

# PHP DESK TOP APPLICATION MAKER AND COMPILER HELP

Step by step guide for create your First PHP desk Top application include installer exe

(1) [Make your first application](#)

(2) [Compile your application](#)

(3) [Generate License for your client](#)

(4) [Settings.json Details \(For advance User\)](#)

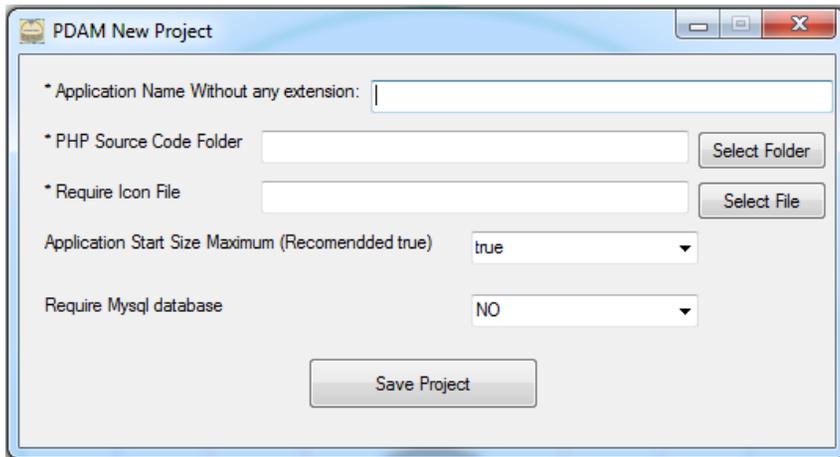
## Make your first application

(1) First open your windows paint and create a suitable image for your application Icon . Image size will be suitable 400X400 or 500X500 Or 600X600 >>(2)Run the PDA\_MAKER.exe , after open it will show like below image



>>(3) connect your internet >>(4) Click on “**MAKE ICON**” button, It will reached you to our online ICON Maker page, here you will select 16X16 + 32X32 + 48X48 , click **browse** button and select your image then click on **upload** button, it will redirect you for download your ICON file, After download you’re your Icon you will be ready to create your First PHP DESK top application.

Now Click on “**NEW\_PROJECT**” button on PDA\_MAKER, It will open a new project windows like below image, Here you will give the [name of your application](#) name, latter your exe and your installation folder create with this name so chose a better name and if possible, try to get the name without any space. >> now you will select your [PHP source Code Folder](#), and select your recently generated [icon file](#). If you are using My sql database into your project then **Require mysql database** field will be YES either leave it to No.



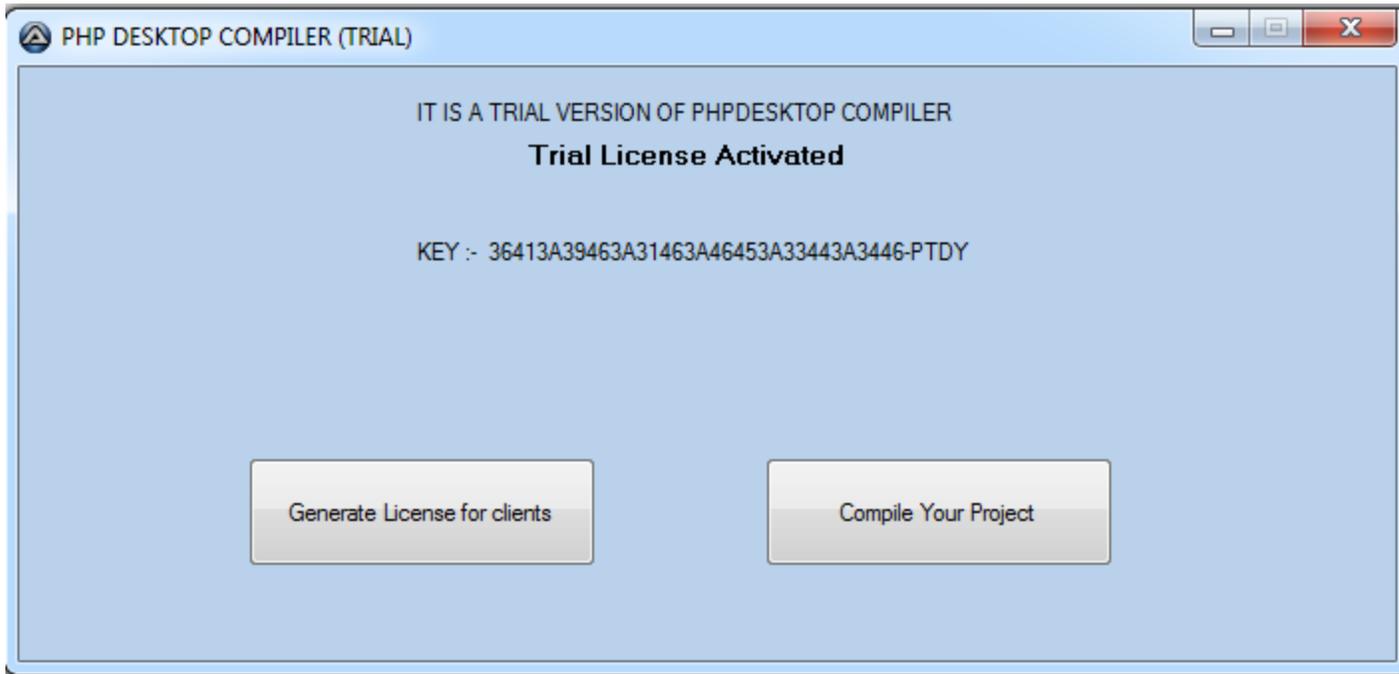
After fill all the fields, click on “**Save Project**” Button.

