

rdkbrowser

RDK Browser

- [Summary](#)
- [Webkit and QtWebkit](#)
- [References](#)
- [HTML Support](#)
 - [Supported Features](#)
 - [Unsupported HTML Features](#)
 - [Limitations](#)
- [Flash Plugin Support](#)
 - [Supported Features](#)
 - [Limitations](#)
- [Comparison of QTWebkit CSS Features to other browsers](#)
- [Comparison of RDK QTWebkit HTML5 features to other browsers](#)
- [Video playback, MSE and EME](#)
- [DRMs supported by RDK QtWebkit EME](#)
- [Codecs supported by RDK QTWebkit](#)
- [Codecs NOT supported by RDK QTWebkit](#)
- [Remote Web Inspector](#)
- [Setting up the STB to use RWI](#)
- [Connecting to Remote Web Inspector](#)

Summary

The RDK Browser is a fully functional web browser built into the RDK which allows RDK Set-top Boxes to browse and display 3rd party web pages /apps. The currently deployed RDK Browser is a port of QTWebkit. A future version based on WebkitForWayland is being developed, as QTWebkit is no longer supported.

Webkit and QtWebkit

Webkit is an open source Apple layout engine for rendering HTML and HTML5 in web browsers, based on KHTML and KJS by KDE. It was originally written to work on the Qt cross-platform application framework, but was made toolkit independent by Apple when they took over the development, which they later open-sourced. Primarily written in C++, but with some added Apple Objective-C message/calls, Webkit provides a set of classes to display web content in layout windows, and implements browser features such as hyperlinks, and the navigation of forward and backward within the history.

QtWebkit is a port of Apple's Webkit which utilizes the functionality of the Qt cross-platform application framework, while including the improvements Apple made from the original KHTML and KDE. The RDK platform uses Qt in other areas, such as the Receiver. The version of Webkit used in the RDK version of the browser is 537.21 (as reported by Javascript).

Webkit is a well-known standard for browsers.

Browser	Rendering Engine
Chrome	Webkit 537.36
Safari	WebKit 601.2.7
Edge	WebKit 537.36
Firefox	Gecko/20100101 (iOS version uses Webkit 602.1.50)
RDK QTWebkit Browser	Webkit 537.21

Mobile Browser	Rendering Engine
Chrome (Android)	Webkit 537.36
Chrome (iOS)	Webkit 602.1.50

Firefox (Android)	Gecko/20100101
Firefox (iOS)	Webkit 602.1.50

References

Wikipedia: WebKit

QT Webkit

HTML Support

Supported Features

- HTTPS
- Redirects
- HTTP Cookies (limited support, in-memory only)
- Javascript
- Clipboard (copy-paste)
- WebGL (experimental)
- HTML5 Canvas
- CSS3 (<http://css3test.com/> 48% score)
- CSS3 downloadable fonts
- Websockets
- audio tag (aac, mp3)
- video tag (h.264)
- Flash Player 11.1

Unsupported HTML Features

- uploading files: type=file of input tag
- downloading files
- popup windows

Limitations

There are no software codecs. Hardware support is limited in a number of processing units, which is one processing unit per a video stream and one processing unit per an audio stream.

