YSLOW

YSlow analyzes web pages, whether web pages are slow or not and suggests ways to improve their performance based on a set of rules for high performance web pages.

YSlow analyzes web page performance by examining all the components on the page, including components dynamically created by using JavaScript. It measures the page's performance and offers suggestions for improvement.

YSlow for Firefox is integrated into the Firebug web development tool for Firefox.

Firefox Addon URL:

https://addons.mozilla.org/en-US/firefox/addon/yslow/
After installation, launch firbug. YSlow will be integrated with firbug.

Feature highlights:

- Grades web page based on one of three predefined rule set or a user-defined rule set;
- It offers suggestions for improving the page's performance;
- Summarizes the page's components;
- Displays statistics about the page;
- Provides tools for performance analysis

Availability

- Firefox
- Chrome
- Mobile/Bookmarklet
- Opera
- Safari
- Command Line (HAR)
- PhantomJS
- Node.js Server
- Source Code

Web Performance Best Practices and Rules

Yahoo!'s Exceptional Performance team has identified 34 rules that affect web page performance. YSlow's web page analysis is based on the 23 of these 34 rules that are testable. Click each performance rule below to see the details.

1. Minimize HTTP Requests
2. Use a Content Delivery Network
3. Avoid empty src or href
4. Add an Expires or a Cache-Control Header

5. Gzip Components
6. Put StyleSheets at the Top
7. Put Scripts at the Bottom
8. Avoid CSS Expressions
9. Make JavaScript and CSS External
10. Reduce DNS Lookups
11. Minify JavaScript and CSS
12. Avoid Redirects
13. Remove Duplicate Scripts
14. Configure ETags
15. Make AJAX Cacheable
16. Use GET for AJAX Requests
17. Reduce the Number of DOM Elements
18. No 404s
19. Reduce Cookie Size
20. Use Cookie-Free Domains for Components
21. Avoid Filters
22. Do Not Scale Images in HTML
23. Make favicon.ico Small and Cacheable

Matrix Table Keys

1. **Rule:** The YSlow performance rule
2. **Weight:** How this performance rule is weighted in the overall page analyis grade
3. **Points**: Number of points deducted per offender (performance infraction occurance), from a total of 100 per rule
4. **Configs:** How many points/threshold per component will be used for computation
5. **Score Computation:** The formula used to compute the final score per rule
6. **Grades from A to F:** How many components/offenders is necessary to reach grades from A to F

YSlow Ruleset Limitations

YSlow generates the page performance summaries by accessing the page components. As of today, browsers allow different levels of access to page components, which limits YSlow’s supported ruleset in each implementation.

Limitations Explained

1. XHRs resolve any redirects on requests, it means when a request is made for url A that gets redirected to B that redirects to C, there's no trace of all redirects involved, thus this rule is never triggered.
2. When scaning the page searching for components, XHRs can't be detected, hence these rules are not triggered.
3. Only cookies sent over HTTP headers are considered and those components hosted in the same domain as the main page.
4. Empty resources are not listed in HAR files.

5. Since there's no real browser involved, a pseudo browser (JSDOM) is used to render the page which does not provide accurate results.
6. No rendered image dimension information is available in HAR files. Inline and style background images can't be distinguished from regular images.
7. Rule is only triggered when favicon is listed in exported HAR files.