If you are using MY SQL database into your project Then Click on “PHP\_MYADMIN” button on PDA\_MAKER and create your database and import your SQL file to the database through PHP-Myadmin, either you can click on” OPEN\_PROJECT” button and select your newly created project file that will open your application on PHP DESKTOP APPLICATION. If you use MY sql then first follow above step to create your MYSQL database >> now click on “EXIT” button on PDA\_MAKER >> now again run PDA\_MAKER >> click on “**OPEN\_PROJECT**” button and select your recently created projet that will be present in “**Save\_Project**” folder into your PHP DESKTOP APPLICATION MAKER installation folder, most probably it will be “**C:\NEW\_PHPDESKTOP\ Save\_Project**”

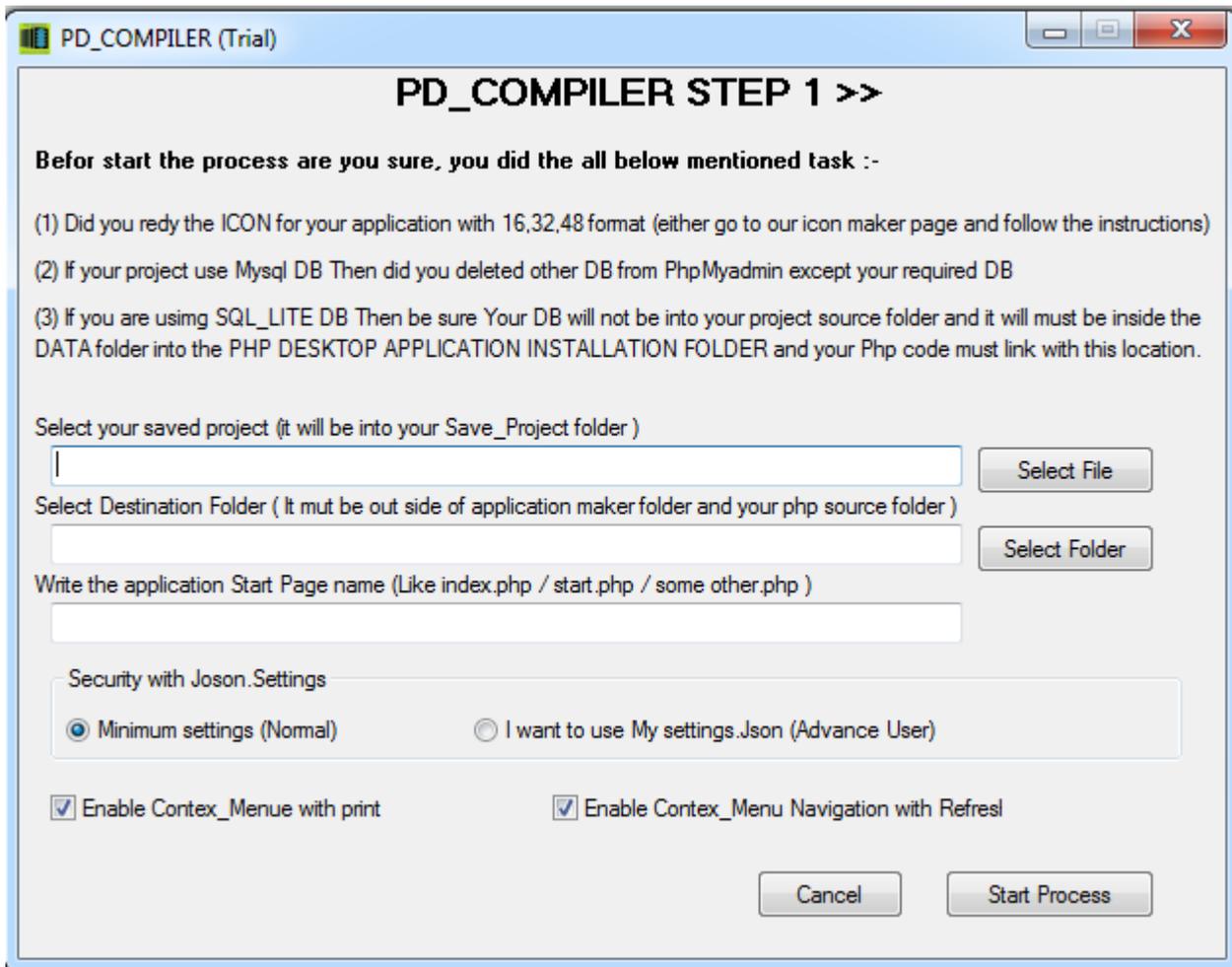
**Note:-** If you are using SQL-LITE database to your PHP project then copy your sql\_lite database file to C:\NEW\_PHPDESKTOP\DATA folder and copy your entire PHP project into www folder of application maker (not any subfolder) your index or start php will must be www\index.php, now configure your database connection into php file with this new location, then create new project as per above guide line except your source php folder, select your www folder from new project windows as source code folder.

