

PRADO performance tuning for high-traffic web applications

Based on 2 years' experience with PRADO v.3.x at



*Note: Tips are marked with stars (*), where 5 stars are the most important tweaks.*

I. Basic settings

<http://www.pradosoft.com/demos/quickstart/?page=Advanced.Performance>

a) Application.Mode = "Performance" *

Time of control's templates modification isn't checked (can be set in application.xml)

b) pradolite.php **

Less include() function usage (just change `include("framework/prado.php")` to `include("framework/pradolite.php")`).

c) Set cache class *****

Don't generate all the data on every request. PRADO can save parsed templates, configurations etc. in cache. (can be enabled in application.xml)

d) eAccelerator ****

Now, server parses scripts once, so PHP compiler has less work to do (see: <http://eaccelerator.net>)

II. Advenced settings

a) TOutputCache * *

Not all parts of HTML code on your page need to change every request. Use this to cache content of the selected controls.

b) Memory cache * *

Why not store all PRADO templates and configurations directly in the memory? (read: [Templates in memory - quick guide](#))

c) Generating HTML on your own * * *

If you have looong list of items, don't use *TRepeater* or *TDataGrid*. They are easy to use, but they cost some performance and higher memory usage. (Generate HTML code in your *.php* file - using *foreach()* function by example, and then set output to *TLiteral*).

d) Keep pagestate in your database * * * *

PRADO pagestate can be really big - it's about 10kb-100kb of data. And by default, you send this data on every AJAX request! (just use [TDatabasePageStatePersister](#) class)

e) Sometimes it's better to set PostState of ActiveControl to false *

If you have a control like *TAutoSuggest*, by setting this option, you prevent *PRADO* from gathering all fields on a form values, on every AJAX request. (`ActiveControl.ClientSide.PostState="false"`)

III. Other tips

a) Optimize your code by using TBrowserLog

By adding this line to application.xml

```
<module id="log" class="System.Util.TLogRouter"><route class="TBrowserLogRoute" Levels="DEBUG, INFO, NOTICE, WARNING, ERROR, ALERT, FATAL" Categories="System" /></module>
```

You can track performance on some of your functions directly in browser.

b) Enable gZip compression in your server

Lower bandwidth usage, faster page load for users.

c) Enable query_cache if your are using MySQL

Read: <http://dev.mysql.com/doc/refman/5.0/en/query-cache.html>

d) Performance for end user

Remember, that server' performance isn't most important. Your visitors must feel, that the site is loading fast.

Read: <http://developer.yahoo.com/performance/rules.html> .

Topics index from that article:

1. [Make Fewer HTTP Requests](#)
2. [Use a Content Delivery Network](#)
3. [Add an Expires Header](#)
4. [Gzip Components](#)
5. [Put Stylesheets at the Top](#)
6. [Put Scripts at the Bottom](#)
7. [Avoid CSS Expressions](#)
8. [Make JavaScript and CSS External](#)
9. [Reduce DNS Lookups](#)
10. [Minify JavaScript](#)
11. [Avoid Redirects](#)
12. [Remove Duplicate Scripts](#)
13. [Configure ETags](#)
14. [Make Ajax Cacheable](#)

Any comments are welcome.

Mariusz Zurawek, Poland, December 2007

Author: Mariusz Żurawek

Article downloaded from page eioba.com