Techniques Used to Access Page Components

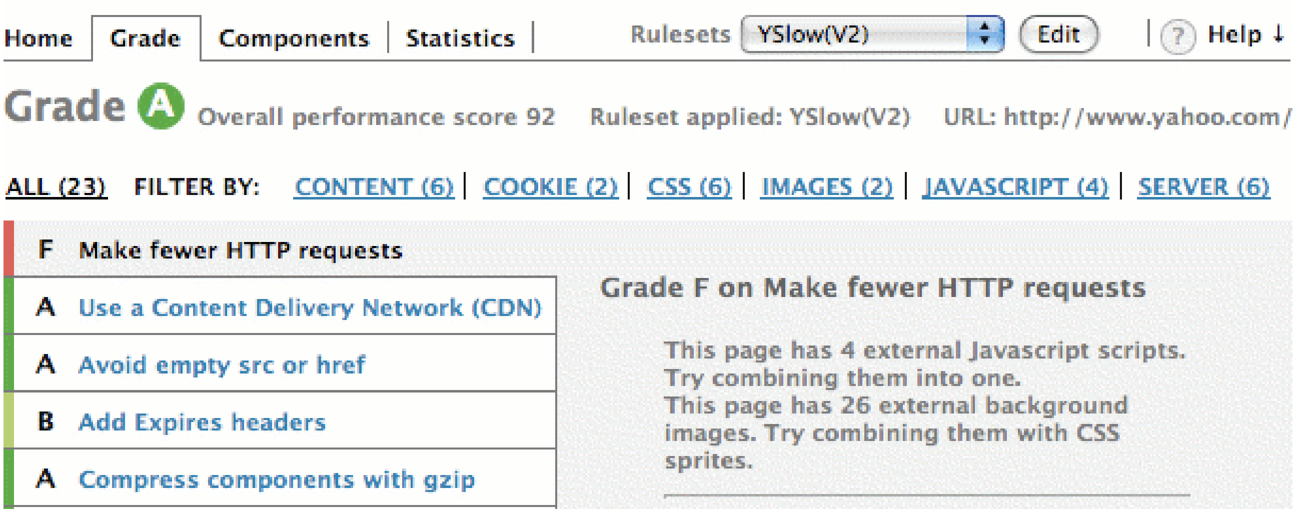
Firefox

It was the original implementation of YSlow and provides full access to page components information throught Firebug Net Panel. Since it's a add-on, Firefox allows cross domain access to iframes and CSS files hosted in different domains than the page domain.

Chrome

Chrome extension API currently does not provide access to the netwok panel data, hence XHR requests are made for each component (script, css, images, etc) found on the page being analyzed. Cross domain XHRs are possible within Chrome extension sandbox environment.