Flash Plugin Support

Supported Features

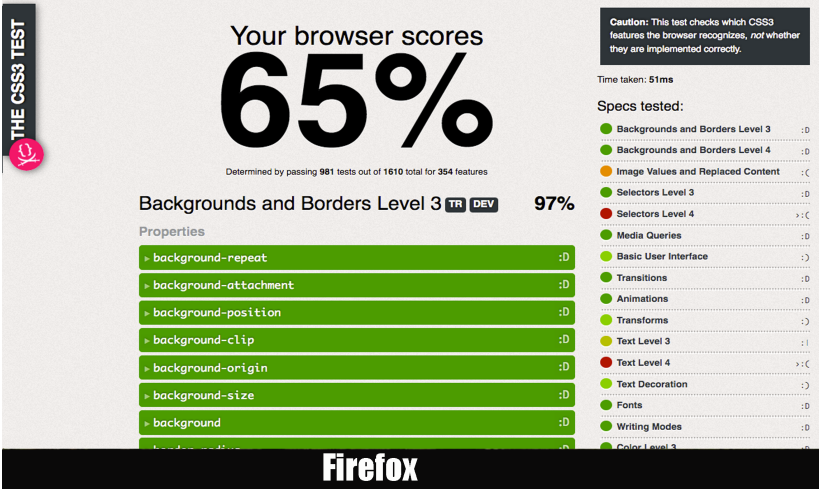
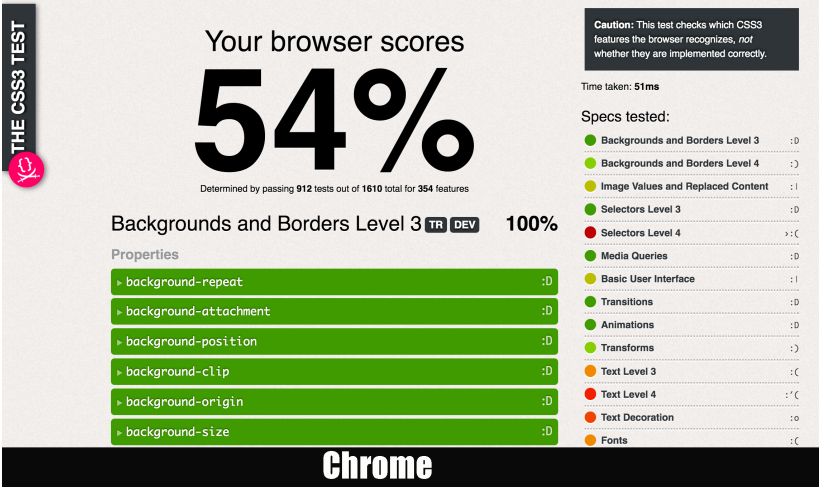
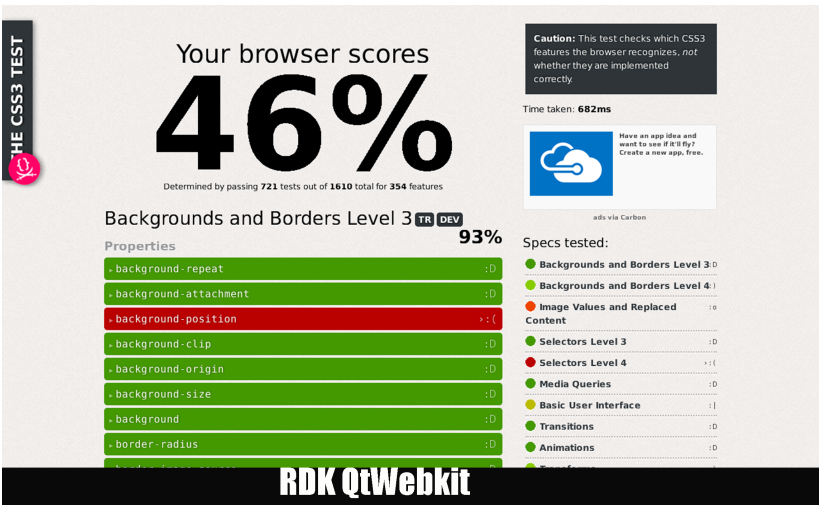
- Flash Plugin 11.1
- SWF (Flash 11.1)
- Video should be limited to H.264 due to presence of hardware decode and rendering support
 - HW decoded video is displayed in a hardware video plane behind the Flash content. Flash content can composite on top of the video. But flash content cannot be composited below video. There is no support for transparency within the video stream.
 - *Other Flash video formats are supported but may not perform adequately due to lack of hardware support.*
- Image file formats are Progressive JPEG, and PNG
- Supported audio codecs are MP3, ADPCM, Nellymoser, Speex and AAC.

