Product Configuration

Support by Time module **WHMCS**

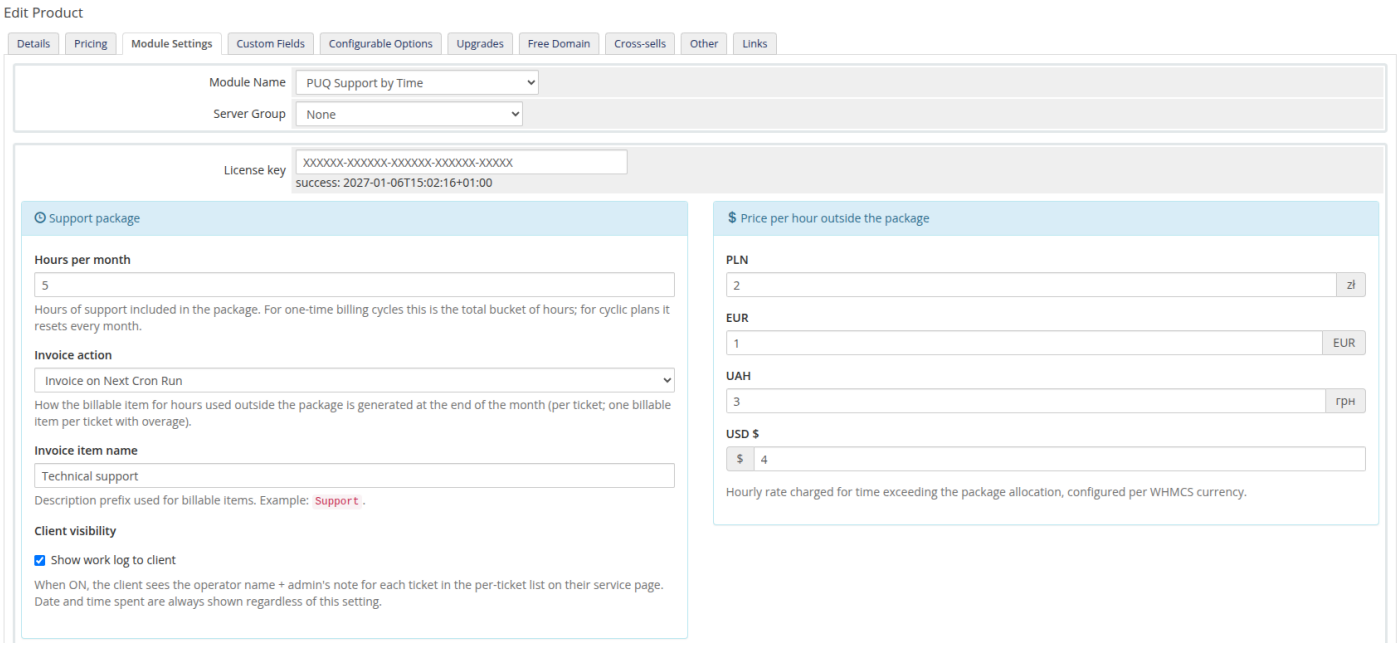
Order now | Download | Community

Add a new product to WHMCS

Navigate to **System Settings** → **Products/Services** → **Create a New Product**.

In the **Module Settings** tab select the **PUQ Support by Time** module (leave **Server Group** as *None* — this is a server module without a remote node). Two billing modes are supported, determined by the product's WHMCS billing cycle:

- **Recurring billing cycle** (Monthly, Quarterly, Annually, ...) — the included hours reset every month and overage hours are billed automatically by the daily cron at the end of each month.
- **One Time** — the included hours are a fixed bucket that the service consumes ticket by ticket. When all hours are used, the service is automatically terminated by the next cron run.



