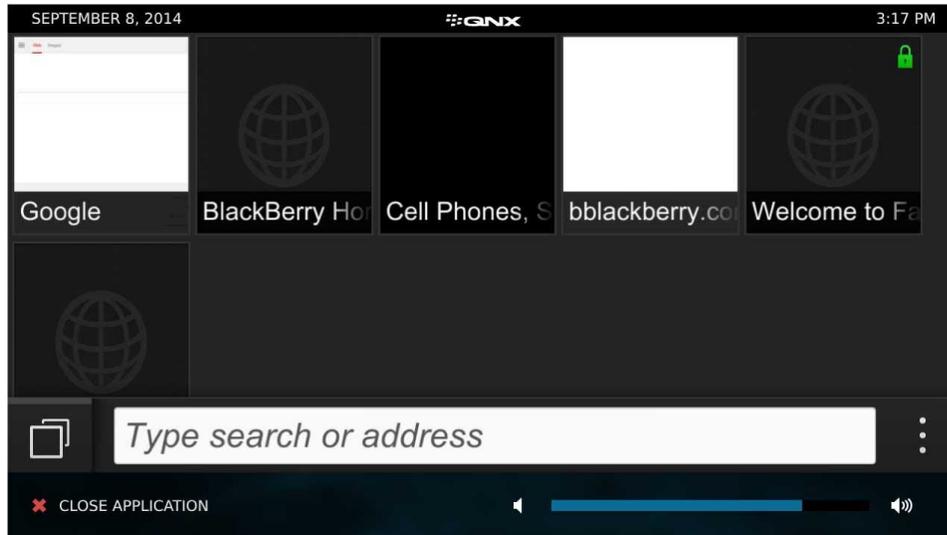
In the Browser application, you can perform various tasks. For example, in the application, you could tap the **ellipsis** button on the bottom-right corner and then tap **Settings** to modify various browser settings.



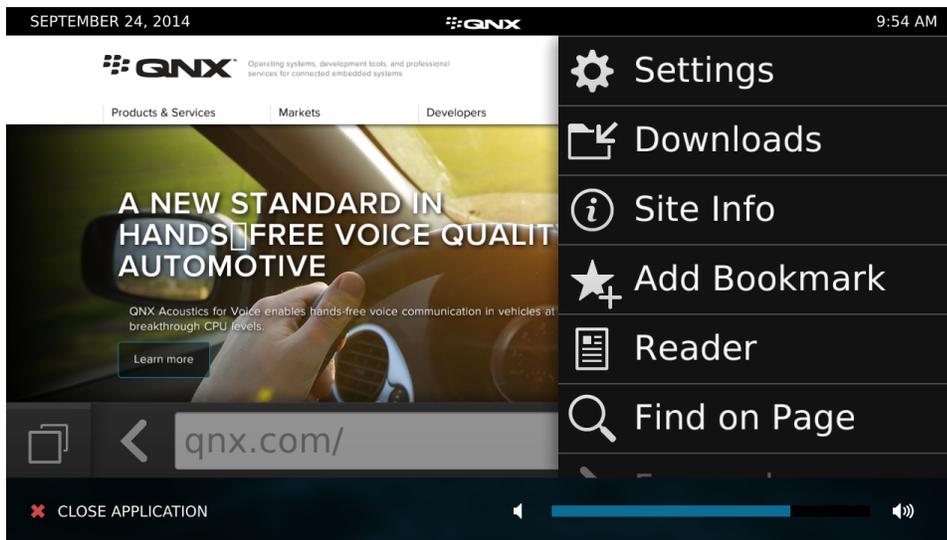
When you perform searches, the browser does auto-completion on terms that you enter. For example, a number of terms appear if you type `qnx.com`.



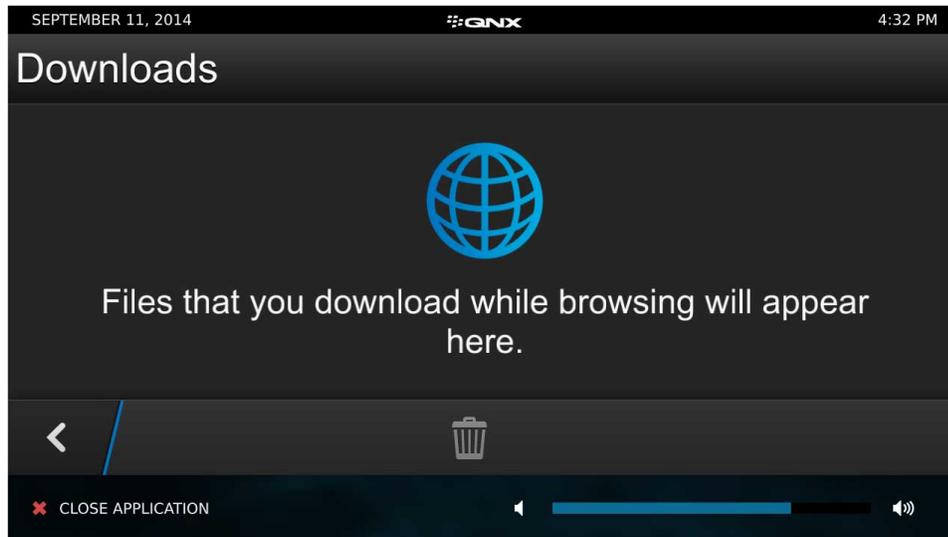
When you restart the browser, your recent history is stored as tiles for quick and easy access.



You can also perform operations on URLs that you have open, such as bookmarking, setting favorites, and so on.



You can also manage download operations completed by the browser and see the progress of active downloads.



Considerations:

To browse the web, ensure that you have a functional network connection to your target.

When there's a USB keyboard connected to your target, it's recognized when the image starts; otherwise, a software keyboard appears when you tap a text field (e.g., the URL bar). In both cases, the browser resizes to make room for the software keyboard, however the software keyboard is only visible when a USB keyboard isn't connected. To prevent the browser from resizing, complete these steps:

1. Connect to the target, edit the `/scripts/env.override.variant` file, and add the following line:

```
export HAS_SOFTWARE_KEYBOARD=false
```



You may need to run the command `mount -uw /base` to write to the filesystem if it's a read-only partition.

