Consent boxes and advertisements

A page fetched from the web seldom comes alone: a consent dialog lies over it, a cookie bar under it, and advertisements between its paragraphs. By default Weblitz prints all of it, as a browser does. Two options leave it out. Both are off unless you set them, and neither clicks anything.

Since WeblitzPdf 2.0.0-beta.4.

Consent boxes

Consent = `HtmlConsent.Hide` leaves consent boxes out of the print. No button is pressed and no answer is given in the reader's name: the page and its consent platform run as they come, and what they show of it is not printed. It hides:

- the boxes of the consent platforms most sites use (OneTrust, Sourcepoint, Cookiebot, Didomi, Usercentrics, CookieYes and others), by the platforms' own containers;
- what the EasyList Cookie List names, a list of thousands of sites' cookie notices kept by the EasyList authors;
- a box of the page's own: one fixed to the window, or a dialog, of a few lines that speaks of cookies, consent or privacy and holds a button to answer with;
- the veil such a box puts over the page, and the lock it puts on the page's scrolling, which would cut the print to its first page.

What was hidden is said in the result's notes (code `consentHidden`), each of the page's own boxes named as a selector names it. A box no rule knows stays: name it in Hide. A site that shows its content only once consent is given prints what it shows without it.

Advertisements and trackers

Ads = `HtmlAds.Block` prints a page as an ad blocker shows it, by the filter lists EasyList and EasyPrivacy:

- a request the lists block is not made: fewer requests, less script to run;
- an element they hide is not printed;
- a frame or a picture whose request was refused leaves no empty box.

The notes count what was left out (code `adsBlocked`). AdFilters adds rules of your own in the lists' syntax, a line a rule: `||ads.example.com^` blocks the requests to a host, `example.com##.sponsored` hides elements on a site, `@@` and `#@#` make exceptions. A site that refuses readers who block its advertisements prints what it shows them.

The lists, and keeping them fresh

The filter lists are the work of the EasyList authors (easylist.to). The package carries a snapshot of EasyList, EasyPrivacy and the EasyList Cookie List as plain files in `lists/` beside your application, unchanged, with their notice and licences in `lists/NOTICE.txt`. The snapshot is as old as the package, and the lists change every day.

HtmlToPdf.UpdateFilterListsAsync() fetches today's lists and uses them from then on. Call it when your application starts:

```
var lists = await HtmlToPdf.UpdateFilterListsAsync();   // optional, at start
```

It is the only thing that fetches the lists: no render ever does, and nothing runs in the background. The copies are kept in a folder (one of your own if you name it) and read again by later runs; a list is not fetched again before its own refresh interval has passed, and one that cannot be fetched leaves the previous copy, or the snapshot, in use, with the reason in the result.

Without the consent boxes

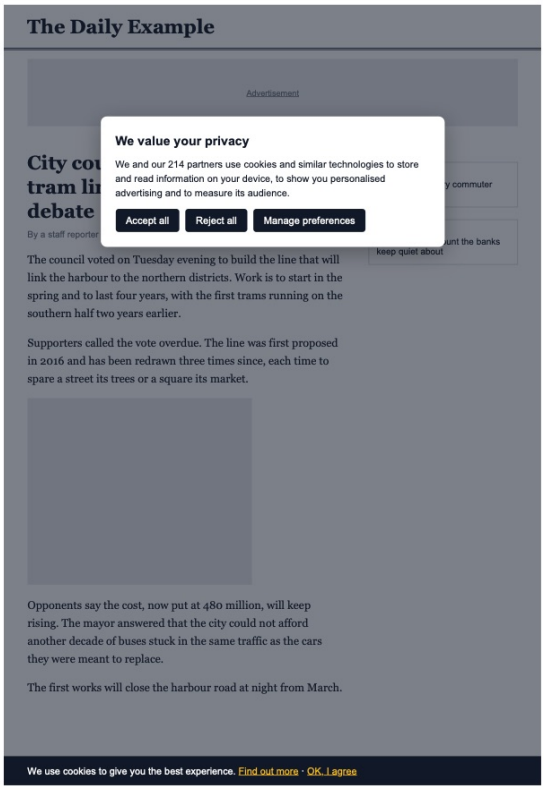
A page from the web often comes with a consent dialog over it and a cookie bar under it. Consent = Hide leaves them out of the print without clicking anything: the page and its consent platform run as they come, no answer is given in the reader's name. It hides the consent platforms' own boxes, the boxes the EasyList Cookie List names, and a box of the page's own (fixed to the window, or a dialog, that speaks of cookies, consent or privacy and holds a button to answer with), with the veil behind it and the lock it puts on the page's scrolling. A consentHidden note says what was hidden.

