HTML5 standardization in auto

Andy Gryc, Automotive Product Marketing Manager
Three topics

• Why HTML5 for the car?

• What can QNX contribute?

• Where do we think the W3C should focus?
Why?
Consumer vs OEM lifecycle

- Applications
- Smartphones
- Vehicle electronics

- SOP
- Year 5
- Year 10
Increasing consumer demand

• Car-device connectivity
  – Strong desire for ubiquitous access to online content and communities

• Up-to-date electronics
  – Consumer expectations of car systems influenced by increasingly better mobile devices

• Personalized experience
  – New era of ‘self branding’ apparent everywhere: ringtones, wallpaper, Facebook, etc
HTML5 is a natural choice

Build on something bigger than automotive

- **Ecosystem** – developers, tools, companies
- **Standards** – no vendor lock-in
- **Flexibility** – fluid deployment and architectures
- **Branding** – brand same app on different cars with CSS
- **Lifespan** – will be supported for long time
- **Time to market** – easy to use and leverage
- **Powerful** – rich application environment
- **Cross-platform** – deploy apps on phones and car
What?
QNX in automotive

In 2011

- 60+% of infotainment systems shipped
- 9+ million world wide (5+ million in NA)
- 40+% of all cars sold in US
QNX CAR 2 application platform

BB10 software stack

+ HTML5 UI and frameworks
+ mobile connectivity
+ automotive specific technologies
+ automotive-hardening
+ driver-friendly UX
+ driver distraction sensitivity
QNX CAR 2 feature highlights

**Navigation**
- Elektrobit embedded
- Telenav hybrid
- TCS off-board

**Infotainment + social media**
- Media/AM/FM/HD
- Pandora, Stitcher
- Web browser, YouTube
- Facebook, Twitter

**Automotive**
- Personalization
- Climate control
- Virtual mechanic
- Audio control

**Connectivity**
- Bluetooth + SMS
- MirrorLink + iPod Out
- Smartphone w/ HTML5
- DLNA

**Platform + framework**
- Torch browser
- Composition manager
- HTML5 framework
- App store support
QNX CAR 2 Applications

Sample HMI Designs
QNX CAR HTML5 framework

• Web technology framework for automotive
  – HTML5 environment w/ front-end + apps
    • allows entire user interface to be easily reskinned
    • launcher, controls, apps, behaviour, etc
  – Sencha Touch Mobile and JQuery for widgets
  – QNX-designed infotainment skin

• App store integration
  – designed for downloading app bundles
HTML5 app packaging

Export

Images, icons

Package

Application (.bar)

Commercial design tools
(optional for asset creation)

CSS HTML5 JS

QNX CAR Framework

Config Files

Sencha Touch 2

Application Packager *

* Uses BB WebWorks (project on github) => Apache Cordova
Deploying HTML5 applications

HTML5 Application Environment

HTML5 Application
- Data Files
- Manifest

QNX Application Package
- Launch Info
- Permission List "use_media_engine"

QNX CAR Runtime Environment
- Package Installer
- Application Launcher
- Authorization Manager
- QNX Media Engine

QNX CAR Runtime Environment
- Application Sandbox
- QNX Media Engine
Ripple for QNX CAR
QNX CAR and native access

- Bindings to native
  - general PPS, app launcher, authorization manager
  - graphics & composition manager
    (window management of other processes, etc)
  - SQL database

- Interact + control any app from HTML5
JavaScript access to platform services

**HTML5 applications**

- **JS wrapper**
  - Media control
  - Media detection
  - Track status
- **JS wrapper**
  - Tune
  - Status
  - RDS
- **JS wrapper**
  - Dial
  - Call
- **JS wrapper**
  - PTT
  - Prompts

**Components**

- Multimedia
  - Metadata
  - Codecs
  - Devices
- Radio
  - DSP
- Phone
  - Bluetooth
- ASR
  - Voice Engine
Example JavaScript components

- Framework classes
  - application
  - event
  - keyboard
  - io
  - message
  - notification
  - settings
  - system
  - theme
  - users

- Access classes
  - audio mixer
  - audio player
  - bluetooth
  - hvac
  - media library
  - navigation
  - voice
  - phone
  - radio
  - sensors
Example audioplayer methods

- `setTrackSession(config, index)`;
- `play()`;
- `playAt(index)`;
- `pause()`;
- `stop()`;
- `next()`;
- `prev()`;
- `seek(position)`;
- `setShuffle(mode)`;
- `setRepeat(mode)`;
- `isStopped()`;
Where?
Needed areas of W3C focus

1. Application packaging
   - Ability to move app development across platforms
   - App developers have larger addressable market
   - OEMs require less customization to adopt specific apps
   - Cordova?
Needed areas of W3C focus

2. Native access APIs
   - Ability to consistently access underlying functionality
   - Consistency for app developers and OEMs
   - OEMs will still need ability to customize what’s available where
Needed areas of W3C focus

3. Mobile + car integration
   – Ability to bring mobile hosted apps into auto experience
   – MirrorLink?
Needed areas of W3C focus

4. Distraction prevention and OEM skinning
   - Consistent style sheet tags and mechanisms
   - OEM-approved templates for common use cases
   - With templates, frees app makers from deep understanding of auto concerns
   - Gives OEMs ability to differentiate and create a consistent brand look and feel
Needed areas of W3C focus

5. App development guidelines
   - “How to” for mobile developers entering the automotive space