05-product-config.png

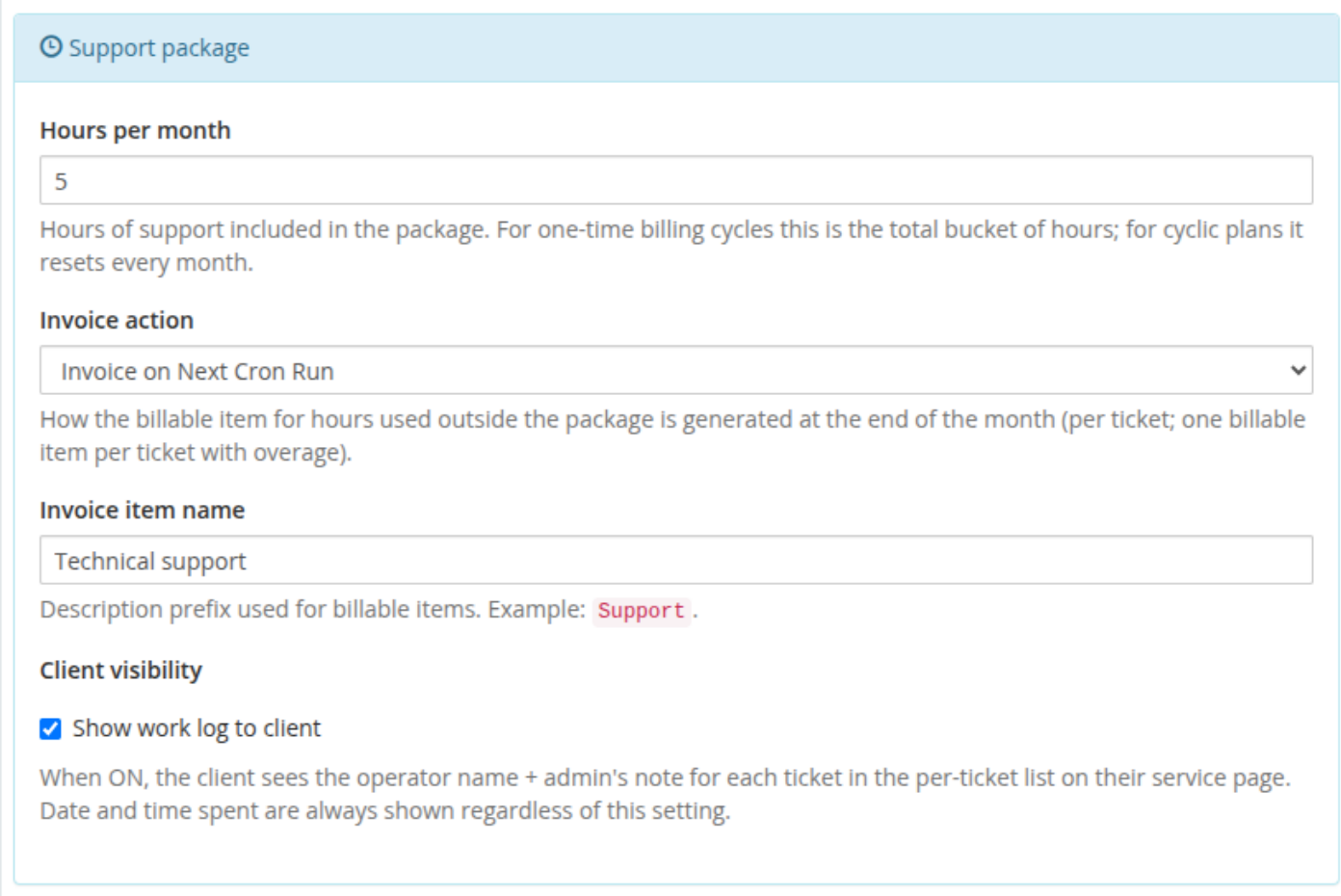
Configuration parameters

All settings live in the **Module Settings** tab. Saving the product persists them as a single JSON document in `tblproducts.configoption24` (the license key stays in `configoption1`).