Reference: `HtmlPdfOptions.Consent`

▶ The page printed: `news-page.html`

Default: the page as it comes

```
var result = HtmlToPdf.Render(html);
```

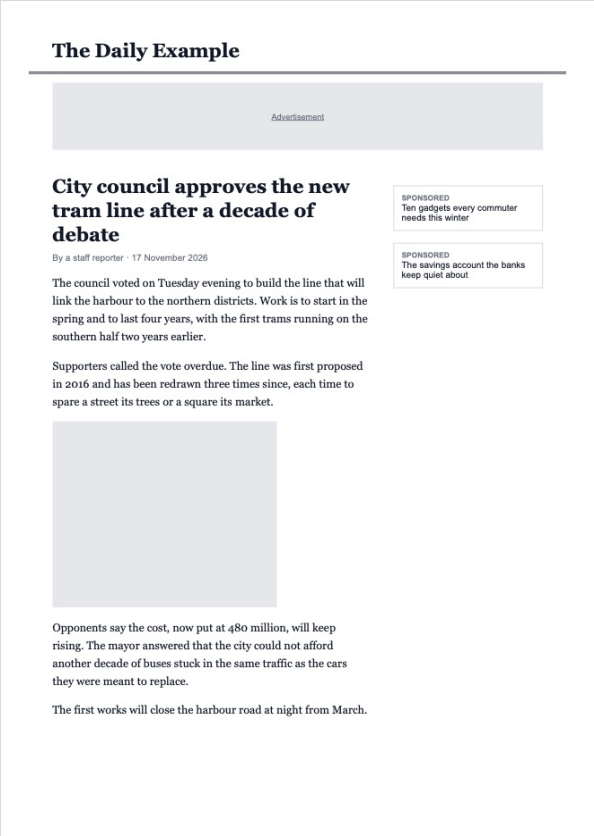


Pages 1
File 100,219 bytes (97.9 KB) · the PDF
Notes

- `Info resourceRefused`: Script https://www.google-analytics.com/analytics.js: no resource loader was given (only data: URLs load).
- `Info resourceRefused`: Frame https://securepubads.g.doubleclick.net/gampad/adx?slot=inline: no resource loader was given (only data: URLs load).
- `Info scriptsRan`: The page's scripts ran (17 ms of virtual time, 2 tasks; 0 canvases drawn).

Consent = Hide

```
var result = HtmlToPdf.Render(html, new HtmlPdfOptions { Consent = HtmlConsent.Hide
});
```



Pages 1

File 70,593 bytes (68.9 KB) · the PDF

Notes

- `Info resourceRefused`: Script https://www.google-analytics.com/analytics.js: no resource loader was given (only data: URLs load).
- `Info resourceRefused`: Frame https://securepubads.g.doubleclick.net/gampad/adx?slot=inline: no resource loader was given (only data: URLs load).
- `Info consentHidden`: Consent boxes are not printed (Consent = Hide): the page's own div#consent, the page's own div#cookie-bar.
- `Info scriptsRan`: The page's scripts ran (33 ms of virtual time, 6 tasks; 0 canvases drawn).

Without advertisements and trackers

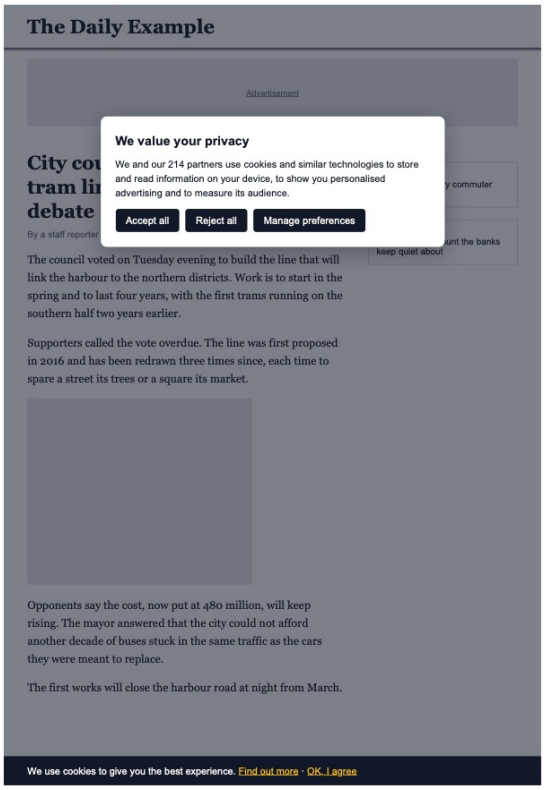
Ads = Block prints a page as an ad blocker shows it, by the filter lists EasyList and EasyPrivacy (the work of the EasyList authors, a snapshot of which the package carries): a request the lists block is not made, an element they hide is not printed, and a frame whose request was refused leaves no empty box. AdFilters adds rules of your own in the same syntax, here for the page's sponsored boxes. An adsBlocked note counts what was left out. Here with Consent = Hide as well.