## Compile your application

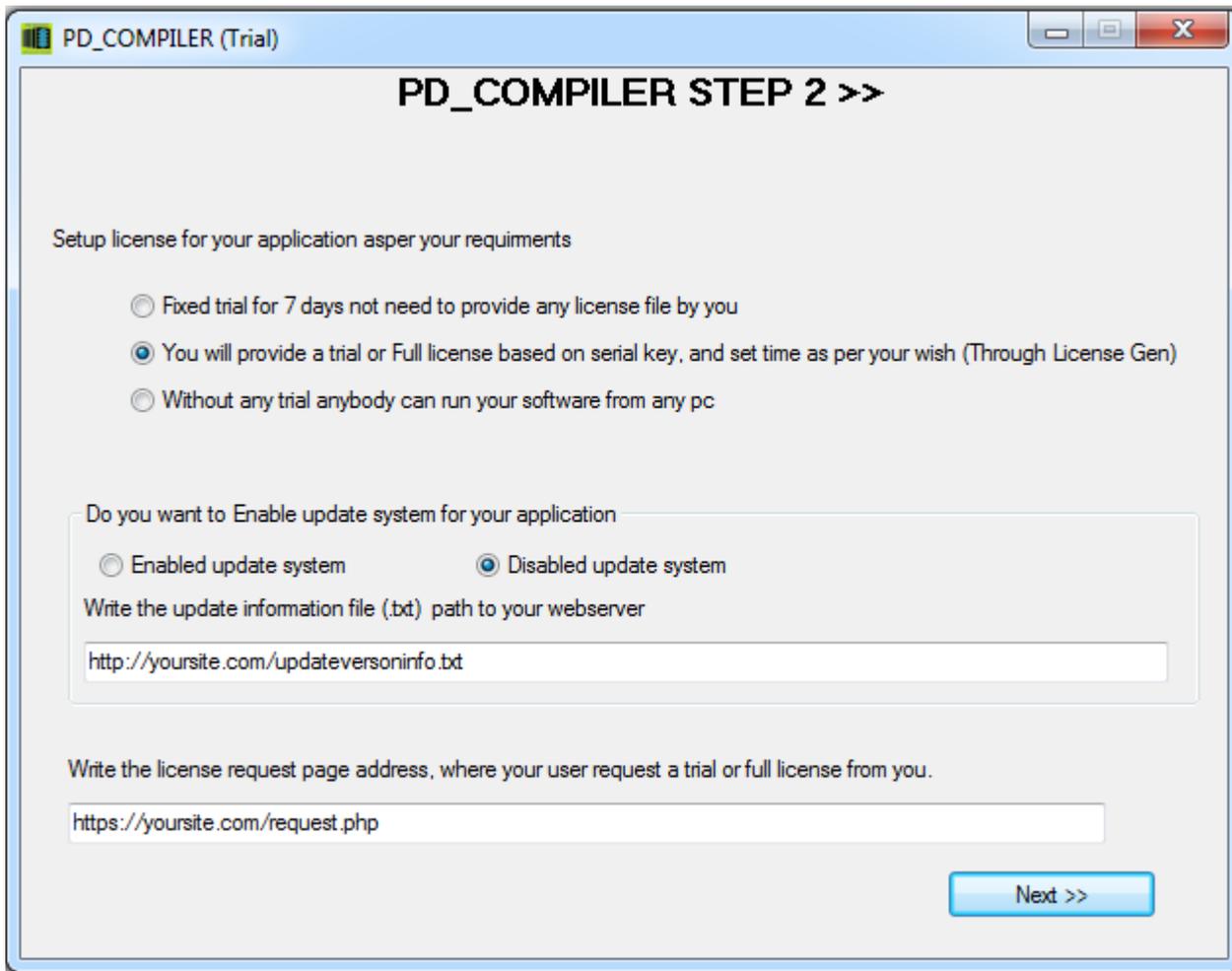
We have been created our First PHP+MYSQL Desk top application, now we will compile it for distribution to client, so click on **COMPILE** button, if you have not installed compiler add-ons then it will show you “Compiler was not installed” message and link of this add-ons for download or buy, from here you can download a 15 day’s trial compiler for your own testing. Compiled application from trial compiler will run up to 48 hours but this time is enough for your testing purpose. After install the compiler when you will click again on “COMPILE” button it will close your PDA\_MAKER and show you below screen.



