



# **TrueCache (High Traffic Caching) and Additional Speed Optimization**

## END-USER LICENSE AGREEMENT (“EULA”) FOR CPV LAB

This End-User License Agreement (“EULA”) pertains to the CPV Lab product (called “Software” in this Agreement). Read the terms and conditions of this End-User License Agreement (“EULA”) before installing, copying, or using CPV Lab.

This End-User License Agreement (“EULA”) is a legal agreement between you, (either an individual person or a single legal entity, who will be referred to in this EULA as “you” or “Customer”), and the owner/s of CPV Lab (“Authors”). **By clicking “I accept”, installing, copying, or otherwise using any part of the above Software, you agree to be bound by the terms of this EULA.** If you do not agree to the terms of this EULA, promptly return the unused Software to your supplier for a full refund.

### 1. GRANT OF NON-EXCLUSIVE, LIMITED LICENSE

The Software is licensed, not sold. Any rights not clearly and expressly granted to you under this EULA are reserved to the Authors. Provided you have paid all applicable fees and complied with this EULA, then you are granted the following personal, perpetual (subject to termination as set forth below), non-exclusive, non-transferable, limited rights:

- a. You may use the Software on a single domain for personal purposes for each license you have acquired; or copy the Software for archival and disaster recovery purposes, provided any copy must contain all of the original Software's proprietary notices. Each user using the Software must own a license.
- b. You may not decompile, disassemble, extract or otherwise reverse engineer any of the Software. You shall not have the right to obtain or use any source code for the Software, nor copy, reproduce, or distribute the Software except as provided above. You agree not to share, rent, lease, or loan the Software. You may not make any commercial use of the Software code except as expressly permitted under this Agreement.

### 2. COPYRIGHT

The Software and documentation is copyrighted by the Authors. All Rights Reserved. Making unauthorized copies is prohibited by law. No part of the software or documentation may be reproduced, transmitted, transcribed, stored in a retrieval system or translated into any human or computer language without prior written permission.

### 3. WARRANTY, REMEDY, AND LIMITATIONS

A. YOU ARE GRANTED A LIMITED WARRANTY THAT THE SOFTWARE WILL PERFORM IN SUBSTANTIAL ACCORDANCE WITH THE ACCOMPANYING USER MANUAL FOR 56 DAYS FOLLOWING YOUR RECEIPT OF THE SOFTWARE. THIS IS YOUR ONLY WARRANTY.

B. AUTHORS' ENTIRE LIABILITY AND YOUR EXCLUSIVE REMEDY FOR ANY CLAIM YOU MAY HAVE AGAINST AUTHORS SHALL BE, AT AUTHORS' OPTION, (I) RETURN OF AMOUNTS PAID FOR THE APPLICABLE SOFTWARE; OR (II) REPAIR OR REPLACEMENT OF SOFTWARE WHICH DOES NOT MEET THE ABOVE LIMITED WARRANTY.

UNLESS OTHERWISE STATED ABOVE, THE SOFTWARE AND THE DOCUMENTATION SOLD WITH THIS PUBLICATION ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND PARTICULAR PURPOSE. THE ENTIRE RISK ARISING OUT OF THE USE OR PERFORMANCE OF THIS SOFTWARE REMAINS WITH YOU. IN NO EVENT WILL AUTHORS OR ANY OF ITS SUPPLIERS BE LIABLE FOR ANY LOST PROFITS, LOST SAVINGS, INCIDENTAL OR INDIRECT DAMAGES, OR OTHER ECONOMIC OR CONSEQUENTIAL DAMAGES, EVEN IF THEY HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. AUTHORS RESERVE THE RIGHT TO MODIFY THIS DOCUMENT AT ANY TIME WITHOUT OBLIGATION TO NOTIFY ANYONE. IN NO EVENT SHALL AUTHORS' OR ITS SUPPLIERS' LIABILITY UNDER THIS AGREEMENT EXCEED THE SUM OF ANY AMOUNTS PAID HEREUNDER BY YOU TO AUTHORS OR THE SUPPLIER.