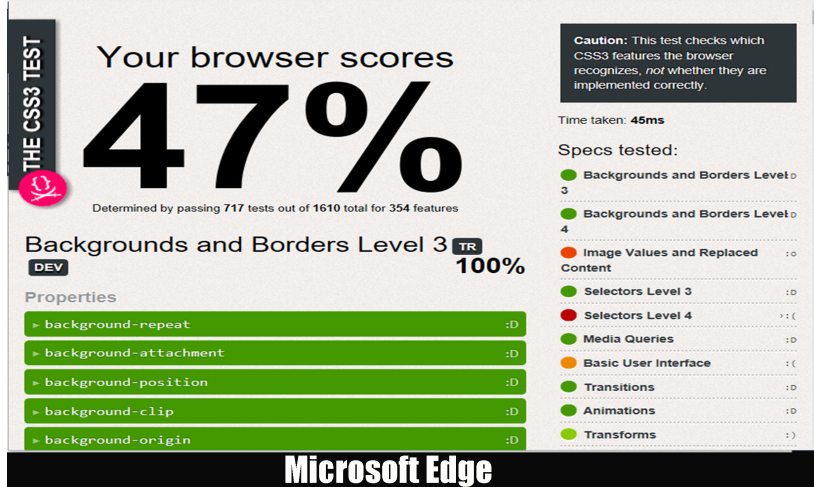
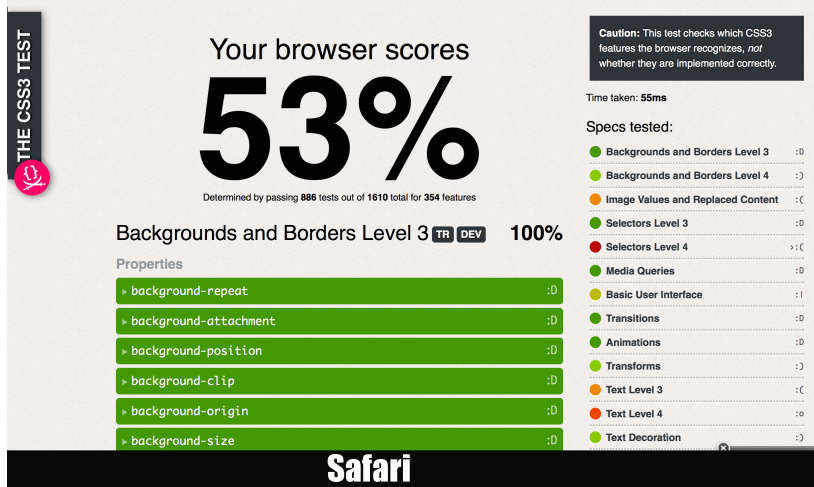
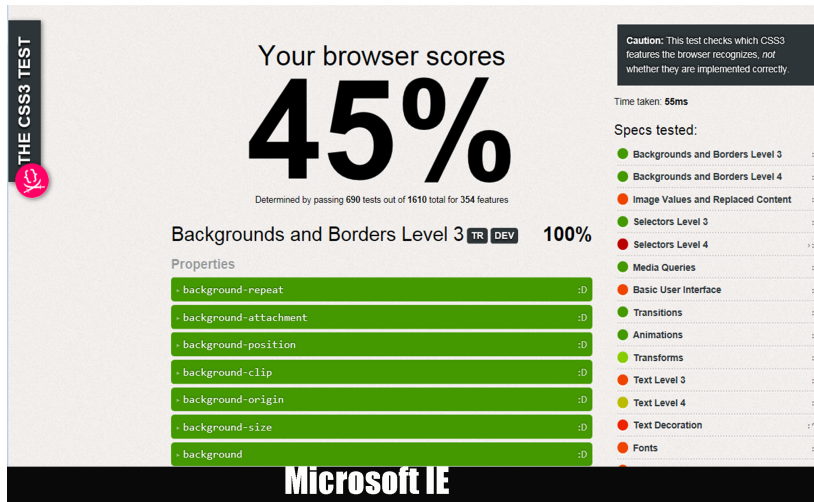
Limitations

There are no software codecs. Hardware support is limited in a number of processing units, which is one processing unit per a video stream and one processing unit per an audio stream.