2. Save the file and reboot your target.

Location on the reference image:

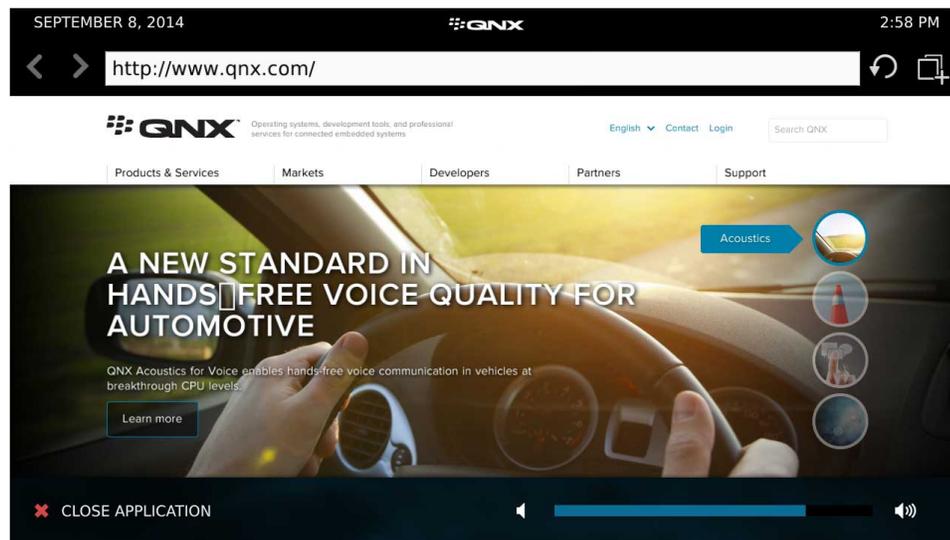
`/apps/sys.browser.gYABgJYFHAzbeFMPCCpYWBtHAm0`

Browser Lite

The Browser Lite application is a reference HTML5 sample that shows how to create a simple web browser using WebViews.

This programming sample demonstrates how you can quickly build a custom web browser using HTML5 and Cordova (with a plugin called `qnx.browser`).

To start the application, tap **Browser Lite** on the Home screen. By default, the **www.qnx.com** URL is loaded when the app first starts and whenever a new tab is created. You can navigate, add, and close tabs. You can also tap the URL bar and type a new URL. The app also provides **Reload**, **Previous URL**, and **Next URL** buttons.



Considerations:

To browse the web, ensure that you have a functional network connection to your target.

When there's a USB keyboard connected to your target, it's recognized when the image starts; otherwise, a software keyboard appears when you tap the URL bar.

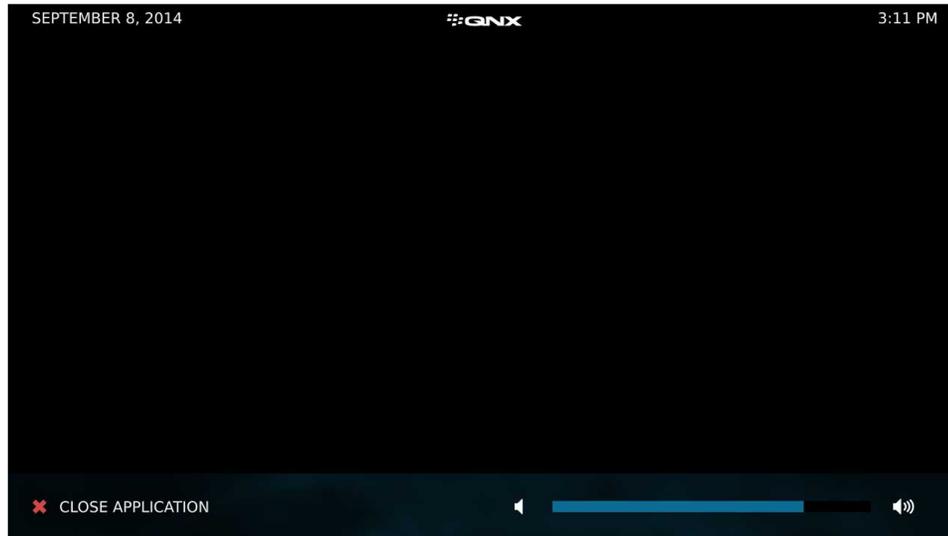
Location on the reference image:

`/apps/BrowserLite.testDev_BrowserLite___6c37c308`

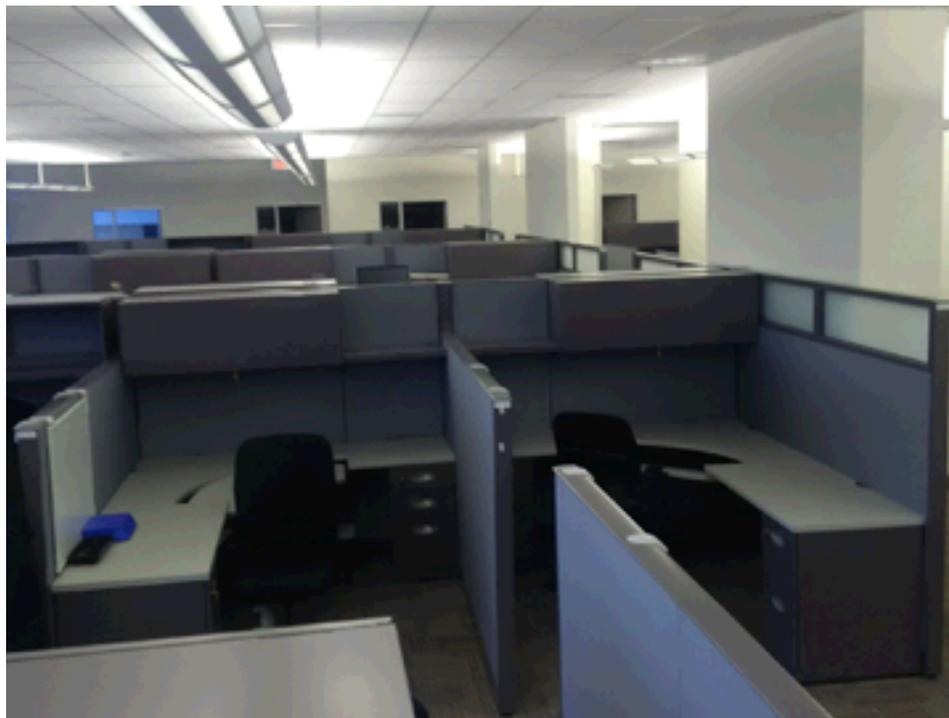
Camera

The Camera app is a reference HTML5 sample that demonstrates how to access the feed of a connected video camera.