Screenshot



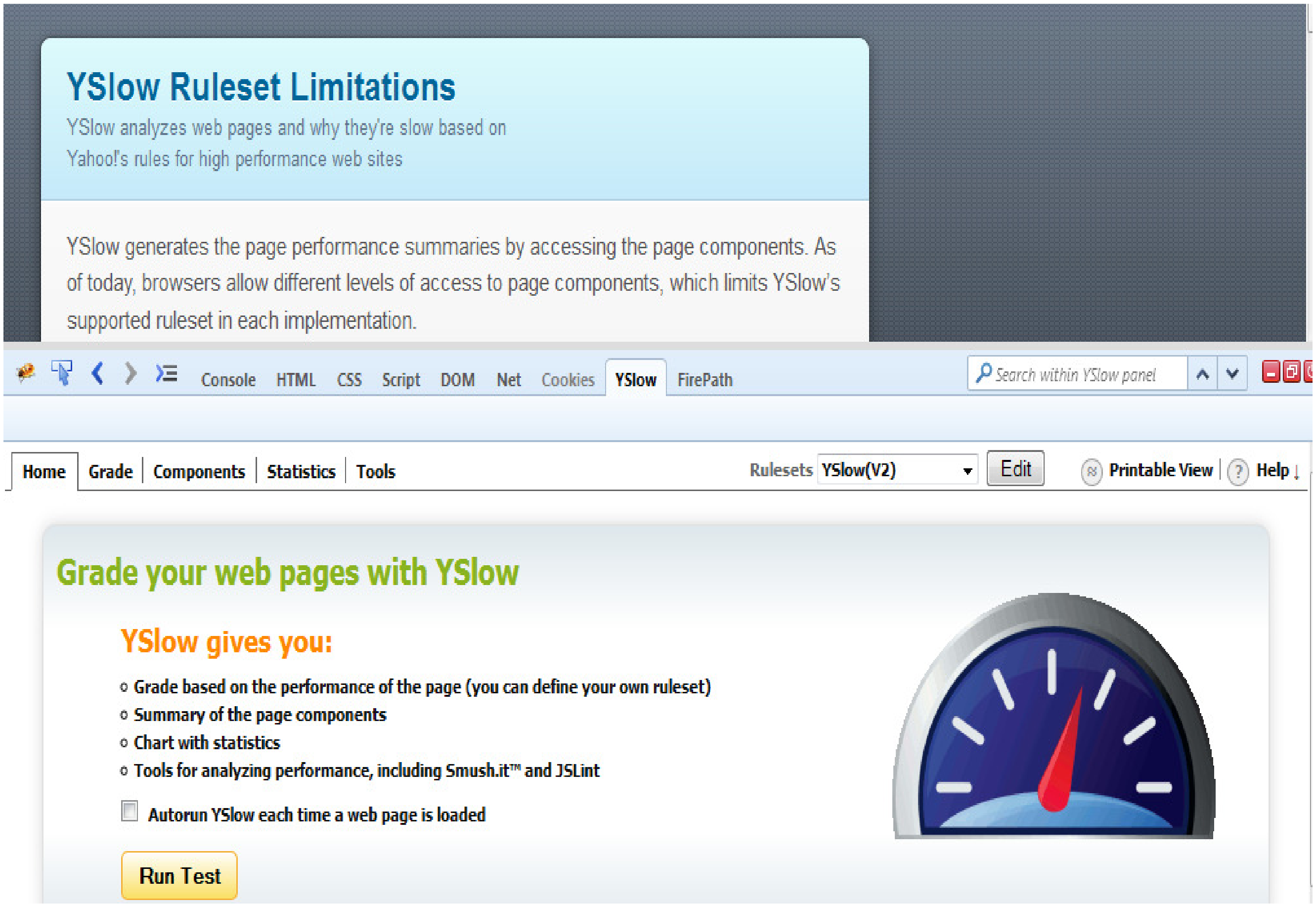
Grade A Overall performance score 92 Ruleset applied: YSlow(V2) URL: http://www.yahoo.com/

ALL (23) FILTER BY: CONTENT (6) | COOKIE (2) | CSS (6) | IMAGES (2) | JAVASCRIPT (4) | SERVER (6)

F	Make fewer HTTP requests
A	Use a Content Delivery Network (CDN)
A	Avoid empty src or href
B	Add Expires headers
A	Compress components with gzip

Grade F on Make fewer HTTP requests

This page has 4 external Javascript scripts. Try combining them into one. This page has 26 external background images. Try combining them with CSS sprites.



YSlow Ruleset Limitations

YSlow analyzes web pages and why they're slow based on Yahoo!'s rules for high performance web sites

YSlow generates the page performance summaries by accessing the page components. As of today, browsers allow different levels of access to page components, which limits YSlow's supported ruleset in each implementation.

Grade your web pages with YSlow

YSlow gives you:

- Grade based on the performance of the page (you can define your own ruleset)
- Summary of the page components
- Chart with statistics
- Tools for analyzing performance, including Smush.it™ and JSLint

□ Autorun YSlow each time a web page is loaded

Questions

- What platforms does YSlow run on?
- How does YSlow work?
- How are the grades computed?
- Does YSlow handle DHTML? Ajax? Web 2.0?
- How does YSlow integrate with Firebug's Net Panel?
- How accurate is YSlow at finding the components in the page? Why do YSlow and Firebug's Net Panel lists differ?
- How is the response time calculated? Why is it sometimes missing?
- How do I add my CDN hostname to YSlow for Rule 2?
- Does YSlow work with frames?
- My component has an Expires or a Cache-Control header - why does YSlow say it doesn't?
- What YSlow results are included in the YSlow beacon?
- What are the license terms that apply to YSlow?