Comparison of QtWebkit CSS Features to other browsers

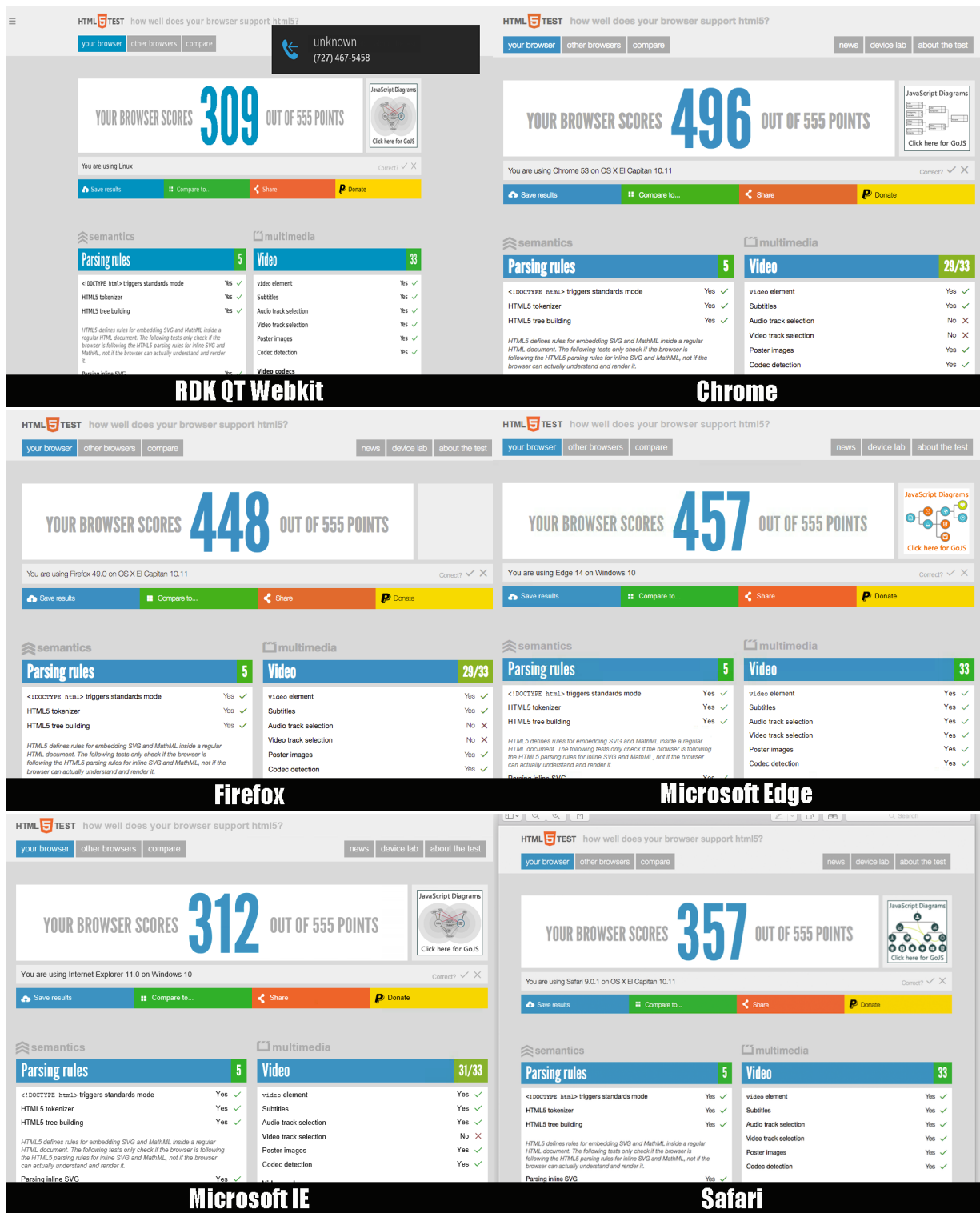
The following is a comparison of the CSS features offered by some of the standard browsers, and the RDK QtWebkit. With the exception of Firefox and IE, the rest of the browsers are all based off of Webkit. A breakdown of the actual features missing can be found in the appendices. These tests are based on the site: <http://css3test.com/>





Comparison of RDK QtWebkit HTML5 features to other browsers

The following is a comparison of the HTML features offered by some of the standard browsers, and the RDK QtWebkit. With the exception of Firefox and IE, the rest of the browsers are all based off of Webkit. A breakdown of the actual features missing can be found in the appendices. These tests are based on the site: <https://html5test.com/>



Video playback, MSE and EME

RDK QTWebkit supports Media Source Extensions (MSE) and Encrypted Media Extensions (EME).