Click on “**Compile Your Project**” button it will start compile wizard like below screen, other button will go to License Generation of your compiled Project we will discuss it latter.



If your project is MYSQL based project then before start your compile wizard, open PHP-MYADMIN from desktop application maker and remove all other unnecessary database if you have any **except ( your own database for this project, information schema, mysql, performance schema and phpmyadmin database)**. After appear the above screen select your newly created project file from “C:\NEW\_PHPDESKTOP\ Save\_Project” folder, create anew folde on different location or other drive and select it as Destination Folder, now type the name of your first page like, Index.php or Index.html or if you have any other file with extension and other field leave as it is for now for this time, whe you understand and learn about Settings.json then you can select advance user option. Now press “**Start Process**” button and go to the next step. Please see the below picture.



Here you will setup licensing system for your client, if you select fixed 7 days trial or without any then not require any license file to run your compiled application, but if you select middle by default option then you need to generate license for your every clients depend on their key, this key will generate automatically by your compiled application.

Now select update checker system for your client, if you want activate this system then select the “Enabled update system” radio button also provide a link of a text file that will be present into your web server.

**How it works** suppose your current version is 2.5.1, and after some modification compiled again your project to 2.9.5 so open your text file from your web server and write 2.9.5 at the very beginning of your text file not any other text or space before or after this numeric value, now if your client is connected to

internet and open your old versioned application then update checker compare the version and give a on screen message to user the version updated and provide your website link to come to your site for download or buy.

Last field here you will get your website license request page link, in this page you will get uniq key from client by using PHP Get Method and other details collect from your html form, but **if you select fixed Trial or without any then not require fill this field leave it as it is.** Now click on **“Next>>”** button, you will get a pop up screen **“Compiler is arranging”** it will take 50 to 90 second depend on your PC speed. After disappear this popup you will get below screen, Here you will set up you **about screen. Your client get your about screen by click on About button of your application or it will be automatically appear when your application will be end within 7 day’s or expired the license.**

PD\_COMPILER (Trial)

### PD\_COMPILER STEP 3 >>

Here you will setup your ABOUT screen That will show everything about your software

About Screen Background color (as per your application)

About Screen Text color

About Screen Logo image (64X64) either leave it blank

Application Name

Your Company name

Your website address

Product Version

Customer support Company name

Support email

Support Contact No

Company Address

**NOTE :-**

- \* If you don't want to show any particular field on your about screen, just leave it
- \*\* Before click NEXT Just Click on save Button , next, Click on 'View your About screen' button
- \*\* After satisfy with your screen , click on NEXT button, either Change the field value and view again
- \*\* remember This screen will automatically appear before 7 days from license expiry time

Here fill everything except application Name click on **“save”** button the click on **“View about Screen”** button , if you want to rest color or any text then change the value and again save then view, after a satisfied result click on **“Next”** button . After click the next button you will get another popup it will last 5 second to 3 minute depend on your project type, if you are compiling MYSQL based project then it will take 3 to 5 minute either 5 to 20 second,. You will get exe property screen after disappear the popup windows. **What is exe property ? when we take our mouse pointer on any exe and hold the position 4 to5 second then it will show some information about that exe.** See the below image for Property settings window.

PD\_COMPILER (Trial)

### PD\_COMPILER STEP 4 >>

File version:

Company name:

Comment:

File description:

Product version:

Product name:

Legal copyright:  ©

Legal trademarks:

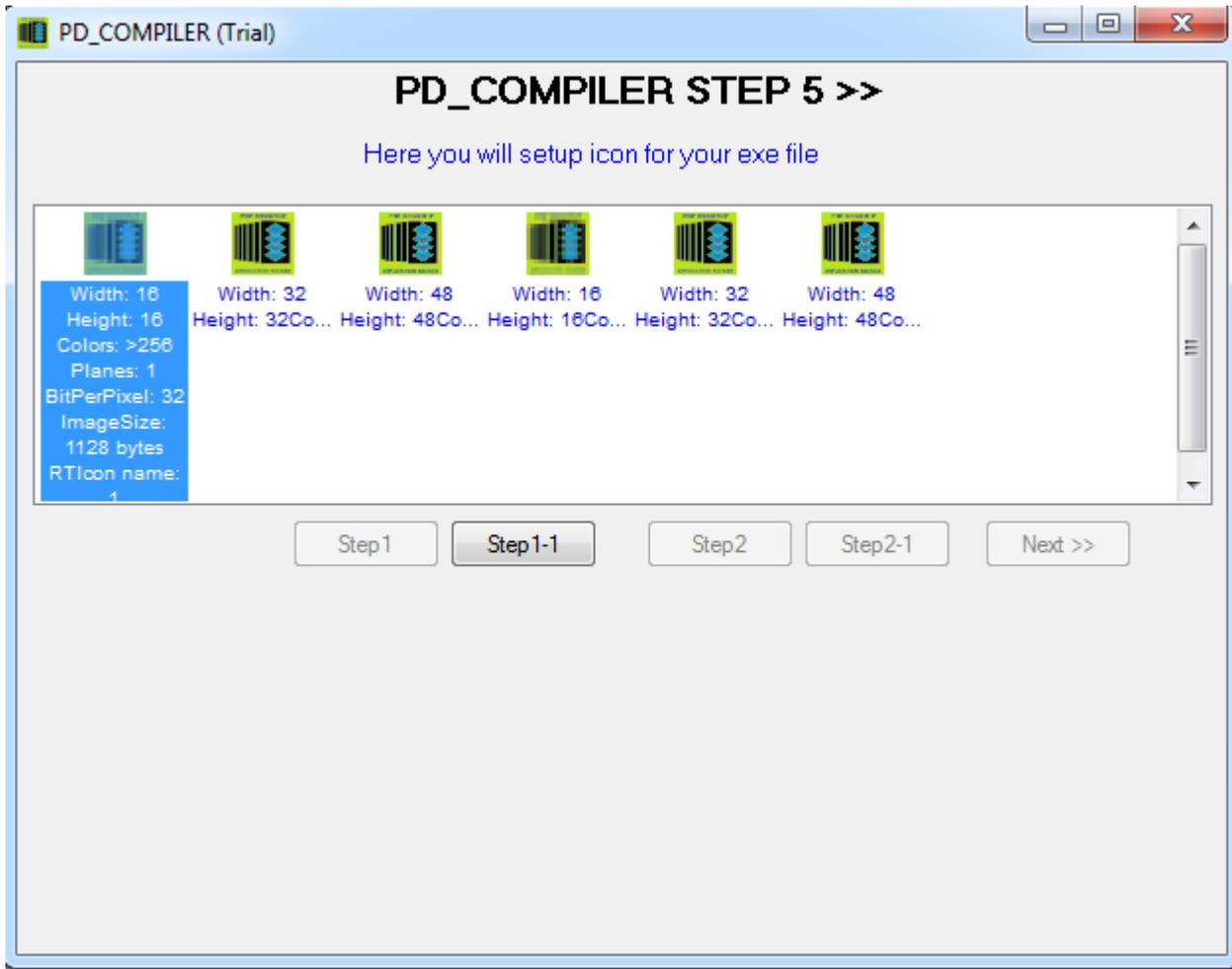
Internal name:

Original file name:

