

Scripting custom queuing algorithms

This is done using a script file plug-in written in Perl. The qb.conf setting for this is as follows:

```
supervisor_queue_binding = Perl
supervisor_queue_library = plugin_path
```

This document isn't meant to be a tutorial on the Perl scripting language so we will assume developers modifying this module in the following documentation already have a functional knowledge of Perl's syntax. Good online resources for learning Perl can be found at <http://www.perl.com> there also many good published books such as *Learning Perl* published by O'Reilly which can get a beginning developer started with Perl scripting.

To create a new queuing algorithm, the developer must create a Perl module with the following Perl functions:

1. qb_init()
2. qb_starthost_init()
3. qb_jobcmp(\$joba, \$jobb, \$host)
4. qb_starthost_deinit()
5. qb_startjob_init()
6. qb_hostcmp(\$hosta, \$hostb, \$job)
7. qb_startjob_deinit()
8. qb_rejectjob(\$job, \$host)
9. qb_preemptjob_init()
10. qb_preemptcmp(\$runningjob, \$candidatejob, \$host)
11. qb_preemptjob_deinit()

A function if not defined will automatically be overridden with the "Priority Queuing Algorithm" defaults.

Example scripts can be found the following locations:

- **Linux:** /usr/local/pfx/qube/examples/supervisor
- **Windows:** C:\Program Files\pfx\qube\examples\supervisor
- **OS X:** /Applications/pfx/qube/examples/supervisor

The functions defined are all used to control the queuing algorithm's properties.

- qb_hostcmp(\$hosta, \$hostb, \$job)
- qb_init()
- qb_jobcmp(\$joba, \$jobb, \$host)
- qb_preemptcmp(\$runningjob, \$candidatejob, \$host)
- qb_rejectjob(\$job, \$host)
- qb_starthost_deinit()
- qb_starthost_init()
- qb_startjob_deinit()
- qb_startjob_init()