Parameter	Description
License key	A pre-purchased license key for the PUQ Support by Time module. The license must be active for correct operation. After saving, the verification status (and the period it is valid until) is displayed below the field.
Hours per month	Number of support hours included in the package. For *One Time* billing cycles this is the total bucket of hours.
Invoice action	How the billable item for overage hours is generated at month-end: **Invoice on Next Cron Run**, **Add to User's Next Invoice**, or **Do not invoice**.
Invoice item name	Description prefix used for billable items (default: `Support`). The final description is composed as `<name> #<ticket> <YYYY-MM> <hours> Hour(s) (<currency> <rate>/Hour)`.
Show work log to client	When ON, the client sees the operator name and the per-entry note for each ticket in their service-page ticket list. Date and time spent are always shown regardless of this setting.
Price per hour outside the package	Hourly rate charged for time exceeding the package allocation, configured per WHMCS currency.

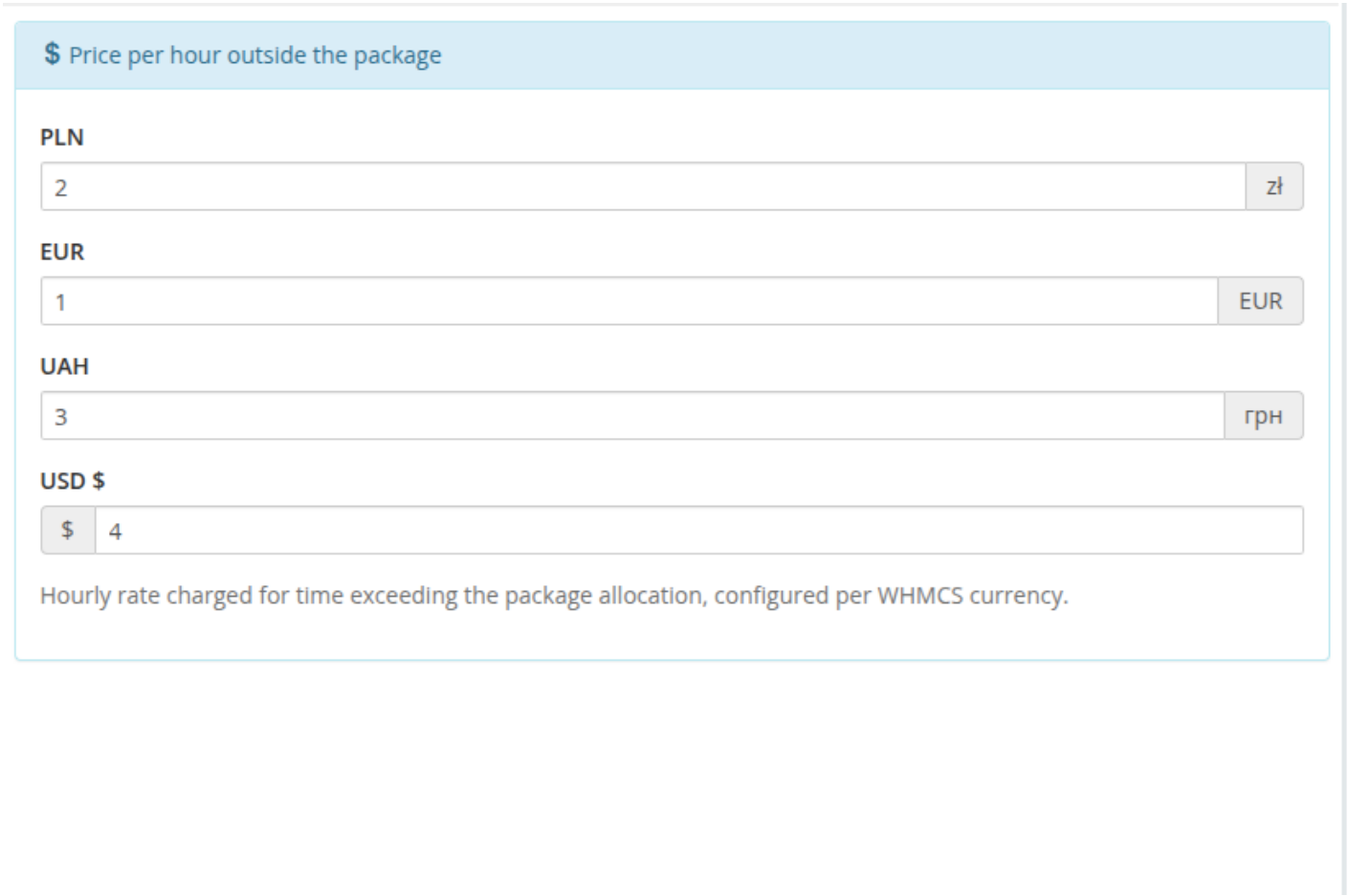
Parameter	Description
Notify thresholds (% of package)	Comma-separated percentages (e.g. `80,100`). When a client's monthly usage reaches a threshold, the email template below is sent — once per threshold per month. Empty disables notifications.
Notification email template	WHMCS Product/Service email template used for usage alerts. Available custom variables: `$puq_used_hours`, `$puq_package_hours`, `$puq_used_percentage`, `$puq_threshold`, `$puq_hours_left`, `$puq_used_hours_outside_package`, `$puq_price_per_hour`, `$puq_currency_prefix`, `$puq_currency_suffix`, `$puq_how_much_pay`.