Reference: `HtmlPdfOptions.Ads`, `HtmlPdfOptions.AdFilters`, `HtmlToPdf.UpdateFilterListsAsync`

▶ The page printed: `news-page.html`

Default: the page as it comes

```
var result = HtmlToPdf.Render(html);
```



Pages 1

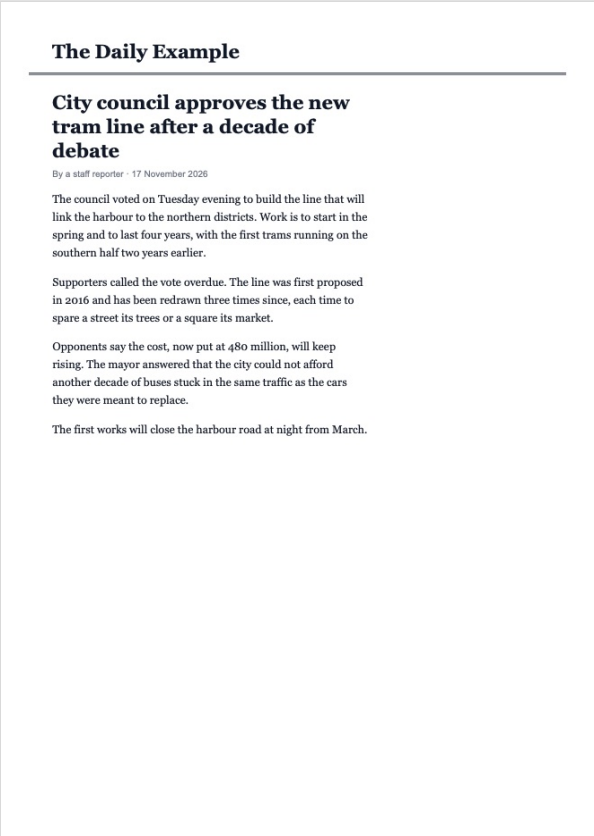
File 100,219 bytes (97.9 KB) · the PDF

Notes

- `Info resourceRefused`: Script https://www.google-analytics.com/analytics.js: no resource loader was given (only data: URLs load).
- `Info resourceRefused`: Frame https://securepubads.g.doubleclick.net/gampad/adx?slot=inline: no resource loader was given (only data: URLs load).
- `Info scriptsRan`: The page's scripts ran (17 ms of virtual time, 2 tasks; 0 canvases drawn).

Ads = Block, with a rule of our own

```
var result = HtmlToPdf.Render(html, new HtmlPdfOptions
{
    Ads = HtmlAds.Block,
    AdFilters = ["##.sponsored"],
    Consent = HtmlConsent.Hide,
});
```



The Daily Example

City council approves the new tram line after a decade of debate

By a staff reporter · 17 November 2026

The council voted on Tuesday evening to build the line that will link the harbour to the northern districts. Work is to start in the spring and to last four years, with the first trams running on the southern half two years earlier.

Supporters called the vote overdue. The line was first proposed in 2016 and has been redrawn three times since, each time to spare a street its trees or a square its market.

Opponents say the cost, now put at 480 million, will keep rising. The mayor answered that the city could not afford another decade of buses stuck in the same traffic as the cars they were meant to replace.

The first works will close the harbour road at night from March.

Pages 1

File 52,292 bytes (51.1 KB) · the PDF

Notes

- `Info consentHidden`: Consent boxes are not printed (Consent = Hide): the page's own div#consent, the page's own div#cookie-bar.
- `Info scriptsRan`: The page's scripts ran (33 ms of virtual time, 6 tasks; 0 canvases drawn).
- `Info adsBlocked`: Advertisements and trackers are left out (Ads = Block): 2 requests not made, 168 of the lists' hiding rules given to the page.