White Paper — *TN002 - QtBrowser Tooling*

Modification Date	*19 February 2015*
Project	*Dawn – UPC Browser integration*
Author	*Paul von Spreckelsen*

Summary

This TN describes the purpose and usage of a number of debug/supporting tools as delivered by Metrological for the Dawn settop box, or box, being the *Web Inspector*, *Garbage Collector* and the *JavaScript Memory Inspector*. These tools are part of the ***qtbrowser*** application.

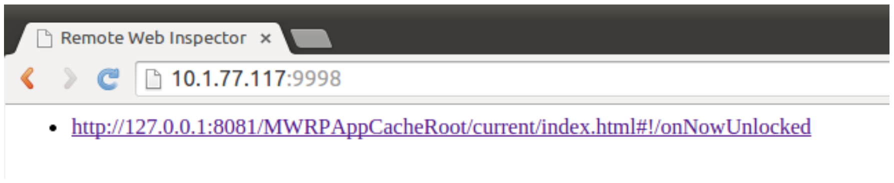
Web Inspector

Description

The Web Inspector enables an user to view the web page source, live DOM hierarchy, script debugging, profiling and more! The Web Inspector resembles the *Inspect element* option (right-click) of the Google Chrome browser (Google Chrome Web Inspector; for more detailed information, see [1]).

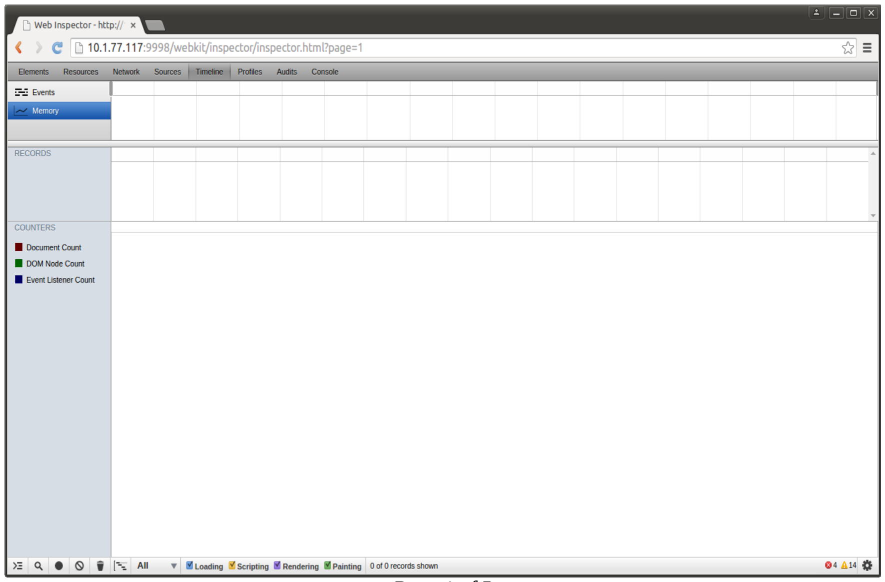
Usage

1. Start ***qtbrowser*** (formerly known as the TntBrowser) with an extra command-line parameter, being: ***—inspector=<port_number>***, for example: *--inspector=9998*
2. Start the Google Chrome browser on a client PC
3. Enter the IP address of the box and the ***port_number*** of the web inspector in the browser URL address field, for example: *10.1.77.117:9998*. See under Tips & Tricks on how-to Obtaining IP address of box for Web Inspector.
4. A single web-page is displayed containing a hyperlink to the output page of the web inspector

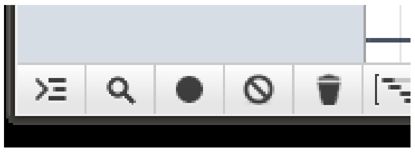


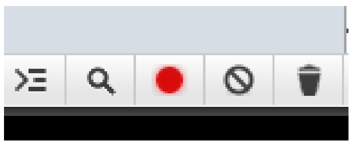
- http://127.0.0.1:8081/MWRPAppCacheRoot/current/index.html#!/onNowUnlocked

5. Clicking on this link will bring up the output of the web inspector (view shown below displays the ***Timeline*** view of the Web Inspector)