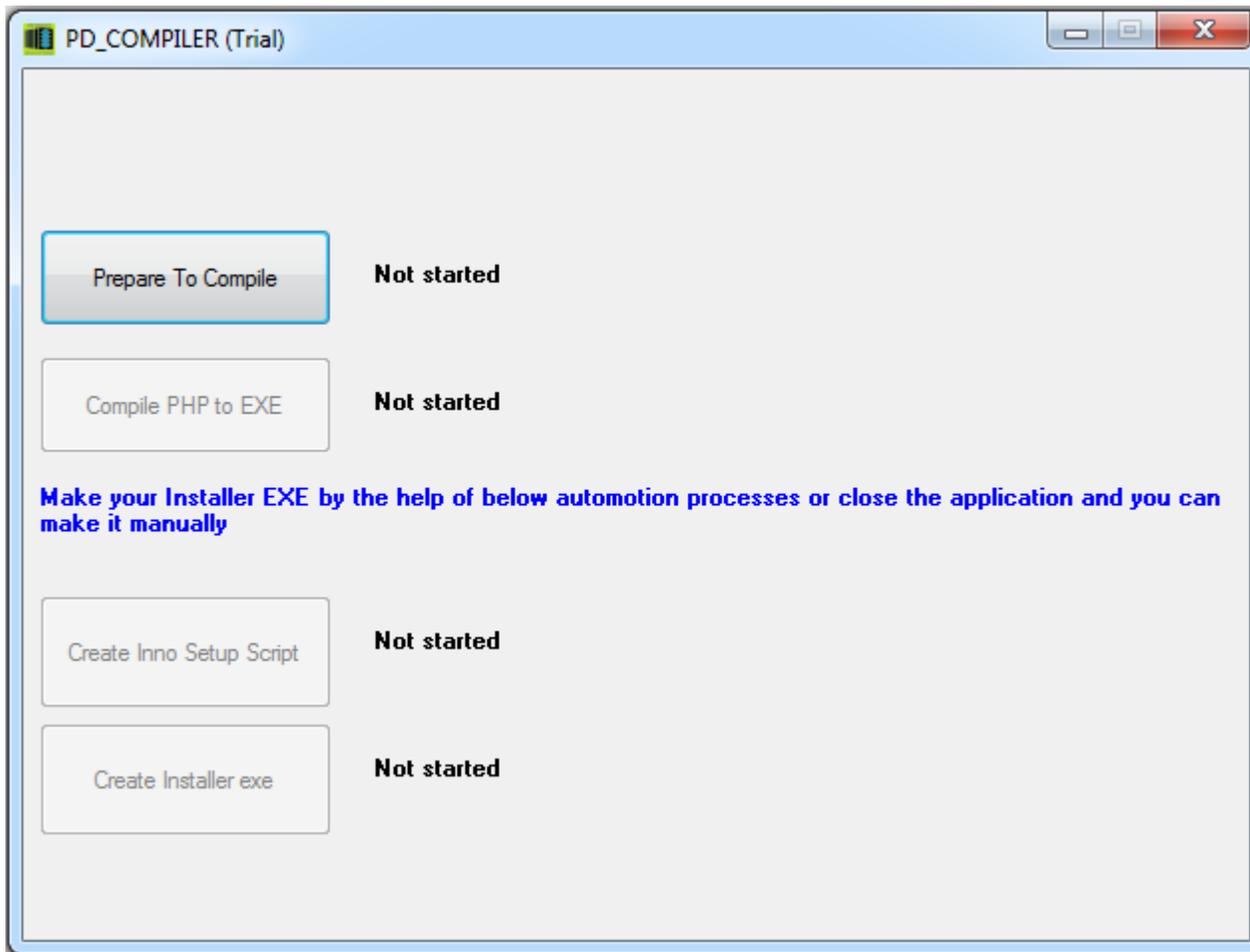
Any Special build:

Any private build:

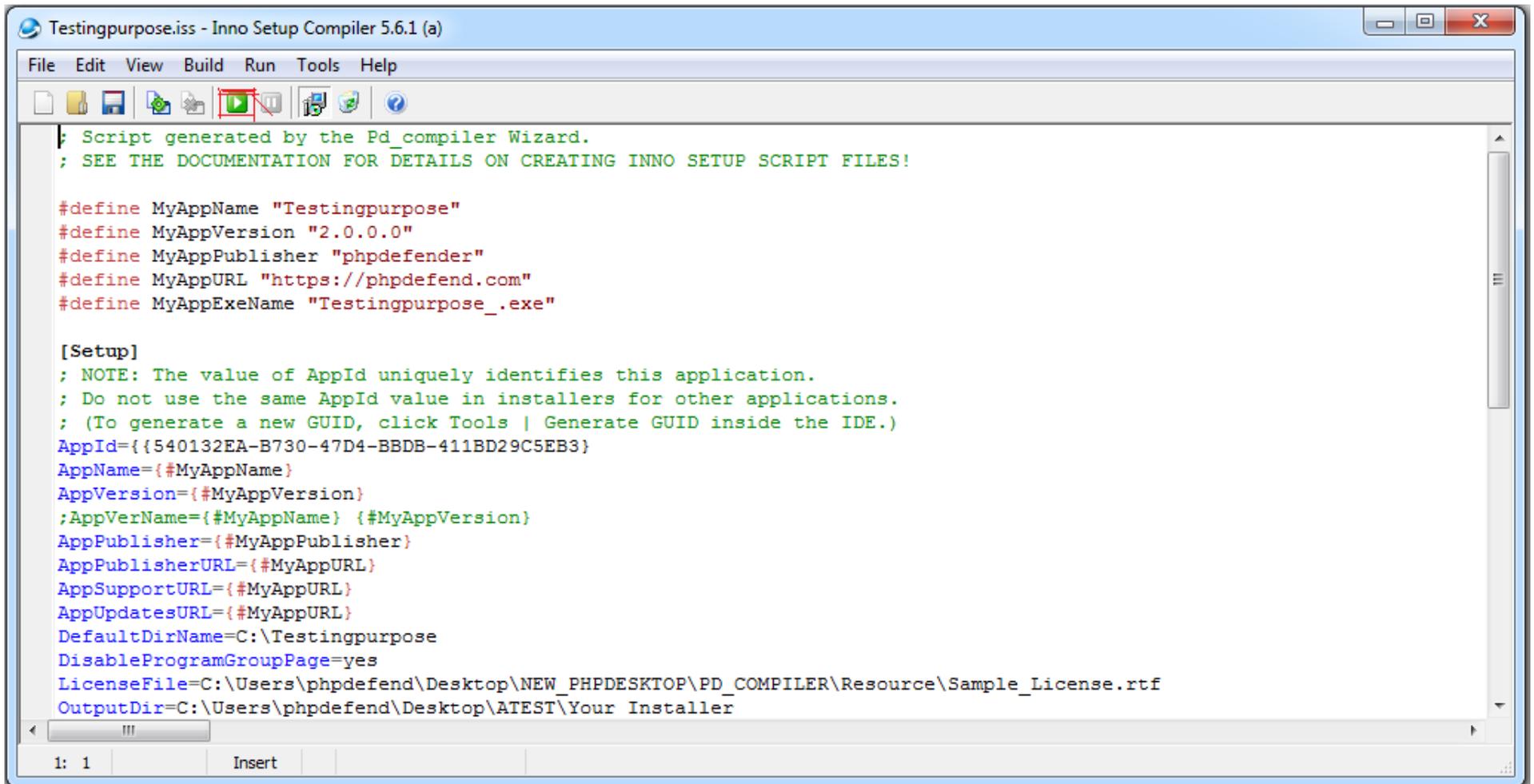
Remember , version that you setup into previous screen ,same version put into the version fields of this screen for avoid the version mismatch, other field edit as per your requirements. Now click on **“SAVE &NEXT”** button it will bring the below screen that will change your EXE icon that you setup on time of New project Creation but be sure your icon file is **not** removed from that location either it will create error and you’re your process will be interrupted. This window will guide you step by step. First select **“Step1”** button it will show the system generated icon, select any one icon then click on **“Step1-1”** button and carry on this processes up to **“Next”** button.



