Best Practices on Browser Caching

Browsers can cache content such as images, javascript or css files. Caching resources greatly improves browsing behavior for revisiting users as they do not need to download the same resources again and therefore save on the number of roundtrips to the server and the transfer size.

In order for that to work the browser needs to be told which elements to cache and which not to cache. This can be achieved by specifying HTTP Caching Headers that define how long a resource can be cached by the browser before requesting an updated version of that resource from the web server.

Too often these Cache-Control settings are not set correctly. They are either simply forgotten or just set wrong which leads to unnecessary roundtrips to the web server and redundant downloads that negatively impact end-user browsing experience.

HTTP Caching Headers Explained

HTTP/1.1 provides two headers to specify cache-control settings for the browser:

Expires Headers

The Expires Header allows you to specify an exact date when the validity of the resource expires. Until that date the browser keeps a local copy of the resource without requesting it again from the server. After this date the browser makes a conditional request to the server to check if there is a new version of the resource.

Cache-Control

The Cache-Control Header allows you to specify a ***max-age*** setting in seconds. This forces the browser to keep the resource in cache for the specified amount of seconds from its original request. After this time period is passed the browser makes a conditional request to the server to check if there is a new version of the resource.

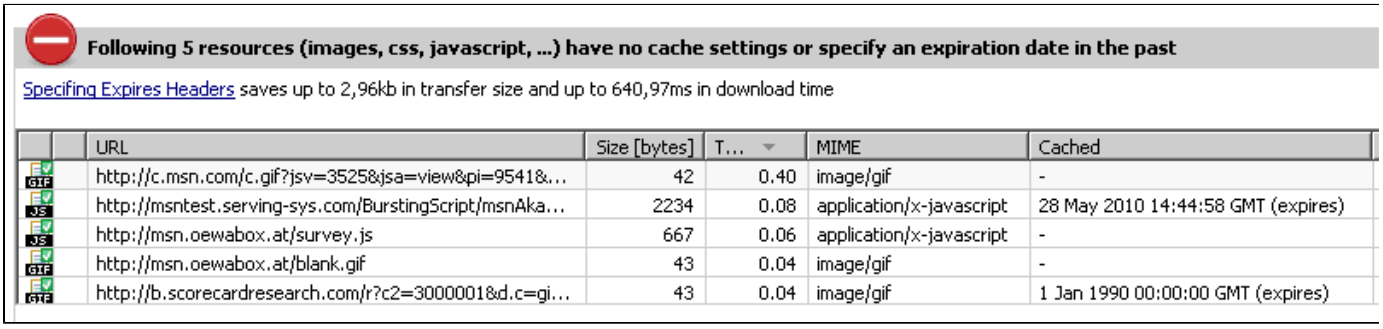
Caching Problem Patterns

No Cache Setting or Expires Date in the past

Resources that have ***no cache*** setting at all (no Expires or Cache-Control) need to be analyzed on whether these resources can potential be cached. Resources that have an Expires header set to a ***date in the past*** need to be analyzed on whether this was done on purpose (to force the browser to reload a resource for every request) or whether it is due to a configuration issue on the web/app server.

dynaTrace AJAX Edition lists all resources with no cache setting or with a date in the past in the first table on the Caching tab of the Performance Report. It analyzes the Expires and Cache-Control:max-age header and calculates the actual expires date.

The Cached column shows whether there was no cache setting ("-") or a date in the past (displaying the exact date). The Cached column also shows which header (Expires or Cache-Control:max-age) was used to set the date.



Following 5 resources (images, css, javascript, ...) have no cache settings or specify an expiration date in the past

Specifing Expires Headers saves up to 2,96kb in transfer size and up to 640,97ms in download time

	URL	Size [bytes]	T...	MIME	Cached
GIF	http://c.msn.com/c.gif?jsv=3525&jsa=view&pi=9541&...	42	0.40	image/gif	-
JS	http://msntest.serving-sys.com/BurstingScript/msnAka...	2234	0.08	application/x-javascript	28 May 2010 14:44:58 GMT (expires)
JS	http://msn.oewabox.at/survey.js	667	0.06	application/x-javascript	-
GIF	http://msn.oewabox.at/blank.gif	43	0.04	image/gif	-
GIF	http://b.scorecardresearch.com/r?c2=30000018&d.c=gi...	43	0.04	image/gif	1 Jan 1990 00:00:00 GMT (expires)