### 4. TERMINATION

The license will terminate automatically if you fail to comply with the terms, conditions, or limitations contained in this EULA, including the payment of applicable license or other fees. On termination, you shall, at Authors' option, either return or destroy all copies of the Software and documentation. Otherwise, this EULA shall remain in force until terminated. You may terminate this EULA at any time by destroying all copies of the Software, deleting any copies of the Software from your hard drives, ceasing all use of the Software and documentation; and providing satisfactory proof to the Authors that you have done so. The disclaimer of warranty and limitations on liability contained in Section 3 shall continue in force even after your rights to use the Software are terminated.

## Table of Contents

True-Cache (Data Caching for High Traffic Users).....	4
Setting Up the CRON Job.....	5
Edit Server Setting.....	6
Configuring Caching.....	7
Common Questions about TrueCache / Caching.....	9
CRON Job for Campaigns Page Data.....	11
GEO-Stats CRON Job.....	13
Checking if CRON Jobs are Setup Correctly.....	14

# TrueCache – Data Caching for High Volume Users

## *CPV Lab 2.17 and above*

This feature addresses one of the biggest requests from Users...More Volume and Fast Stats within the Interface.

Caching of data in CPV Lab means aggregating visitor details (clicks) and keeping them in a format that will allow reports/stats to load faster and that the “clicks” table will have a reduced size. A CRON job is also setup to remove old records from the “clicks” table, to keep the DB clean and under control.

CPV Lab instances with small traffic figures should not use caching because it won't bring advantages in this case and will add an extra server load. Caching is effective for CPV Lab Users that run high volume traffic.

As a general idea...if you have less than 500,000-700,000 clicks in the “clicks” table and the CPV Lab interface and Stats pages are quick and responsive, you most likely won't see a direct benefit from caching.

**You also have 2 additional CRON functions to optimize interface speeds, in the event you're not a High Traffic User and wouldn't benefit from the Full Functions of Caching.**

Additional Setup Details are provided in this document.

### **cron-campaigns.php – Page 11**

This feature gives you the option to move some operations from the Campaigns page in an offline process that can be run periodically as a Cron job. The main advantage is that the amount of time required for the calculations isn't wasted on the campaigns.php page, it is moved in a separate process to further optimize your interface for speed.

### **GEO Stats CRON – Page 13**

This CRON controls 2 specific options that are controlled within the “constants-user.php” file, located in the “lib” folder within the root of your installation.

Note: The Cron job uses the same page (cron-cache.php) as the Caching Cron job, so if you already have Caching Enabled, you don't need to re-enter a new Cron job. All you have to do is define the “UseGeoCron” key in the constants-user.php file located in the “lib” folder within the root of your install.

## Setting up the CRON Job

The Cron job consists of a PHP page (cron-cache.php) that will be called periodically by the server and will perform the caching processes.

### #1) Login into cPanel and find the 'Cron jobs' page



Enter the Cron job details in this page. The recommended interval for the job to run is every 10-15 minutes or more.

#### **Then enter as command:**

```
cd /home/<server-name>/public_html/cpvlab/; php cron-cache.php
```

Or by an alternate command if the command above doesn't function properly on your server.  
`php /home/<server-name>/public_html/cpvlab/cron-cache.php`

**\*\*\*Replace <server-name> with the correct Server Name based on your server configuration.**

This is a sample setup for the Cron job to run at 5 minutes over the hour, every hour:

#### Add New Cron Job

Common Settings: -- Common Settings --

Minute: 5 :05 (5) ✓

Hour: \* Every hour (\*) ✓

Day: \* Every day (\*) ✓

Month: \* Every month (\*) ✓

Weekday: \* Every weekday (\*) ✓

Command: php /home/.../public\_html/cpv/cron-cache.php ✓

Click the 'Add New Cron Job' button and the new Cron job will appear in the jobs list.

MINUTE	HOURL	DAY	MONTH	WEEKDAY	COMMAND	ACTIONS
5	*	*	*	*	php /home/.../public_html/cpv/cron-cache.php	<a href="#">Edit</a> <a href="#">Delete</a>

## Edit Server Setting

Before enabling caching, increase the "group\_concat\_max\_len" system variable in MySQL to a larger value:

- at least 1048576 (1MB) for medium volumes of traffic
- for high traffic, set to could be 10485760 (10MB)

The maximum allowed value for this variable is 4294967295 (4GB), however you don't need to adjust this setting close to that.