How does YSlow work?

YSlow works in three phases to generate its results.

1. YSlow crawls the DOM to find all the components (images, scripts, stylesheets, etc.) in the page. After crawling the DOM, YSlow loops through Firebug's Net Panel components and adds those to the list of components already found in the DOM
2. YSlow gets information about each component: size, whether it was gzipped, Expires header, etc. YSlow gets this information from Firebug's Net Panel if it's available. If the component's information is not available from Net Panel (for example, the component was read from cache or it had a 304 response) YSlow makes an XMLHttpRequest to fetch the component and track its headers and other necessary information.
3. YSlow takes all this data about the page and generates a grade for each rule, which produces the overall grade.

How are the grades computed?

The grades for individual rules are computed differently depending on the rule. For example, for Rule 1, three external scripts are allowed. For each script above that, four points are deducted from the grade. The code for grading each rule is found in rules.js. The overall grade is a weighted average of the individual grades for each rule, calculated in controller.js The rules are approximately in order of importance, most important first. The specific weights are in the ruleset objects in rules.js.

Rule weights of YSlow V2 Ruleset.

Rule Name	**Rule Id**	**Default weight**
Make fewer HTTP requests	ynumreq	8
Use a CDN	ycdn	6
Avoid empty src or href	yemptysrc	30
Add an Expires header	yexpires	10
Compress components	ycompress	8
Put CSS at top	ycsstop	4
Put Javascript at the bottom	yjsbottom	4
Avoid CSS expression	yexpressions	3
Make JS and CSS external	yexternal	4
Reduce DNS lookups	ydns	3
Minify JS and CSS	yminify	4
Avoid redirects	yredirects	4
Remove duplicate JS and CSS	ydupes	4
Configure ETags	yetags	2
Make Ajax cacheable	yxhr	4
Use GET for AJAX requests	yxhrmethod	3
Reduce the Number of DOM elements	ymindom	3
No 404s	yno404	4
Reduce Cookie Size	ymincookie	3
Use Cookie-free Domains	ycookiefree	3
Avoid filters	ynofilter	4
Don't Scale Images in HTML	yimgnoscale	3
Make favicon Small and Cacheable	yfavicon	2

Does YSlow handle DHTML? Ajax? Web 2.0?

Many web pages are moving to a Web 2.0 style where the components in the page and the page itself are built dynamically in the browser using JavaScript. The main function behind Ajax is XMLHttpRequest. Similar functionality can be achieved using JSON and hidden iframes. YSlow analyzes all the components in the page, including components downloaded using these Web 2.0 techniques. If these dynamically loaded components are missing an Expires header or aren't gzipped they will be reported by YSlow, as expected.

If your Web 2.0 page dynamically downloads components after the onload handler or uses Ajax, these components will be excluded from YSlow grading. To run YSlow on these components, you will need to set extensions.yslow.excludeAfterOnload, *disable* the YSlow Autorun option, and launch YSlow manually after all these requests are done.

1. Go to about:config in Firefox. You'll see the current list of preferences.
2. Enter extensions.yslow.excludeAfterOnload in Filter:.
3. If it does not exist, right-click in the window and choose New and Boolean to create a new Boolean preference. Enter extensions.yslow.excludeAfterOnload for the preference name.
4. Choose false and click OK.

How does YSlow integrate with Firebug's Net Panel?

YSlow uses the components' HTTP response headers to compute a score for each of the performance rules. For example, YSlow uses the Expires header to evaluate **Rule 4**, and the ETag header for **Rule 14**. If necessary, YSlow gets this header information by re-requesting the components using XMLHttpRequest, but this takes time and CPU.