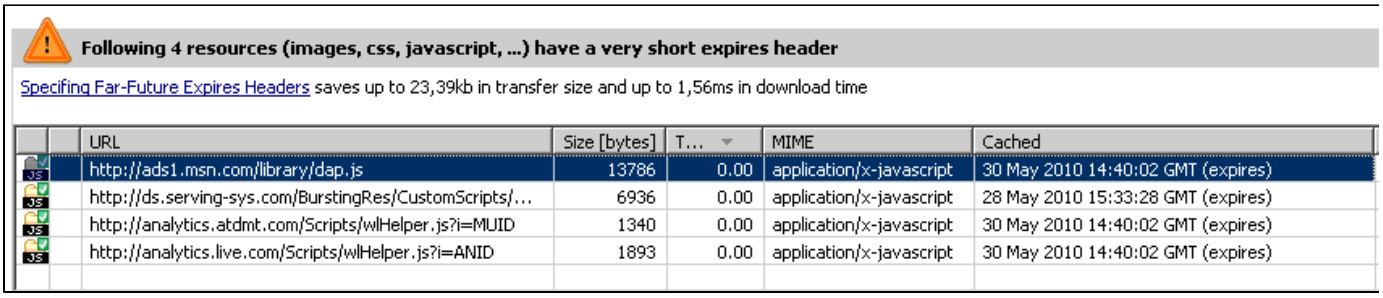
Short Expires Header, ETag or Last-Modified

Resources that are known to not change for a longer period of time can be set with an expiration date in the far future. Due to configuration problems or misunderstanding of how Cache-Control works expiration dates are often set too short, e.g.: only several minutes our hours instead of days, weeks or months. The confusion here often comes from the misunderstanding of the max-age value. ***Max-age sets an expiration timeframe in seconds***. A value of e.g.: 24 means that the resource expires in 24 seconds instead of 24 hours.

With gaining popularity of ETag we updated our analysis to also include resources in that list that have an ETag and/or a Last-Modified Response Header. ETag, Last-Modified and short expiration dates cause the browser to send a conditional request to the web server to validate if there is

new content available or not.

dynaTrace AJAX Edition lists all resources with an expiration date within the next 48 hours, with an ETag or a Last-Modified date in the second table on the Caching tab of the Performance Report. The Cached column shows the cache setting. This can either be the expiration date as specified by either the Expires or Cache-Control:max-age header or it can be an ETag or Last-Modified Header:



Performance Savings, Recommendations and Rank Calculation

The ultimate goal for caching is to reduce the number of roundtrips to the server. No cache settings means that the browser needs to request content all the time for every page request. Short expiration dates cause the browser to send conditional requests. These requests are better than full requests in case the content hasn't changed on the server – but – we still have the extra roundtrips to the server that have to be avoided.

Recommendations and Savings

Analyze all resources on the page. If you know that certain objects won't change for a longer time set an expiration date far ahead in the future to avoid roundtrips from revisiting users.

Don't be redundant with the Expires and Cache-Control Header. Cache-Control overrules the Expires headers and is therefore ignored by the browser.

Caching saves on roundtrips as well as transferred content which ultimately leads to much faster end-user experience for revisiting users.

Rank Calculations

dynaTrace AJAX Edition calculates a rank based on the current cache settings. We acknowledge that not all resources on a page are suitable for caching - so we allow 5 resources of with no cache settings. A page therefore gets a score of 100 if there are fewer than 5 resources with missing cache settings or a setting in the past.
The rank is negatively impacted by the ratio of cached and short-cached resources to the overall number of resources on that page. We also believe that objects with NO cache settings or an expires date in the past are more severe as those with a short expires header as they indicate that cache settings were forgotten or simply done wrong. Therefore we penalize these instances by multiplying the ratio with 1.5.

Note: We do not consider dynamic URLs as being cachable. These are all URLs that have query parameters and all XHR Requests
Note: Due to the factor we use for non cached objects it is possible that more than 100 points get penalized. We would show Rank 0 which corresponds to an F in these cases

Example

Take a page that has a total of 50 resources such as images, css, javascript files. If 10 out of the 50 are not cached at all or have an expiration header in the past the rank is degraded by 15% (5 out of 50 = 10% * Factor 1.5) and goes down to 85. Remember - we do not penalize the first 5 requests here.
If 10 have a short expiration header the rank gets additionally degraded by 20% (10 out of 50). In total this page would have a cache ranking of 65 which corresponds to a D Grade.

Further Readings

Here are further reads and detailed explanation on Cache Settings

- dynaTrace Best Practices on Web Site KPIs, Network Resources, JavaScript/AJAX and Server-Side Activities
- Yahoo's Best Practice Section on Expires Headers and Cache-Control
- Google's Best Practice Section on Cache Optimization
- HTTP/1.1 RFC, sections 13.2, 14.21 and 14.9.3.