After click the next option it will go to the Wizard Last screen, Although last screen will guide you step by step but here we also discuss about it. See the below for last screen.



Here First Click **“Prepare To compile”** Button after finish the process **“Compile PHP to EXE”** button will activate after finish the compile it will activate **“Create Inno Setup script”** button but don’t click on it, first open the **Sample\_License.rtf** (C:\NEW\_PHPDESKTOP\PD\_COMPILER\Resource\Sample\_License.rtf) file on Microsoft Word and replace it red text to your own company name and product name , save it. After save the document click on **“Create Inno Setup script”** button it will create your script and activate the last button. Now click on **“Create Installer exe”** button, it will cloe the PD compiler Wizard and run the Inno set up script like below



```
; Script generated by the Pd_compiler Wizard.
; SEE THE DOCUMENTATION FOR DETAILS ON CREATING INNO SETUP SCRIPT FILES!

#define MyAppName "Testingpurpose"
#define MyAppVersion "2.0.0.0"
#define MyAppPublisher "phpdefender"
#define MyAppURL "https://phpdefend.com"
#define MyAppExeName "Testingpurpose_.exe"

[Setup]
; NOTE: The value of AppId uniquely identifies this application.
; Do not use the same AppId value in installers for other applications.
; (To generate a new GUID, click Tools | Generate GUID inside the IDE.)
AppId={{540132EA-B730-47D4-BBDB-411BD29C5EB3}}
AppName={#MyAppName}
AppVersion={#MyAppVersion}
;AppVerName={#MyAppName} {#MyAppVersion}
AppPublisher={#MyAppPublisher}
AppPublisherURL={#MyAppURL}
AppSupportURL={#MyAppURL}
AppUpdatesURL={#MyAppURL}
DefaultDirName=C:\Testingpurpose
DisableProgramGroupPage=yes
LicenseFile=C:\Users\phpdefend\Desktop\NEW_PHPDESKTOP\PD_COMPILER\Resource\Sample_License.rtf
OutputDir=C:\Users\phpdefend\Desktop\ATEST\Your Installer
```

Here you will see a red marked run button just click on it up to finish the process. After finish it will ask you for install the current compiled application select NO.

If everything was Ok then go to your Destination folder that you had selected on first screen of wizard, her you will see a folder **"Your\_Installer"** Inside this folder you will get your final Installer exe for your clients. Also here you see another folder with the name of your application here you will see after install your application your client what will get, after check and took attest run you can remove this folder, also you will get three files **"compile\_details.txt, Config "** and your inno setup script file. Yor first to files **"compile\_details.txt, Config "** is very important for you, you must save this file to a proper location. When you generate any license for your client the it will demand compiler id you will get this id from the **compile\_details.txt** file, without this id your license will not work on your client machine.

## Generate License for your client

See the below image for license generation screen.