The app accesses the video camera attached to your target board. If a video camera isn't available, the sample app shows black when it runs.



To start the application, tap **Camera** from the Home screen. If you do have a video camera connected to your system, you should see a video feed.



Considerations:

This app isn't available in VMware images.

The board that runs the reference image should have a camera. If it doesn't, you can connect a USB camera to the board. The USB camera you use should be USB Video Class (UVC) Compliant. USB cameras, such as iGo and iCubie have been tested with the QNX SDK for Apps and Media and are recommended.

To run the USB camera, use `elvis` or `vi` on the target board to edit `/var/etc/services-enabled` and change this line:

```
USBCAM:false
```

to:

```
USBCAM:true
```

Then, save the `services-enabled` file and restart the board.

When the HMI starts, the video feed from the camera signal is detected during bootup. The HMI starts and stops the video rendering based on the activity of any apps that interact with the camera service through a specific Persistent Publish/Subscribe (PPS) object.



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

Location on the reference image:

`/apps/rearview_camera.testDev_view_camerad91629db`

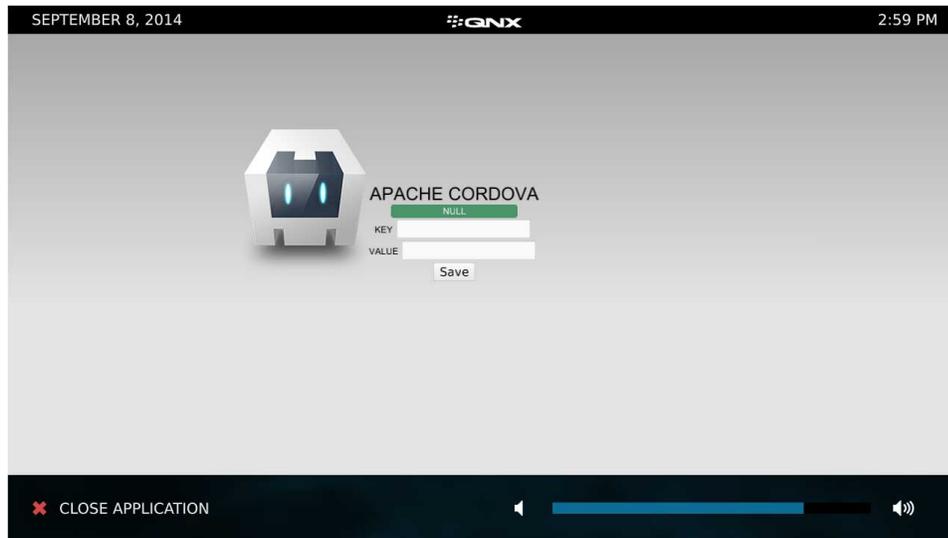
Cordova PPS Demo

The Cordova PPS Demo app is a reference HTML5 sample that demonstrates how to use the Persistent Publish/Subscribe (PPS) service in an HTML5 application.

The app also demonstrates how to restrict access to a PPS object. You can use this sample to:

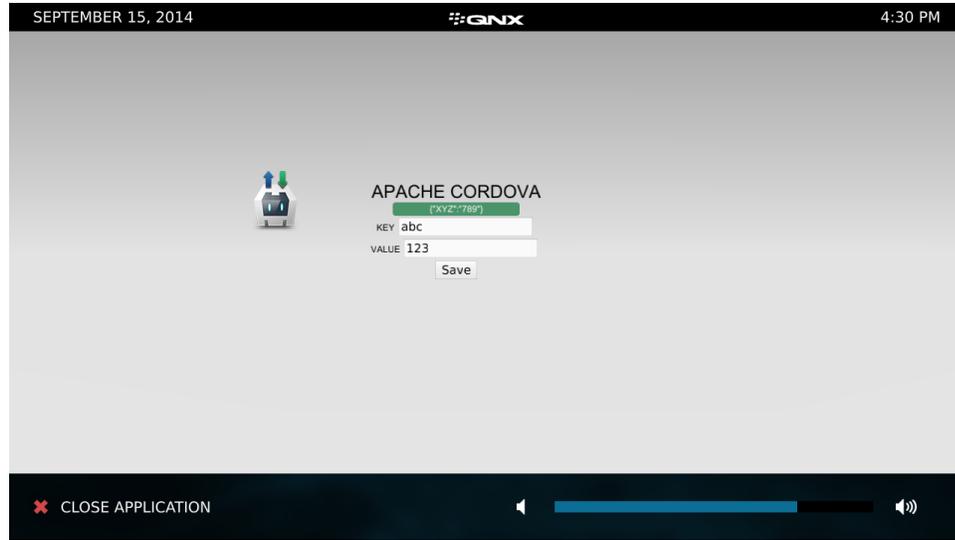
- learn how to use PPS and HTML5 together
- understand the value of using PPS in sharing information between applications or processes
- understand how to set up permissions and access control permissions

To start the application, tap **Cordova PPS Demo** on the Home screen. The app stores values in a PPS object as key-value pairs that you can retrieve across system restarts. After the app starts for the first time, you might see a value of NULL. However, after you've added entries to the PPS object, the key-value pair that you last entered appears.



To add a key-value pair, you tap and type values in the **KEY** and **VALUE** boxes, and then tap **Save**. For example, you can type `abc` for the key and then `123` for the value. When you tap the **Save** button, the app saves the key-value pairs in a PPS object called `/pps/qnx/demo`.

To further demonstrate the sharing of information with another process or application, if you connect to your target with a serial connection, you can execute the command `echo xyz: :123 >> /pps/qnx/demo` and see the key-value pair appear in the green box above the **KEY** box. You can also look at the `/etc/authman/sys.acl` and `/etc/authman/sys.res` files to understand how the Authorization Manager (authman) is used to control access to the PPS object.



Location on the reference image:

`/apps/CordovaPPSdemo.testDev_dovaPPSdemod339185a`

Full Screen HMI

The reference image includes a second HMI, also written with Qt, that demonstrates how to run a single, monolithic HMI without packaging apps into Blackberry ARchive (BAR) files.

The Full Screen HMI has a simple user interface and doesn't use the *Application and Window Management* components, such as the *Authorization Manager (authman)* or *Application Launcher (launcher)*. To display the simple HMI instead of the Home screen, change the following line in the **/var/etc/services-enabled** file using `elvis` or `vi`:

```
QTHOMESCREEN:true
```

to

```
QTHOMESCREEN:false
```

Then, save the **services-enabled** file and restart your board. After the board restarts, the simple HMI showing the QNX logo and the target's IP address appears:



It's expected that developers write their own Full Screen HMI, as the sample provided is intended just to demonstrate how to start Qt runtime and its dependencies (e.g., `graphics/Screen`, `multimedia/mm-renderer`).

