

rdkbrowser2

- [Overview](#)
 - [How to Launch rdkbrowser2 ?](#)
- [Architecture Diagram](#)
- [rdkbrowser2 Controls](#)

Overview

rdkbrowser2 is the browser application based on WPE, it is integrated with following components which enables it with new features compared with legacy rdkbrowser.

- [injectedBundle](#) : Provides integration Layer for Service manager and Player inside WPE, adds ability to send messages between JavaScript & other application.
- [pxcore-libnode](#) : pxcore - Provides a lightweight Frame-buffer library, pxscene - 2D scene graph API library.
- [wpe-webkit](#) : WebKit porting which is used with Wayland.
- [rdk-logger](#) : Provides console and file level message logging feature to RDK component.
- [westeros](#) : Wayland compositor library that allows applications to create their own Wayland displays, which allows nesting and embedding of third party application

How to Launch rdkbrowser2 ?

rdkbrowser2 can be launched in two different ways:

- Server Mode: `--server` - Launch the browser in a server("tab") mode

It could be directly launched as:

`/usr/bin/rdkbrowser2 --server`

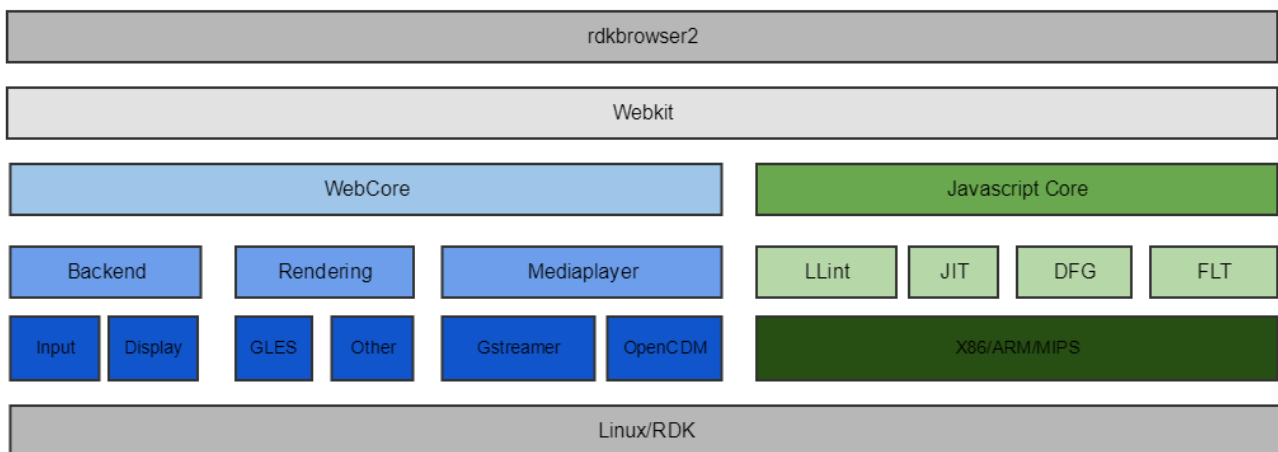
Also can launch `_rdkbrowser2_server.sh`, script file to run rdkbrowser2 as a server.

- Standalone mode: `--url <url>` - Launch opening a specific URL

web pages could be launched using rdkbrowser2 by passing the URL.

`/usr/bin/rdkbrowser2 --url http://www.google.com`

Architecture Diagram



Webkit : Thin layer to link against from the applications .

WebCore : Provides rendering, layout, network access, multimedia, accessibility support etc.

JavaScript Core : Javascript engine.

Platform : Provides platform-specific hooks to implement generic algorithms.

rdkbrowser2 Controls

rdkbrowser2-control is an app for discovering RDKBrowser2 RtRemote Server Object and invoking its RtRemote based API.

Only works in conjunction with /usr/bin/rdkbrowser2.sh

Usage:

start rdkbrowser2: **/usr/bin/rdkbrowser2.sh 'about:blank'**

start **/usr/bin/rdkbrowser2-control** in a separate shell

Sample control sequence:

```
rdkbrowser2tab = rt.locate()
rdkbrowser2tab.transparentBackground = false
rdkbrowser2tab.proxies = [ { "pattern" : "**", "useproxy" : "http://<ip>:<port>" } ]
rdkbrowser2tab.url = "https://www.google.com/"
rdkbrowser2tab.evaluateJavaScript("startTest()")
```