

## Wt - Feature #5308

### Possibility of setting a different thread\_pool count for main process and session processes, or detect if I'm a session process or not

09/29/2016 04:57 PM - Aarón Bueno

|  |                 |                        |            |
|--|-----------------|------------------------|------------|
| <b>Status:</b>   | Closed          | <b>Start date:</b>     | 09/29/2016 |
| <b>Priority:</b>   | Normal          | <b>Due date:</b>       |            |
| <b>Assignee:</b>   | Michiel Derhaeg | <b>% Done:</b>         | 0%         |
| <b>Category:</b>   |                 | <b>Estimated time:</b> | 0.00 hour  |
| <b>Target version:</b>   |                 |                        |            |
| <b>Description</b>   |                 |                        |            |
| <p>(Sorry for my English) At least when using the wthttpd connector in dedicated process mode, the configured thread-pool size (wt_config.xml, wthttpd config file or command line parameter) is the same for the main and any of the session processes.</p> <p>However, the main process is outside of user control, but any session process is where user code lives, and perhaps the user wants to customize its own thread-pool, for any reason (for example, to shrink the thread-pool to reduce parallelism, if he is unsure on synchronization issues). Of course, you can set programatically the thread-pool size through <code>`server.ioService().setThreadCount(your_size)`</code>, but that sets the <code>`thread-pool`</code> for main and session servers, again.</p> <p>My "trick" to config the <code>`thread count`</code> separately was to check who am I, whether a main process or server process, looking for the <code>`---parent-port`</code> parameter:</p> <pre>Wt::WServer server(argc, argv, WTHHTTP_CONFIGURATION);  std::string arg(argv[argc - 1]);  auto* test = "---parent-port";  // If I'm a session process  if (arg.compare(0, strlen(test), test)) {  std::string thread_count;  server.readConfigurationProperty("session-process-thread-count", thread_count); // I added it as in my wt_config.xml  server.ioService().setThreadCount(std::stoi(thread_count));  }</pre> <p>My feature request can be any of:</p> <p>*) Adding an extra configuration option to wt_config.xml, for example: <code>`dispatcher-thread-pool`</code>, as suggested by Koen Deforche by email. Only if present, it's used as thread-pool size of session processes.</p> <p>*) Alternately, or in addition, a set of functions to know if I'm a main or session process, connector type, and so on, to allow the user makes any kind of customizations according to general server/process properties. I would do it through a WServer member function returning an object of a <code>`WServerInfo`</code> class or something like that. In the case of the wthttpd connector, it could be implemented through a wrapper class of <code>http::server::Configuration</code> and/or <code>Wt::Configuration</code> with only getters.</p> <pre>// connector_type is a enum of possible connectors: httpd, fcgi, isapi  ConnectorType WServer::connector_type() const;  const WServerInfo&amp; WServer::serverInfo() const;</pre> <p>Initially, that <code>Wt::WServerInfo</code> class can have just two or three function for very basic server information: whether dedicated/shared mode, if "I'm" a main or child server, and so on. If more info is needed, the <code>`WServer`</code> remains untouched; you only need to change the <code>`WServerInfo`</code> interface.</p> |                 |                        |            |
| <b>History</b>   |                 |                        |            |

**#1 - 04/21/2017 01:43 PM - Roel Standaert**

- Status changed from *New* to *InProgress*

- Assignee set to *Michiel Derhaeg*

I'm not sure about the cleanest way of checking in the child process whether it is a child process or not, but we could definitely add that configuration option.

**#2 - 05/04/2017 01:36 PM - Michiel Derhaeg**

- Status changed from *InProgress* to *Implemented @Emweb*

Added a new `wt_config.xml` option.

**#3 - 05/04/2017 04:14 PM - Roel Standaert**

- Status changed from *Implemented @Emweb* to *Resolved*

**#4 - 08/16/2017 10:07 PM - Roel Standaert**

- Status changed from *Resolved* to *Closed*