Hello World

The Hello World app is a reference sample that shows how to set up an HTML5 project and define HTML5 code to display text.

To start the application, tap **HelloWorld** on the Home screen.



Location on the reference image:

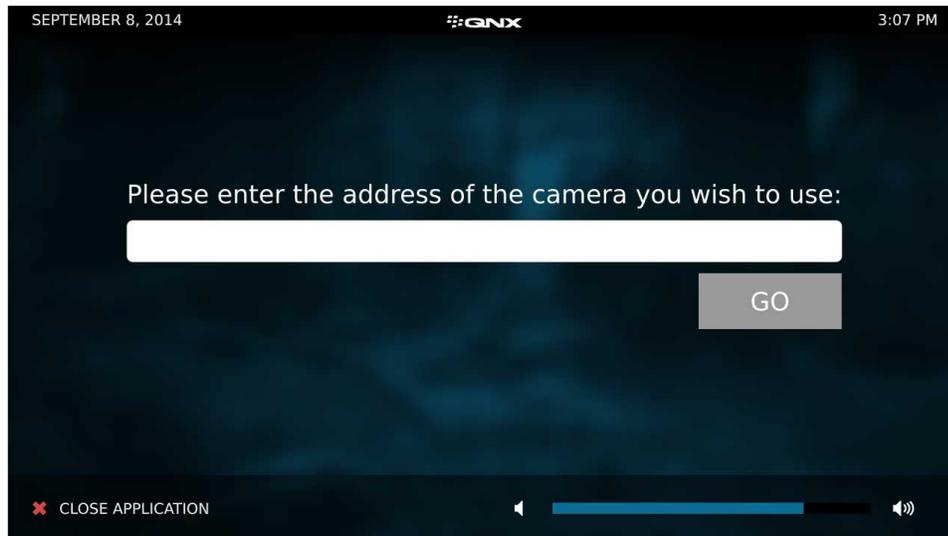
`/apps/helloworld.testDev_helloworld_a520b600`

IP Camera

The IP Camera app demonstrates how to use Qt to play a video feed sent by a remote camera.

This video capture sample uses the multimedia components of the reference image to display a video feed supplied by an RTP/IP-based camera.

To start the app, tap **IP Camera** from the Home screen. A prompt appears, requesting the IP address of a camera to use as the input.



In this release, the IP Camera app supports only Axis cameras. The method of obtaining the IP address of the camera depends on your network setup. Generally, you can connect to your network router or DHCP server and obtain the IP address based on the camera's MAC address. Refer to the Axis documentation (www.axis.com/products/m10_series/index.htm) for additional assistance in obtaining the IP address.

The IP address can be listed in a URL starting with `rtsp:`, which names an RTSP server that delivers video streamed over RTP. You can include a username and password in front of the IP address, for authentication:

```
rtsp://username:password@10.222.97.225/axis-media/media.amp
```

You can also enter an `rtp:` URL that specifies a port for receiving a unicast RTP stream:

```
rtp://@:49152
```

This second URL type is useful if you want to configure a camera to stream to a QNX Neutrino host.

After you enter the address, the app attempts to read video streamed from that address over RTP. If this is successful, the input-prompting screen is replaced with the camera's video feed. When the video feed stops, the app redisplay the IP address prompt. If a media player error occurs, the error name and information are displayed.



RTP is supported only for camera integration, not general networking. You can't access or play video using RTP except when using the IP Camera app.

Considerations:



This app isn't available in VMware images.

The two camera models tested for this release of the QNX SDK for Apps and Media were:

- [Axis M1014](#)
- [AXIS M1034-W](#)

Location on the reference image:

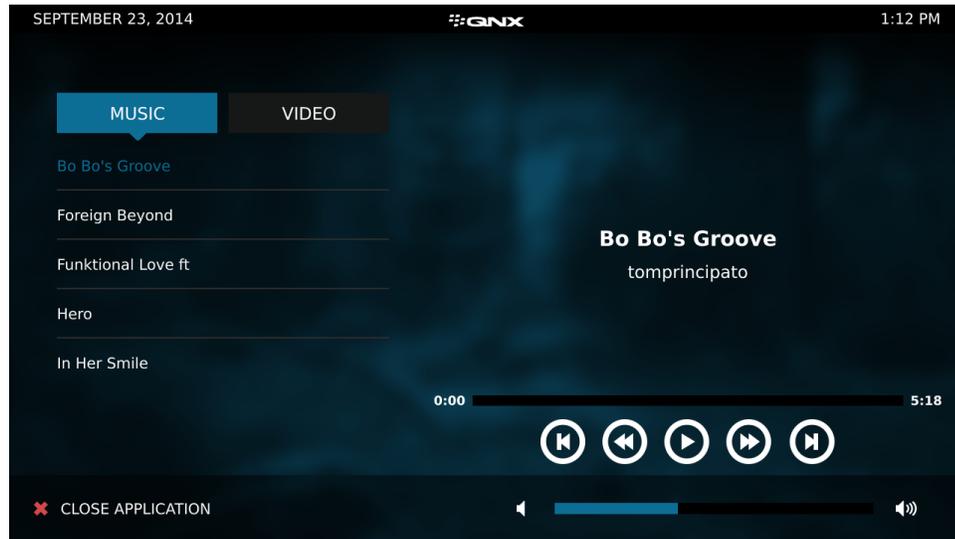
`/apps/com.example.ipcamera.testRel_le_ipcamerad35b63dd`