This is a system variable that should be changed the web host since it cannot be modified from PhpMyAdmin on most servers.

## Configuring Caching

Caching can be configured based on the server settings and properties using 4 configuration keys from the “**constants-user.php**” file from the “lib” folder. If these keys don’t exist in your “constants-user.php” file, you can add them at the end of the file, before the last “}” sign.

<b>Option Name</b>	<b>Default Value</b>	<b>Description</b>
UseDataCaching	false	“true” if caching of data is used; beside this option, you also have to setup the Cron job
CachingMaxItems	10000	The maximum number of clicks that are processed in a single Cron job execution
CachingDelay	600	The delay in aggregating clicks –clicks from the last xxx seconds won’t be aggregated yet
CachingRemoveTime	2592000	The interval after which clicks are removed from the non-aggregated state

A typical caching configuration section from the “constants-user.php” looks like this:

```
// Controls if cached data are used in application ("true" or
// "false" values); you must also setup the Cache Cron job
$GLOBALS['UseDataCaching'] = "true";
// The maximum number of clicks processed in a single Cron job execution
$GLOBALS['CachingMaxItems'] = 10000;
// The delay in aggregating clicks - clicks from the last xxx seconds
// won't be aggregated yet - in seconds
//   300 - 5 minutes
//   600 - 10 minutes
//   900 - 15 minutes
//  1800 - 30 minutes
$GLOBALS['CachingDelay'] = 600;
// The interval after which clicks are removed from the "clicks" table
// (after these clicks have been aggregated) - in seconds
//  86400 - 1 day
// 432000 - 5 days
// 864000 - 10 days
//1728000 - 20 days
//2592000 - 30 days
$GLOBALS['CachingRemoveTime'] = 2592000;
```

### The “CachingMaxItems” option

Should be set to a value that is greater than the number of visitors you usually receive in the time interval between 2 consecutive Cron job executions. Setting this option to a large value will slow down the server on the first few Cron job runs if you have a large existing database with non-aggregated data (applies to upgrades because on new installations you won’t have any non-aggregated data initially).

### **The “CachingDelay” option**

Specifies a time interval when clicks won't be aggregated by the Cron job.

For example, if you run the Cron job every hour and set the “CachingDelay” to 600 seconds (10 minutes), then clicks from the last 10 minutes before the Cron job execution, won't be aggregated by this Cron job run. These clicks will get aggregated by the next Cron job execution, when again the last 10-minutes clicks will be skipped and so on.

This option is useful in order to aggregate clicks after visitors click to the offer page(s) for optimal caching of data.

### **The “CachingRemoveTime” option**

Specifies the time interval to keep non-aggregated clicks. The non-aggregated clicks will still be used by the Visitors Stats page and the Report Upload page, so if your server has the resources, it's recommended to keep 10, 20, 30, 60 days of non-aggregated clicks in the database together with the aggregated clicks.

It is important to keep some non-aggregated data in the “clicks” table for Reports and Visitor Stats. If you want to see Visitors Stats from the last 60 days, then you have to keep 2 months of non-aggregated data in the “clicks” table by configuring the “CachingRemoveTime” with a value of at least “5184000” seconds (60 days).

It's advised to start this with a higher value in order to not remove too much data initially and see how the server handles a larger number of non-aggregated clicks.

As above for 60 Days, or even higher using 7776000 seconds (90 days) as the value for "CachingRemoveTime" and this means they will be able to upload reports for the last 90 days and also see Visitors Stats for the last 90 days.

Your Server Admin can fine tune these settings, as they are server-specific. Each server has its own configuration and settings. You Server Admin will be able to adjust these settings based on your server resources/hardware and your individual needs for your volume of traffic.

They will check and monitor your Memory and CPU usage while the Caching / CRON Jobs are running in order to see how much extra load it adds and can make adjustments to run the Cron job while not affecting other server processes if the requirements are high.