Support package



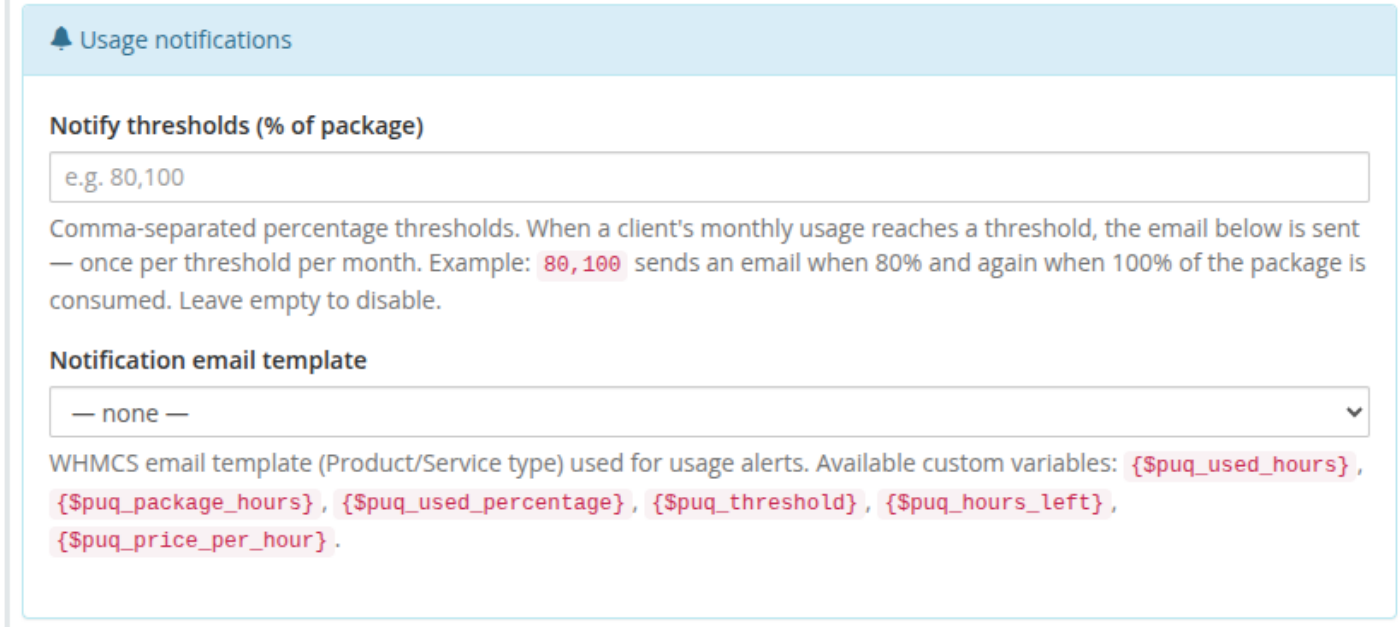
06-product-support-package.png

Price per hour (per currency)