Media Player

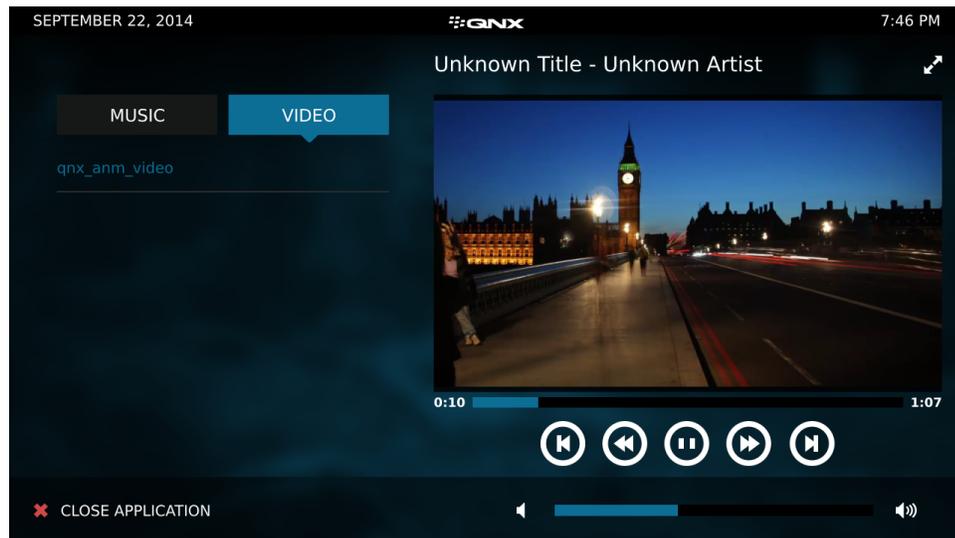
The Media Player app in the reference image is the same multimedia player sample provided in the Qt 5.3.1 distribution (Media Player Example).

This sample uses the standard Qt APIs to browse and play media content stored on local devices. Audio and video playback is supported.

To start the application, tap **MediaPlayer** on the Home screen. When the app runs, the screen lists the available media content on the left side and displays the output window for playback and the playback controls on the right side.



The app provides separate tabs (**Music** and **Video**) that let you browse audio and video content. The **Music** tab is selected when the app starts. Under this tab, the available audio files are listed (the reference image comes with several songs). If you tap the **Video** tab, the available videos are listed (this release provides only one video).



To browse content, swipe up or down to scroll through the list of files. To play a file, tap its list entry (this entry gets highlighted). When playing audio, the application displays the track and artist names in the output window. With videos, the video stream is rendered in this window.

The playback controls are the same for music and videos. The middle button is initially the **Play** button. When you tap it to start playback, this button changes to a **Pause** button. You can then toggle between the pause and play functions.

On both sides of the middle button are the controls for fast-forward and fast-backward. You can tap either button multiple times to increase the playback speed to 2, 4, 8, and 16 times normal speed (for the forwards and backwards directions).

At the left and right ends of the playback buttons are the controls for skipping to the next and previous media files. The **Next** button always skips to the next file, but the **Previous** button can skip to the previous file (if you tap this button before 3 seconds into playback) or restart playback at the beginning of the current file (if you tap this button at 3 seconds or later into playback).

Updating the media libraries:

The audio and video files accessible to the Media Player app are stored in `/accounts/1000/shared/music` and `/accounts/1000/shared/videos`. You can add to the music and video libraries by copying media content from a device to those directories on the target system. You can also delete files from those directories to remove media content. The media file formats supported by this application are given in the *Release Notes* for the QNX SDK for Apps and Media.

The process for updating the media libraries is as follows:

1. Close the Media Player app (if it's open).
2. Add or delete media files in the **music** or **videos** directories as desired.
3. Open the Media Player app to see and play the updated content.



The application does *not* support dynamic media content. New media files won't be listed or played properly if you add them when Media Player is open.

Considerations:

Connect the target board to speakers so that you can hear the audio.



VMware doesn't support video playback.

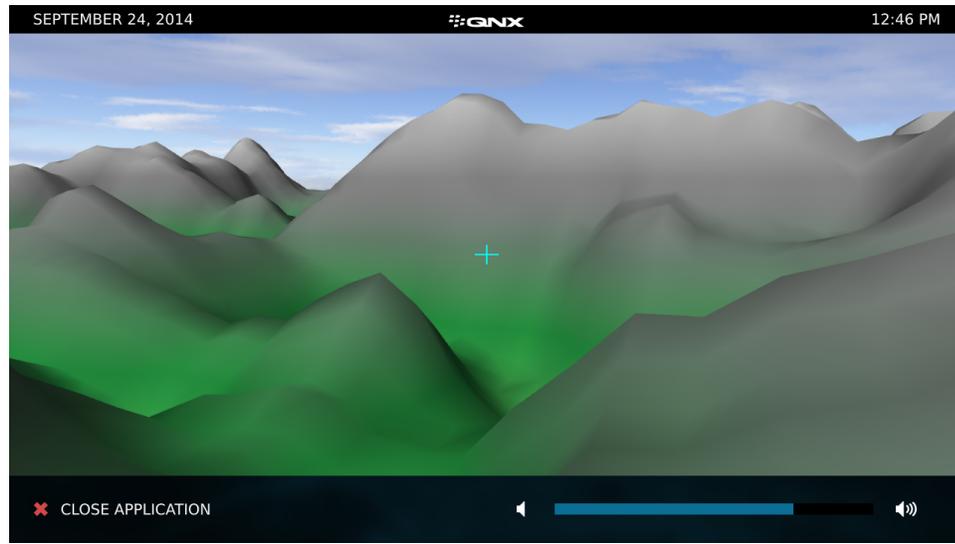
Location on the reference image:

`/apps/com.example.mediaplayer.testRel_mediaplayer178ac554`

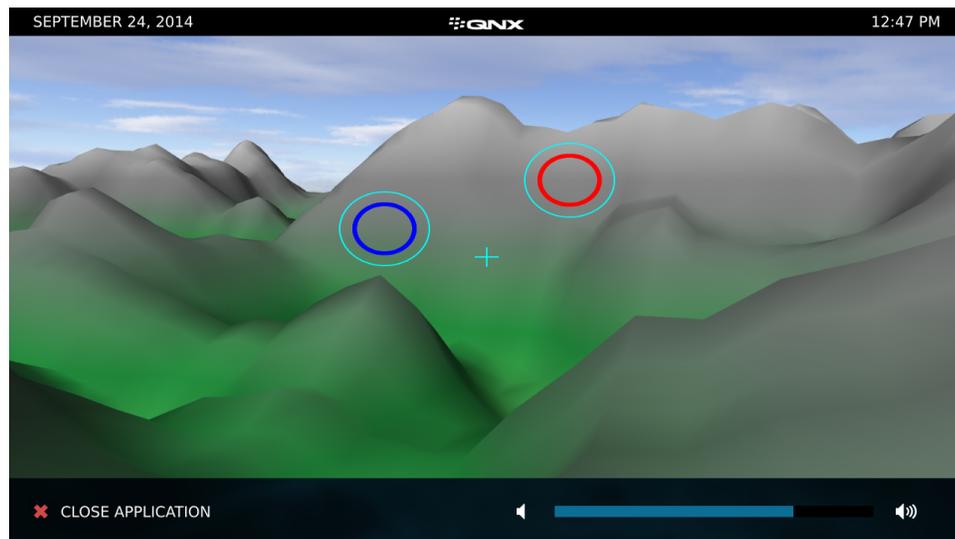
Peaks and Valleys

The Peaks and Valleys app is a reference HTML5 sample that shows you how to use the WebGL API to render 3D graphics and how to navigate the image using HTML5.