The alternative that YSlow tries first is to get this header information from Firebug's Net Panel. Since YSlow is part of Firebug, it has access to Net Panel's data. Net Panel collects the HTTP header information from the original request for the component, so an additional XMLHttpRequest is not required. YSlow still has to do the XMLHttpRequest if Net Panel doesn't have a component (it was cached) or there was a 304 Not Modified response status code.

YSlow also uses Net Panel to "discover" components that are not part of the DOM (such as XMLHttpRequests) and examines those as well.

How accurate is YSlow at finding the components in the page? Why do YSlow and some packet sniffers differ?

There are two main approaches for finding components in a web page: sniffing packets and crawling the DOM. Firebug's Net Panel, ethereal, and IBM Page Detailer are packet sniffers. YSlow, Instant Source, and the AIS Accessibility Toolbar crawl the DOM.

The problem with sniffing packets is sometimes components are read from the browser's disk cache. If the user forgets to clear their cache, the components shown by the packet sniffer don't reflect all the components in the page. Packet sniffers tell the user only about the components that generated network traffic for the current web page request.

DOM crawlers, on the other hand, report all the components found in the DOM of the current page, whether they were read from disk or downloaded over the Internet. The problem with DOM crawlers is not all HTTP requests show up in the DOM. Specifically, beacons, XHR, and JSON requests aren't found in the DOM.

YSlow is a combination of a DOM crawler and a packet sniffer. YSlow has a built-in DOM crawler that discovers components, but in addition YSlow also integrates with Net Panel to find additional components that are not part of the DOM, such as XMLHttpRequests and image beacons. The Stats view in YSlow gives a breakdown of empty cache versus primed cache component counts, so you can get an idea of the amount of network traffic that would occur under each scenario.

Firebug's Net Panel generally lists only the components that generated network traffic, as expected of a packet sniffer.

How is the response time calculated? Why is it sometimes missing?

The response time shown by YSlow in the status bar is the number of milliseconds between the `onbeforeunload` and `onload` events, in other words, the time between when the previous page unloaded and the requested page was done loading. If you download components after the onload event, those aren't included. Sometimes the response time isn't shown. This typically happens if there was no previous page, such as the first page viewed after the browser starts or after a browser tab was just opened. This also happens if the page being viewed is not an HTML document, such as `about:blank`.

How do I add my CDN hostname to YSlow for Rule 2?

Rule 2 says to use a content delivery network (CDN). The score for this rule is computed by checking the hostname of each component against the list of known CDNs. Unfortunately, the list of "known CDNs" are the ones used by Yahoo!. Most likely these are not relevant to your web site, except for potentially `yui.yahooapis.com`. If you want an accurate score for your web site, you can add your CDN hostnames to YSlow for Firefox and Chrome only, using the button right next to each offender hostname under details. YSlow recalculates scores and grade automatically.

The list of custom CDN hostnames can also be set on Firefox's preferences or Chrome's options. Here are the steps to follow:

Firefox

1. Go to `about:config` on location bar. You'll see the current list of preferences.
2. Right-click in the window and choose New and String to create a new string preference.
3. Enter `extensions.yslow.cdnHostnames` for the preference name.
4. For the string value, enter the hostname of your CDN, for example, `mycdn.com`. Do not use quotes. If you have multiple CDN hostnames, separate them with commas.

Chrome

Right-click YSlow button on extension bar and choose Options. You'll see the current list of preferences.

1. Look for `extensions.yslow.cdnHostnames` preference item.
2. Enter/edit the hostname of your CDN, for example, `mycdn.com`. Do not use quotes. If you have multiple CDN hostnames, separate them with commas.

Firefox and Chrome no longer need to be restarted in order to the list of custom CDNs to become effective. Just hit Run button to recalculate YSlow scores and grade.

If you specify CDN hostnames in your preferences, they'll be shown under the details for Rule 2 in the Performance view.

Does YSlow work with frames?

YSlow loops through all frames (recursively) searching for components. Frames are listed as `doc` components, while iframes show up as `iframe`.