07-product-pricing.png

Usage notifications



08-product-usage-notifications.png

How billing works: One Time vs Recurring packages

The module supports two distinct operational and billing models, selected via the product's **Payment Type** / **Billing Cycle** in WHMCS:

Characteristic	Recurring Package (Monthly, Quarterly, Annually, etc.)	One Time Package
WHMCS Payment Type	**Recurring** (with active billing cycles such as Monthly, Annually)	**One Time**
Hours Allocation	Included allowance **per month** (e.g. 5 hours each month)	Fixed single bucket of hours (e.g. 5 hours total)
Monthly Reset	**Yes.** On the 1st of each calendar month, the usage bucket resets. Unused hours do not roll over.	**No reset.** The bucket remains until depleted, regardless of how many months pass.
Hours Outside Package (Overage)	**Allowed.** If client exceeds included hours, additional time runs "outside package" and is tracked as billable overage.	**Strictly prohibited.** When remaining hours reach 0:00, no further time can be added to the service.
Overage Rate & Billing	Charged at the configured **Price per hour outside the package** in the client's currency. Generated as a WHMCS Billable Item at month-end.	**None.** No post-payment overage invoices are created. Additional support requires purchasing a new package.
Ticket Time Input Limits	**No upper cap.** Operators can enter any number of hours/minutes. An orange notice informs of overage rates.	**Capped.** The HH:MM time picker is capped at the remaining bucket hours (`used_hours_left`). Operators cannot enter more time than remains.
Ticket Button State when Depleted	**Always active.** Action buttons (*Save time*, *Close & save*, *Start*) remain enabled even if the monthly quota is exceeded.	**Blocked.** When hours reach 0, action buttons and inputs are disabled with a red warning banner: *Package hours exhausted*.
Live Timer Behavior	Can run indefinitely; extra time automatically goes to overage.	Blocked from starting if hours left is 0. If stopped with time exceeding available bucket, hours are capped at the remaining balance.
Service Lifecycle & Termination	Stays **Active** continuously as long as the recurring WHMCS subscription invoice is paid.	When 100% of hours are consumed, the daily cron runs `ModuleTerminate` and sets the service to **Terminated**.

Characteristic	Recurring Package (Monthly, Quarterly, Annually, etc.)	One Time Package
Client Area Interface	Shows monthly stats, overage cost calculator, and a multi-month **History** tab with a 12-month usage graph.	Shows single bucket progress bar, remaining hours, and ticket list. No monthly overage calculator and no History tab.
Next Service Purchase	Automatic through WHMCS renewal invoices.	Operator can use the inline **Order another service** form directly in the ticket header once hours run low or deplete.

Recurring billing cycle workflow

1. **Active tracking throughout the month:** Operators log work on tickets via **Save time**, **Close ticket and save time**, or the live timer. Each log entry is saved into `puqSupportByTime_items` and linked to the parent ticket record in `puqSupportByTime_tickets`. At this stage, all entries are unbilled (`billableid = ''`).
2. **Quota and overage calculation:** Throughout the active month, WHMCS tracks hours used against the product's `hours_per_month`:
   - If total usage is within quota, it is covered by the customer's subscription.
   - If usage exceeds quota, the difference is tracked as *Hours outside package* (overage). Both admin and client area display the overage hours and calculate the projected cost based on the configured rate per hour.
3. **Monthly settlement via WHMCS daily cron (`PreCronJob` hook):** When the WHMCS daily cron runs, the module's `PreCronJob` hook processes all unbilled tickets from the **previous calendar month** (`date = date('Y-m', strtotime('-1 month'))`):
   - **Deterministic ticket pool allocation:** All unbilled tickets for the service from that month are sorted by ticket ID in ascending order (`ksort($unbilled_tickets, SORT_NUMERIC)`).
   - **Package allowance distribution:** The service's included monthly hours (`package_hours`) are allocated to tickets starting from the lowest ticket ID:
     - If a ticket's logged time is fully covered by the remaining monthly package hours, no billable item is needed. The ticket is marked with the sentinel `billableid = '0'`.
     - When the included package pool is exhausted (`remaining_pkg = 0`), any remaining time on that ticket or subsequent tickets becomes billable overage (`overage = hours - covered`).
   - **Creation of WHMCS Billable Items:** For each ticket with overage (unless *Invoice action* is set to *Do not invoice*), the module calls the WHMCS `AddBillableItem` local API:
     - `clientid`: Customer ID
     - `description`: `<Invoice Item Name> #<ticket_id> <YYYY-MM> <overage> Hour(s) (<currency><rate>/Hour)` (e.g. `Support #259 2026-10 1.5 Hour(s) ($10.00/Hour)`)
     - `amount`: `overage_hours × price_per_hour`