To start the application, tap **PeaksAndValleys** on the Home screen. When the app starts, you should see a number of curved lines and shapes representing the terrain. The image you see may differ from the one shown below.



To navigate the image, use one or two fingers to pan, zoom, and change the perspective of what you see. The terrain is navigated based on which vertical half of the screen you touch. If you touch both halves, a blue circle appears on the left and a red circle appears on the right. If you touch only one half of the screen, only one circle appears. The left, blue circle allows you to move forward, backward, left, and right in the terrain. The right, red circle allows you to rotate the camera along the Y and Z-axis of the terrain.



Location on the reference image:

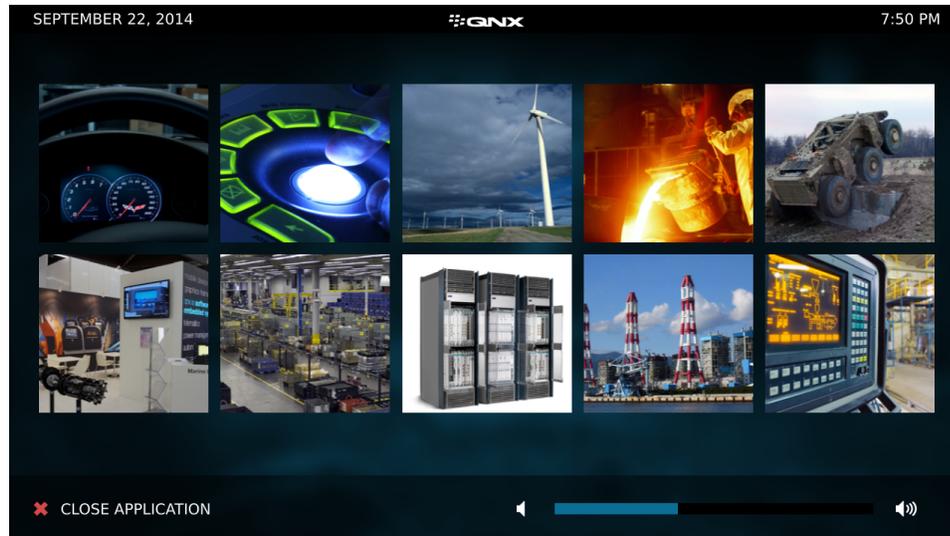
`/apps/PeaksAndValleys.testDev_sAndValleys6bb91d91`

Photo Viewer

The Photo Viewer app in the reference image is the same photo display demo provided in the Qt 5.3.1 distribution (Photo Surface).

This sample uses the standard Qt APIs to access photo files stored in a local folder and to display them.

To start the app, tap **PhotoViewer** on the Home screen. You will then see a set of small images providing previews of the available photos.



The previews are shown in rows of 5 thumbnail images. This release includes 10 sample photos. To display a photo full size (or as large as it can be scaled on the screen), tap its preview image. When a photo is displayed, you can swipe left or right to scroll through the set of photos. If you tap anywhere on the screen again, the app returns to the preview display.

Location on the reference image:

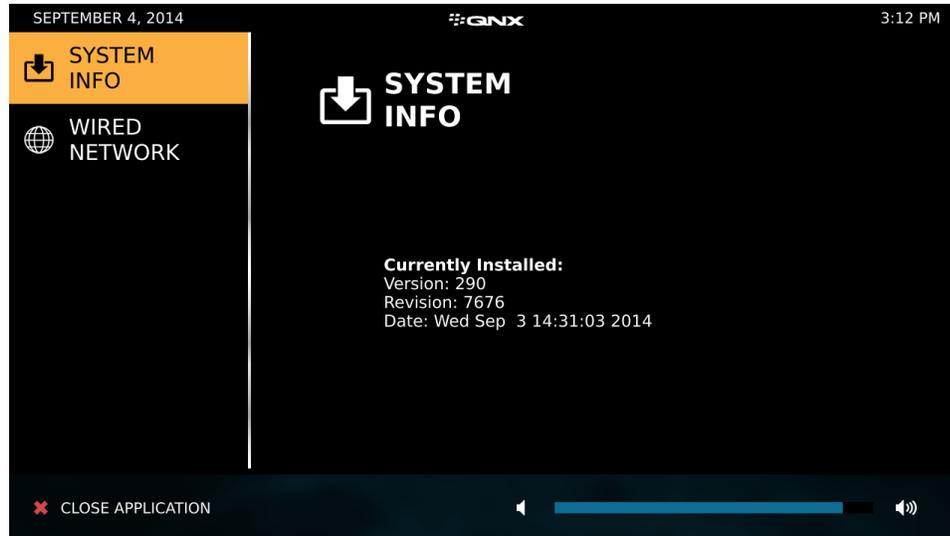
`/apps/com.example.photoviewer.testRel_photoviewer775eb853`

Settings

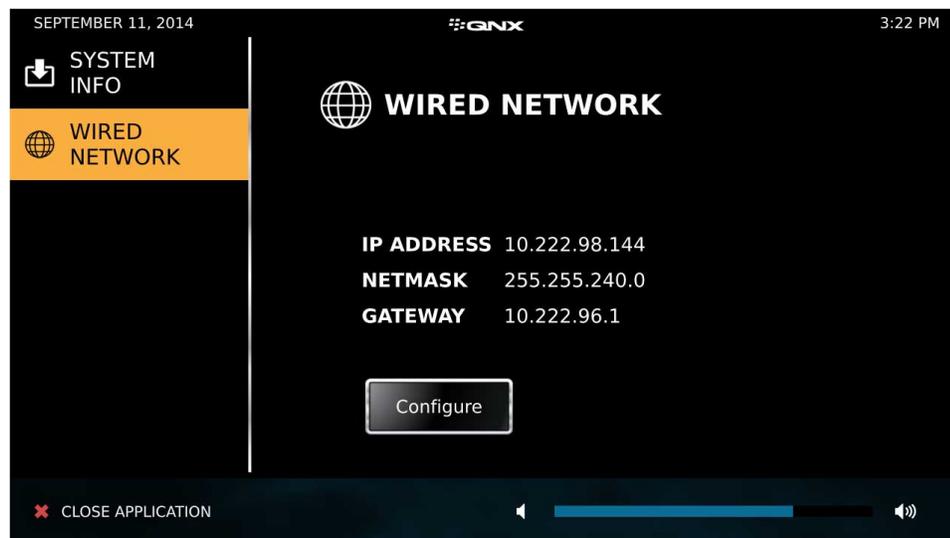
The Settings app is a reference Qt sample that displays system information and configures wired network settings.