The version of MSE supported is: **MSE Spec Version: 10 December 2013** ([conformance results here](#))

Google MSE conformance test used:

<http://yt-dash-mse-test.commondatastorage.googleapis.com/unit-tests/2016.html?enablewebm=off×tamp=1480519673236>

Currently only **EME version 0.1b** is supported. Playready is currently the only DRM supported.

DRMs supported by RDK QtWebkit EME

DRM	SUPPORTED
Playready	YES
Widevine	NO
ClearKey	NO
Axinom	NO
AES-128	Not natively, may be supported by a specific player

Codecs supported by RDK QTWebkit

Mime Type / Codec	Description
video/mp4;codecs="avc1.42E01E, mp4a.40.2"	H.264 Simple baseline profile video (main and extended video compatible) level 3 and Low-Complexity AAC audio in MP4 container
video/mp4;codecs="avc1.58A01E, mp4a.40.2"	H.264 Extended profile video (baseline-compatible) level 3 and Low-Complexity AAC audio in MP4 container
video/mp4;codecs="avc1.4D401E, mp4a.40.2"	H.264 Main profile video level 3 and Low-Complexity AAC audio in MP4 container
video/mp4;codecs="avc1.64001E, mp4a.40.2"	H.264 'High' profile video (incompatible with main, baseline, or extended profiles) level 3 and Low-Complexity AAC audio in MP4 container
video/ogg;codecs="theora, vorbis"	Theora video and Vorbis audio in Ogg container
audio/ogg;codecs=vorbis	Vorbis audio alone in Ogg container
video/mp4; codecs="avc1.42E01E"	H.264 baseline in MPEG-4 container
audio/mp4; codecs="mp4a.40.2"	AAC low-complexity in MPEG-4 container
audio/mpeg;codecs="mp3"	Audio mp3
video/webm; codecs="vorbis, vp8"	WebM with VP8 video and Vorbis

Codecs NOT supported by RDK QTWebkit

Mime Type / Codec	Description
application/x-mpegurl	HLS x-mpegurl
application/vnd.apple.mpegurl	HLS vnd.apple.mpegurl
video/mp4;codecs="mp4v.20.8, mp4a.40.2"	MPEG-4 Visual Simple Profile Level 0 video and Low-Complexity AAC audio in MP4 container
video/mp4;codecs="mp4v.20.240, mp4a.40.2"	MPEG-4 Advanced Simple Profile Level 0 video and Low-Complexity AAC audio in MP4 container

video/3gpp;codecs="mp4v.20.8, samr"	MPEG-4 Visual Simple Profile Level 0 video and AMR audio in 3GPP container
video/ogg;codecs="theora, speex"	Theora video and Speex audio in Ogg container
audio/ogg;codecs=speex	Speex audio alone in Ogg container
audio/ogg;codecs=flac	FLAC audio alone in Ogg container
video/ogg;codecs="dirac, vorbis"	Dirac video and Vorbis audio in Ogg container
video/x-matroska;codecs="theora, vorbis"	Theora video and Vorbis audio in Matroska container

Remote Web Inspector

The RDK browser has a hidden feature called "Remote Web Inspector." This feature sends information to a remote browser for debugging purposes.

Setting up the STB to use RWI

The STB and Computer/Laptop/Terminal which is displaying the information must be on the same network for Remote Web Inspector to work. There are several ways to accomplish this:

- 1) Use a Moca to Ethernet bridge. This is the easiest solution. The bridge provides an IP address with which to connect to the STB using its Moca IP address, without requiring it to obtain a local one.
- 2) If the STB is a dev box, it is sometimes possible to connect both Computer and STB to a hub, and connect through the local network. DHCP must be enabled on the STB so it may obtain its own IP address, if this is the case.
- 3) If you have access to the STB with root access, you can connect it to a PC that is also connected to a PC, and then use the "ifconfig eth0 x.x.x.x" command to give the STB an IP address that is on the same network as the PC.

If you can ping the PC from the STB, you have a good connection (make sure your PC doesn't ignore pings, of course).

Connecting to Remote Web Inspector

Now that everything else is set up, you can connect to the RWI. Open up either Safari or Chrome > v 5.5 and browse to the following address: `http://<ESTB_IP_ADDR>:9222`

If 9222 doesn't work, try port 9223

A link may be presented, if so, click on it, and you're in!