**NOTE: THIS DOES NOT REMOVE ANY DATA FROM YOUR STATS OR REPORTS.**



## Common Questions about Caching

### Question - Should I use caching?

**Answer:** If you are tracking high volumes of traffic, and notice any delays when loading stats or reports in the interface...then Yes! For Users that don't run high traffic, do not need to enable caching as it won't be required. You also need to have some server knowledge in order to enable and customize caching on your server.

### Question - How often should I setup the Cache Cron job to run?

**Answer:** This depends on what you need to see in reports and if you need "real-time" data and the delay in data that you can accept. The default setting is to run the Cron job every hour, but you can lower or increase this interval based on your needs. It's very important that you don't set a very small interval that won't allow the previous Cron job execution to complete (for example if the Cron job takes 2 minutes each time to execute and you set the interval between executions to 1 minute).

For Users with Solid Hosting, 5-15 minutes will work well. However, users with lower performance servers may need a higher time interval to complete the CRON job.

### Question - Visitors don't appear in Stats; I've tested with the Campaign URL and no visitors appear in the Stats page.

**Answer:** When the Stats page uses cached data there is a delay before the Cron job runs, so visitors won't appear immediately. You can use the Visitors Stats page to check real-time traffic.

### Question - I'm keeping non-aggregated data in the "clicks" table together with aggregated data in the cache tables. Which version of data will be used in reports?

**Answer:** When caching is enabled (with the "UseDataCaching" configuration option), pages that allow caching will use the aggregated data in order to lower the page load times. The Visitors Stats and Reports Upload pages will still use non-aggregated data.

**Question - Records disappeared from the “clicks” table**

**Answer:** This is normal and it is done in order to keep the “clicks” table size under control. Old visitors’ details will be removed from the “clicks” table only after they were aggregated, so nothing is lost because reports can be obtained from the aggregated data.

You can configure the time to keep clicks in the database with the “CachingRemoveTime” configuration option.

**Question - I have visitors that convert after a long interval of time (24 hours), should I set the “CachingDelay” configuration option to not aggregate data from the last 24 hours?**

**Answer:** No, that is not necessary. Conversions will be marked correctly in the database even if the click of the visitor the converted was previously aggregated. The conversion will be added to the aggregated records in this case.

**Question - I want to upload a list of converting subIDs for old visitors that were already aggregated. Will this work?**

**Answer:** Yes, this will work and the uploaded conversions will be marked in the aggregated data. This will work even if the original visitors’ details were removed from the “clicks” table.

**Question - I want to remove a false conversion from the database for a visitor that was already aggregated. Is this possible?**

**Answer:** Yes, you will use the Stats Management page for this (Settings → Stats Management) The conversion and the associated revenue will be removed from the aggregated data.

**Question - Do I need to keep running the Cron job for the Campaigns page (cron-campaigns.php) if I run the Cron job for caching (cron-cache.php)?**

**Answer:** No, if you have caching enabled, you can Remove the Cron job for the Campaigns page. This process will be completed in the new CRON Job. In fact, the Cron job for the Campaigns page checks if caching is enabled and exits quickly in this case.

## CRON Jobs for Campaign Data

\* Optional and recommended for high traffic users who DO NOT have caching enabled.

**NOTE:** If you have TrueCache Enabled...do not run the Campaign Data CRON Job, as this process is already built in to Caching Process.

This feature gives you the option to move some operations from the Campaigns page in an offline process that can be run periodically as a Cron job. These operations calculate data on the Campaigns Page.

The main advantage is that the amount of time required for the calculations isn't wasted on the campaigns.php page, it is moved in a separate process, so users won't have long intervals when logging into the application.

### Activation

In order to activate the Cron job in CPV Lab and stop performing those calculations inside the page, you have to check the option 'Use Cron Job for data on campaigns.php' from the Settings page.

Use Cron Job for data on campaigns.php

### Installation

The CRON job consists of a PHP page (cron-campaigns.php) that will be called periodically and will perform the calculations. The job can be configured in cPanel by following the next steps:

1. Login into cPanel and find the 'Cron jobs' page



- Enter the Cron job details in this page. The recommended interval for the job to run is 'Every 15 minutes', meaning you have to enter '\*/15' for Minute and '\*' for Hour, Day, Month, Weekday (these options can also be selected from the dropdown controls on the right).