To start the application, tap **Settings** on the Home screen. When the app starts, the **SYSTEM INFO** and **WIRED NETWORK** tabs are available. You can tap either tab to access its information or controls.

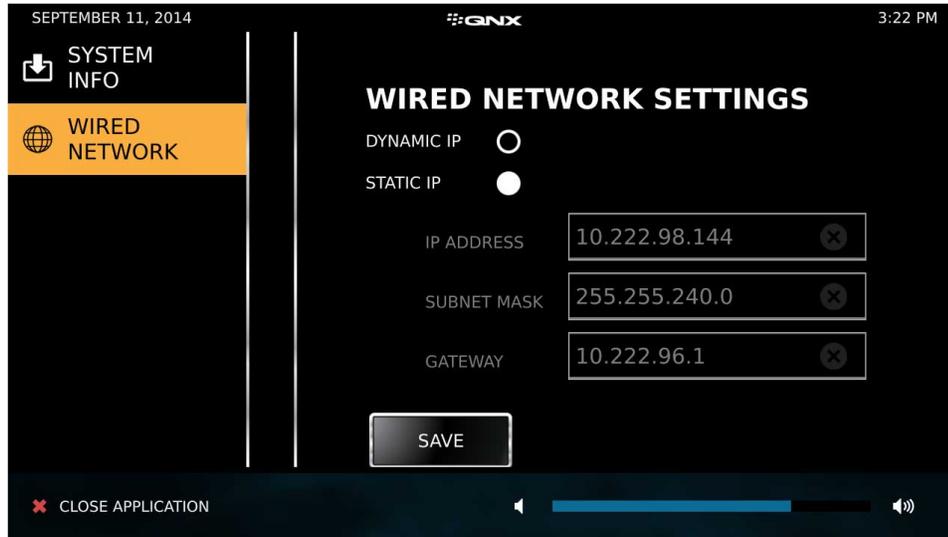
In the **SYSTEM INFO** tab, you can see the version, revision, and date of the system software installed on the image.



If you tap **WIRED NETWORK**, you can view the IP address, network mask, and gateway of the connection to your target board.



You can also tap **Configure** to change your network connection to a dynamic or static IP address. Ensure that you tap **Save** before you close the app; otherwise, the settings aren't applied. You may need to restart your system for the configuration changes to take effect.



Considerations:

Your target board must be connected to a network using an ethernet cable. If the target board isn't connected to a network, you won't see IP address settings.

Location on the reference image:

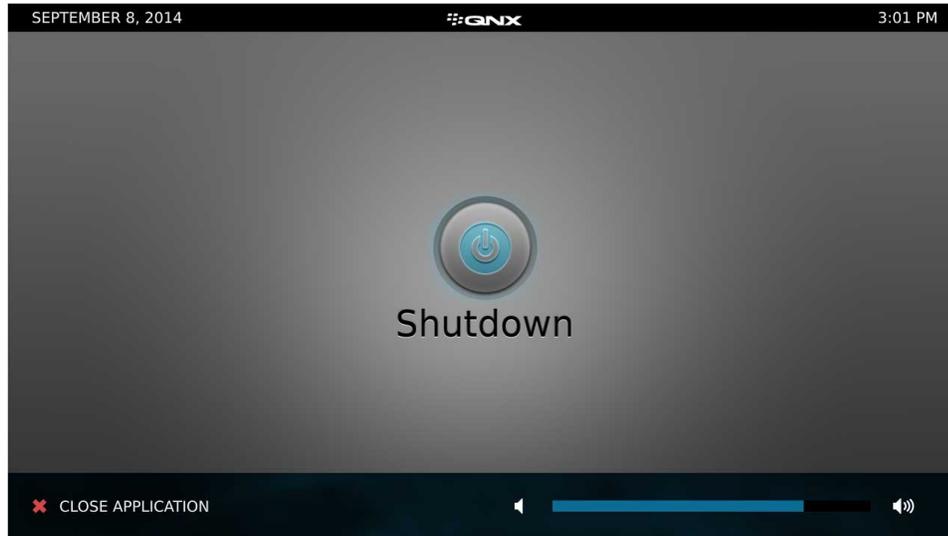
`/apps/Settings.testDev_Settings___595d2043`

Shutdown

The Shutdown app is a reference HTML5 sample that shuts down the system.

This app provides a mechanism to gracefully shut down the system and perform a reboot. Built using the Cordova `qnx.power` plugin, this sample app is useful for understanding how to restart the system using HTML5 controls.

To start the application, tap **Shutdown** on the Home screen. After the app starts, a **Shutdown** button appears. If you tap this button, the application shuts down and restarts the system.



Location on the reference image:

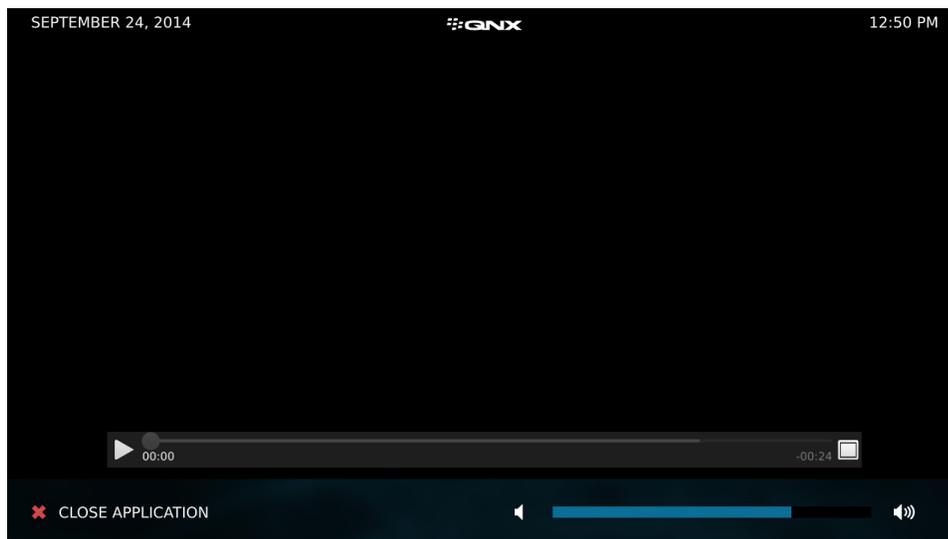
`/apps/Shutdown.testDev_Shutdown___f9c26a76`

Video Demo

The Video Demo app is a reference HTML5 sample that plays a short video.