    - `unit`: `'hours'`
    - `hours`: `overage`
    - `invoiceaction`: Configured product option (*Invoice on Next Cron Run*, *Add to User's Next Invoice*, or *Do not invoice*)
  - **Database stamping & audit logging:** The generated WHMCS `billableid` (or `'0'` if covered) is written to both `puqSupportByTime_tickets` and `puqSupportByTime_items`, along with `billed_hours` and `billed_at` timestamps. An audit event `ticket_billed` is logged.
4. **Post-billing lock & immutability:** Once a ticket has been stamped with a `billableid` (either an actual billable ID or `'0'`), its time entries are permanently locked. Attempting to edit logged time or use the **ReOpen** button on closed tickets is refused by the system. This prevents accidental double-billing or discrepancies with generated invoices.
5. **Invoice generation & live status linking:** Depending on the configured *Invoice action*:
  - If set to **Invoice on Next Cron Run**, WHMCS generates an invoice on its next billable items cron run.
  - If set to **Add to User's Next Invoice**, WHMCS includes the billable item in the customer's next renewal invoice.
  - The module automatically resolves generated invoices by joining `tblinvoiceitems` (`type = 'Item'`, `relid = billableid`) with `tblinvoices`. In both the admin product view and the client area History tab, tickets display direct clickable links to the invoice (`#12345`) along with real-time payment status (`Paid` / `Unpaid`).

One Time billing cycle workflow

1. **Prepaid bucket consumption:** The client purchases a fixed package (e.g., 5 or 10 hours total). Each logged ticket entry permanently deducts time from this bucket until `used_hours_left` reaches zero.
2. **Zero overage & no billable items:** One Time packages **never** produce monthly overage billable items (`SetAddBillableItem` immediately exits with 0). All support is prepaid.
3. **Strict exhaustion locking:** Once all prepaid hours have been consumed:
  - The ticket header blocks time-logging buttons and displays a notice: *All prepaid hours for this One Time service have been used. Additional time cannot be added. Please order a new support package.*
  - Starting a live timer is refused by both frontend and backend validation.
  - The ticket header presents the inline **Order another service** form so the operator or customer can immediately order a new support package.
4. **Automatic service termination (`AfterCronJob` hook):** During the daily cron run, `puqSupportByTimeHooks::afterCronJob()` checks all active One Time services. When `package_hours <= used_hours` (100% of hours consumed):
  - The module executes WHMCS `localAPI('ModuleTerminate', ['serviceid' => $service_id])`, moving the service status to **Terminated**.

- All items and parent ticket records are stamped with `billableid = '0'` so they are excluded from any candidate scans in the future.

Important notes

> **Warning:** This module is a **server module** (Products/Services). It cannot be used as an addon. Attempting to use it as an addon will return an error.

> **License gate:** Every entry point (Create / Suspend / Unsuspend / Change Package / Terminate, ticket header, configuration page) verifies the license. If `license.puqcloud.com` is unreachable and the offline cache has expired, module actions fail until connectivity is restored.

> **Configuration storage:** The entire product configuration (hours, invoice action, item name, work-log visibility, per-currency rates, notification settings) is stored as a JSON document in `tblproducts.configoption24`. Existing installations are migrated automatically on first read.

Revision #16
Created 27 September 2022 04:57:40 by Ruslan
Updated 7 October 2026 17:48:29 by Ruslan