Then enter the command:

```
cd /home/<server-name>/public_html/cpvlab/; php cron-campaigns.php
```

Or by an alternate command if the command above doesn't function properly on your server.

```
php /home/<server-name>/public_html/cpvlab/ cron-campaigns.php
```

**\*\*\*Replace <server-name> with the correct Server Name based on your server configuration.**

#### Add New Cron Job

Common Settings: -- Common Settings -- ▼

Minute:  Every 15 minutes (\* /15) ▼ ✓

Hour:  Every hour (\*) ▼ ✓

Day:  Every day (\*) ▼ ✓

Month:  Every month (\*) ▼ ✓

Weekday:  Every weekday (\*) ▼ ✓

Command:  ✓

- Click the 'Add New Cron Job' button and the new Cron job will appear in the jobs list.

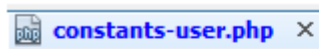
#### Current Cron Jobs

MINUTE	HOUR	DAY	MONTH	WEEKDAY	COMMAND	ACTIONS
*/15	*	*	*	*	php /home/.../public_html/cpvlab/cron-campaigns.php	<a href="#">Edit</a> <a href="#">Delete</a>

## GEO-Stats CRON Job

This feature allows you to move the process of capturing GEO data to a Cron job, to move the process outside of the Campaign URL and won't slow the redirects, recommended for high traffic figures and large databases. In order to enable the Cron job for Geo Details you have to follow these steps:

1. Set the key named "UseGeoCron" from the constants-user.php file to "true". If this key doesn't exist in the constants-user.php file, then you should add the key like in the image below:



```
//Controls if the GEO details are pulled by the Cron job;
//otherwise they are captured by the base.php page
$GLOBALS['UseGeoCron'] = "true";
```

2. Setup the Cron job on the server. Based on the server settings the command should be one of the following:

```
cd /home/<server-name>/public_html/cpvlab/; php cron-cache.php
php /home/<server-name>/public_html/cpvlab/cron-cache.php
```

**\*\*\*Replace <server-name> with the correct Server Name based on your server configuration.**

3. The Cron job should be setup to run every 10-15 minutes. This interval is the delay that the Geo Details will appear in CPV Lab for visitors.


### Add New Cron Job

Common Settings:	-- Common Settings --	
Minute:	*/15	Every 15 minutes (*/15) ✓
Hour:	*	Every hour (*) ✓
Day:	*	Every day (*) ✓
Month:	*	Every month (*) ✓
Weekday:	*	Every weekday (*) ✓
Command:	cd /home/<server-name>/public_html/cpvlab/; php cron-cache.php ✓	
<input type="button" value="Add New Cron Job"/>		

**Note:** The Cron job uses the same page (cron-cache.php) as the Cache Cron job, so if you already have the Cache Cron job setup you don't need to re-enter a new Cron job. All you have to do is define the "UseGeoCron" key in the constants-user.php file.

## Checking if CRON Jobs are Setup Correctly

Each time the Cron job runs, it will write 2 records in the Error Log, like in the image below. One for marking the start of the process and One for the end of the process.

 **Error Log**

Interval: from  (mm/dd/yyyy) to  (mm/dd/yyyy)   ▾

Error Type:  All  Website  Install  Upgrade  Cron Jobs

Remove errors before:

<a href="#">Date/Time</a> ↑	<a href="#">Type</a>	<a href="#">Page</a>	<a href="#">Message</a>	<a href="#">Query</a>
12/11/2013 6:42 AM	Cron job		CRON WURFL job finished for ID=7bc42325-bf8f-ccc9-52cc-00001e4b3935	
12/11/2013 6:40 AM	Cron job		CRON WURFL job started for ID=7bc42325-bf8f-ccc9-52cc-00001e4b3935	

# Knowledgebase & Helpdesk

@

[www.TheLabHelpdesk.com](http://www.TheLabHelpdesk.com)