The app demonstrates how to use the HTML5 `<video>` element to play video.

To start the application, tap **VideoDemo** on the Home screen. After the app has loaded, a screen with a **Play** button and a video progress bar appears.



To play the video, tap the **Play** button. This button changes to a **Pause** button as playback begins. You can toggle between the pause and play functions. After playback starts, you can tap the **Pause** button to stop video playback and adjust the volume using the volume controls at the bottom. You can also tap the **Fullscreen** button (on the right of the video progress bar) to have the video play in full-screen mode. When you're in full-screen mode, you can tap the **Back** button to return to the normal screen size.



Considerations:



This app isn't available in VMware images.

Connect the target board to speakers so that you can hear the video's audio.

Location on the reference image:

`/apps/VideoDemo.testDev_VideoDemo__c6ddf0de`

Index

3D graphics 32

 WebGL 32

A

access 24

 controlling to PPS object 24

access control 24

 setting up permissions 24

adding 24

 key-value pair 24

app 11, 15, 17, 21–22, 24, 27–28, 30, 32, 34–35, 37–38

 Audio Demo 15

 Browser 17

 Browser Lite 21

 Camera 22

 Cordova PPS Demo 24

 Hello World 27

 IP camera 28

 launching from Home screen 11

 Media Player 30

 Peaks and Valleys 32

 Photo Viewer 34

 Settings 35

 Shutdown 37

 Video Demo 38

application management framework 9

Apps and Media 9, 11, 13, 26

 apps 13

 components 9

 Full Screen HMI 26

 Home screen 11

 keyboard 13

 overview 9

 reference apps 13

 sample HMI 11

Audio Demo 15–16

 connecting speakers 15

 reference image 16

 sample app 15

authman 24

Authorization Manager 24

 using to control access to PPS object 24

B

booting 11

 calibrating screen after first boot 11

browser 9

 chrome 9

 chromeless 9

 embedded 9

 in SDK target image 9

 WebKit-based 9

Browser 15, 17, 20

 connecting speakers 15

 location of reference image 15

 network connection 20

 reference image 20

 sample app 17

 starting 17

Browser Lite 21

 network connection 21

 reference image 21

 sample app 21

C

calibrating screen 11

camera 23, 28

 configuring reference image 23

 getting IP address 28

 IP 28

 requirements 23

 RTP 28

 using with Camera app 23

Camera 22–23

 camera support 23

 reference image 23

 sample app 22

chrome 9

 browser 9

chromeless 9

 browser 9

connecting 23

 camera 23

Cordova PPS Demo 24–25

 reference image 25

 sample app 24

F

Full Screen HMI 26
displaying 26

G

graphics 32
sample 3D in app 32

H

Hello World 27
reference image 27
sample app 27

HMI 11, 26
app-based 11
monolithic 26

Home screen 11
about 11
status bar 11

HTML5 15, 24, 38
<audio> element 15
<video> element 38
using with PPS 24

HTML5 sample apps 15, 17, 21–22, 24, 27, 32, 37–38
Audio Demo 15
Browser 17
Browser Lite 21
Camera 22
Cordova PPS Demo 24
Hello World 27
Peaks and Valleys 32
Shutdown 37
Video Demo 38

I

information 24
sharing between processes 24

IP address 28, 35–36
getting for IP camera 28
viewing 35
wired connection 36

IP Camera 28–29
camera support 28
reference image 29
sample app 28

K

key-value pair 24
adding 24

keyboard 13
Apps and Media 13
external 13
software 13

M

media libraries 31
updating 31

Media Player 30–31
reference image 31
sample app 30
updating media libraries 31

mm-renderer 26
starting 26

multimedia 9, 26, 30
components 9
sample player 30
starting 26

music 30
playing 30

N

network 20–21, 35
connection for Browser app 20–21
viewing information about 35

P

Peaks and Valleys 32–33
reference image 33
sample app 32

permissions 24
access control 24
setting up 24

Photo Viewer 34
reference image 34
sample app 34

playing 30, 38
music 30
sample video 38
video 30

PPS 24
controlling access to object 24

PPS (*continued*)

- Cordova demo app 24
- using with HTML5 24

Q

qnx.power plugin 37

Qt runtime 26

- starting 26

Qt sample apps 28, 30, 34–35

- IP Camera 28
- Media Player 30
- Photo Viewer 34
- Settings 35

R

reference apps 13

reference image 15–16, 20–21, 23, 25, 27, 29, 31, 33–34, 36–37, 39

- Audio Demo 15–16
- Browser 20
- Browser Lite 21
- Camera 23
- Cordova PPS Demo 25
- Hello World 27
- IP Camera 29
- Media Player 31
- Peaks and Valleys 33
- Photo Viewer 34
- Settings 36
- Shutdown 37
- Video Demo 39

S

screen 11, 17

- calibrating 11
- resolution 17

Screen 26

- starting 26

Settings 35–36

- reference image 36
- sample app 35

sharing 24

- information with another process 24

Shutdown 37

- reference image 37
- sample app 37

shutting down 37

- Apps and Media 37

software 35

- viewing information about 35

starting 17, 26

- Browser 17
- mm-renderer 26
- Multimedia 26
- QT runtime 26
- Screen 26

status bar 11

- Home screen 11

T

Technical support 8

Typographical conventions 6

U

updating 31

- media libraries 31

USB Video Class (UVC) 23

- camera compliance 23

V

video 30–31, 38

- playing 30, 38
- VMware support for playback 31

Video Demo 38–39

- connecting speakers 39
- sample app 38

viewing 35

- network information 35
- software information 35

VMware 31

- support for video playback 31

W

WebGL 32

- 3D graphics in app 32

Webkit 9

- browser 9