6. To start/stop the web inspector press the Record icon in the status bar at the bottom of the page



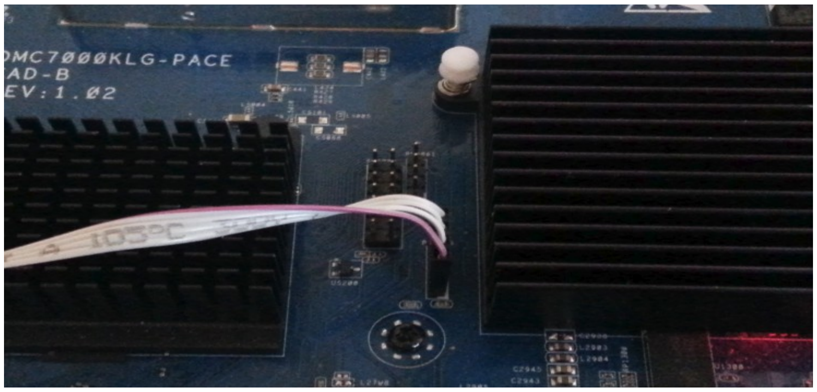


Tips & Tricks

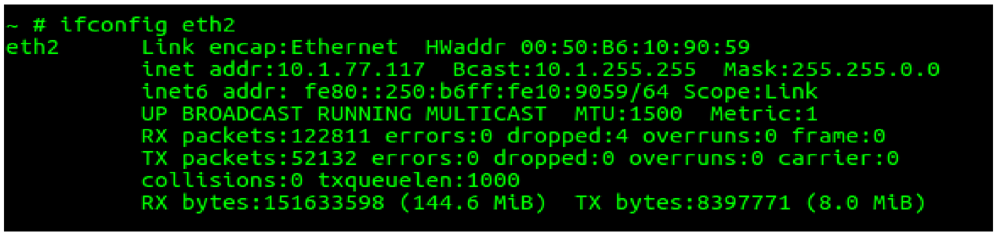
1. Web Inspector can only be used with the Google Chrome browser; other browsers, like Chromium or Firefox, will experience issues.
2. For logging and debugging purposes the box supports an output on an internal connector, which can be viewed by means of a serial terminal application. For Windows, a commonly used serial terminal application is PuTTy (see for example [2] on how-to configure PuTTy for serial communiations); whereas for Linux a commonly used serial terminal application is minicom. (See [3] for more detailed background information on serial communications). In order to view the serial output properly, the following terminal settings need to be set:

• speed/baud-rate:	115200
• data bits:	8
• parity:	N(one)
• stop bits:	1
• Software flow control	No (optional, only change in case of issues)
• Hardware flow control	No (optional, only change in case of issues)

3. Obtaining IP address of box for Web Inspector
   - open the hood of the box and connect a serial cable on the one side to the connector inside the box as shown below, and the other side to a host PC;



- on the host PC, open a serial terminal window to the box; this serial terminal windows enables you to send commands remotely over the serial line to the box by means of a command-line inside the serial terminal window;
- on the command-line in the serial terminal enter: ***ifconfig eth2*** to find the IPv4 address of the box (being the inet addr, as seen in the figure below);
- the IP address obtained is the IP address to be used for the Web Inspector.



```
~ # ifconfig eth2
eth2      Link encap:Ethernet  HWaddr 00:50:B6:10:90:59
          inet addr:10.1.77.117  Bcast:10.1.255.255  Mask:255.255.0.0
          inet6 addr: fe80::250:b6ff:fe10:9059/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:122811 errors:0 dropped:4 overruns:0 frame:0
          TX packets:52132 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:151633598 (144.6 MiB)  TX bytes:8397771 (8.0 MiB)
```

4. The USB port to the side of the box can be used as an Ethernet port (being configured as **eth2**) by means of an USB-to-Ethernet cable. In the Release and Debug build, this port is enabled and not protected by any Firewall. This in contrast to the Ethernet ports at the back of the box which are, by default, protected by means of a firewall. This means that attempting to connect to the Web Inspector may not be possible via these ports, unless the Firewall had been turned off manually. So, in Release and Debug build, connecting to the Web Inspector will be possible from the USB port. Note: in case of a Production build this option is disabled as well!

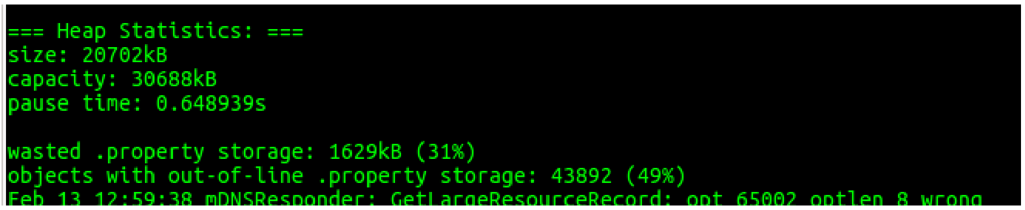
Garbage Collector Statistics

Description

Shows statistics about the garbage collection, current and total heap size, how much was cleared and how long it took the garbage collector to collect and free up the garbage.

Usage

It's usage is triggered by setting an environment variable: ***export JSC_showObjectStatistics=1***. Once set, the garbage collection statistics are printed to screen every 110 seconds, give or take. The output can be seen in a serial terminal window and looks like:



```
=== Heap Statistics: ===
size: 20702kB
capacity: 30688kB
pause time: 0.648939s

wasted .property storage: 1629kB (31%)
objects with out-of-line .property storage: 43892 (49%)
Feb 13 12:59:38 mDNSResponder: GetLargeResourceRecord: opt 65002 optlen 8 wrong
```

where:

• Size	: total size of objects allocated on the js heap after last garbage collection;
• Capacity	: total size of memory allocated for the js heap;
• Pause time	: duration of last garbage collection action;
• wasted .property storage	: freed heap memory during last garbage collection;
• objects with out-of-line	: number of objects having memory allocated on the js heap.

Tips & Tricks

-

Composting Rectangulars

Description

When enabled, the browser will show rectangulars around the viewports being drawn/updated on the display by the compositor. Furthermore, within the rectangular a number is shown indicating the number of copy actions from CPU memory to GPU memory. Typically, this number should be low (in the order of 1 or 2, depending on the operation performed on the data, i.e. moving data around on the display does not require a CPU-to-GPU memory copy, it is something that can be handled perfectly by the GPU itself). Note: copying data from CPU memory to GPU memory are performance impacting operations and should be avoided if not required.

Usage

This tool is enabled by setting the following environment variable:

> ***export WEBKIT_SHOW_COMPOSITING_DEBUG_VISUALS=1***.

JSMInspector

Description

The JavaScript Memory Inspector, being part of the ***qtwebkit***, can be used to track RAM allocations by JavaScript App's.

Usage

TBD

Tips & Tricks

TBD

References

[1] https://developer.chrome.com/devtools/docs/timeline

[2] http://kb.cyberoam.com/default.asp?id=2193

[3] http://en.wikipedia.org/wiki/Serial_port