**Your Client Name**  
**Client Mail ID**  
**UNIQUE KET that you will get from your websit by application**  
**Compile ID That you will get from compile\_details.txt**

## SETTINGS.JSON

- [application](#)
  - [single instance guid](#)
  - [dpi aware](#)

- [debugging](#)
  - [show\\_console](#)
  - [subprocess\\_show\\_console](#)
  - [log\\_level](#)
  - [log\\_file](#)
- [main\\_window](#)
  - [title](#)
  - [icon](#)
  - [default\\_size](#)
  - [minimum\\_size](#)
  - [maximum\\_size](#)
  - [disable\\_maximize\\_button](#)
  - [center\\_on\\_screen](#)
  - [start\\_maximized](#)
  - [start\\_fullscreen](#)
  - [always\\_on\\_top](#)
  - [minimize\\_to\\_tray](#)
  - [minimize\\_to\\_tray\\_message](#)
  - [save\\_position](#)
- [popup\\_window](#)
  - [icon](#)
  - [fixed\\_title](#)
  - [center\\_relative\\_to\\_parent](#)
  - [default\\_size](#)
- [web\\_server](#)
  - [listen\\_on](#)
  - [www\\_directory](#)
  - [index\\_files](#)
  - [cgi\\_interpreter](#)
  - [cgi\\_extensions](#)
  - [cgi\\_temp\\_dir](#)

- [404 handler](#)
- [hide files](#)
- [Chrome settings](#)
  - [cache\\_path](#)
  - [command\\_line\\_switches](#)
  - [context\\_menu](#)
  - [devtools\\_F12](#)
  - [external\\_drag](#)
  - [external\\_navigation](#)
  - [log\\_file](#)
  - [log\\_severity](#)
  - [reload\\_page\\_F5](#)
  - [remote\\_debugging\\_port](#)

## Application

### **single\_instance\_guid**

Default value: ""

Set this to non-empty string to disallow multiple instances of your application. This string should be unique. It is recommended to use a [GUID generator](#) for that. An example GUID would be: "4C00DEC0-9298-4dc4-98BD-0FAE9F3BA730".

### **dpi\_aware**

Default value: true

Since PHPDesktop Chrome 39 behavior of DPI and zooming has changed. DPI awareness is always enabled in phpdesktop (DeclareDPIAware.manifest is embedded in exe). Setting this option to false will only affect window size during creation.

Whether application should be DPI aware. If you change text size settings on Windows (in Win7 go to Control Panel -> Appearance and Personalization -> Display), then app will detect this and enlarge texts in the Chrome browser. Chrome zooming technology will be used. System's default text enlarging mechanism that would make text look fuzzy won't be used.

When dpi aware is set to true, besides enlarging text size, the window size (specified by the "default\_size" setting under the "main\_window" section) will also be enlarged respectively. So if the Win7 text size settings are set to Large (150%) and you set the default\_size to [1024, 768], then the window will be resized to [1536, 1152].

---

## **debugging**

### **show\_console**

Default value: false

Whether to show command-line console with debug output. This is almost the same output that is written to the log file with the exception that messages from the subprocesses (Renderer, GPU) are not visible in the main console window.

### **subprocess\_show\_console**

Default value: false

Show more consoles for all Chrome subprocesses.

### **log\_level**

Default value: "INFO"

Application level logging (do not mistake with php logging), possible values are: ERROR, WARNING, INFO, DEBUG, DEBUG1, DEBUG2, DEBUG3, DEBUG4. Level of ERROR will only log severe errors, DEBUG4 is most verbose.

### **log\_file**

Default value: ""

Log file for application logging. Do not mistake with php logging. To set php log file set the "error\_log" option in php.ini:  
<http://www.php.net/manual/en/errorfunc.configuration.php#ini.error-log>

Debug output is written to this file, set it to empty string to disable logging to file.

Restriction: Only one instance of an application can log to that file at the same time, when second instance of an application is started, it won't write messages to the log file.

---

## **main\_window**

### **title**

Default value: "PDAM"

Main window title. If set to empty executable's name will be used instead.

### **icon**

Default value: " "

### **default\_size**

Default value: [1024, 768]

Default size of the window, width & height.

### **minimum\_size**

Default value: [800, 600]

Minimum size of the window, width & height. User won't be able to resize the window to a size smaller than this. A value of 0 means there is not minimum size limit when resizing.

### **maximum\_size**

Default value: [0, 0]

Maximum size of the window, width & height. User won't be able to resize the window to a size bigger than this. A value of 0 means there is no maximum size limit when resizing.

### **disable\_maximize\_button**

Default value: false

Whether to disable maximize button. You can mix this option along with `minimum_size` & `maximum_size`, thus allowing you to define a window with a fixed size that cannot be resized in any way.

### **center\_on\_screen**

Default value: true

Whether to center window on screen. If you set this to false, OS will take care of positioning the window.

### **start\_maximized**

Default value: false

Starts the main application window maximized.

### **start\_fullscreen**

Default value: false

Starts the main application window in fullscreen mode.

### **always\_on\_top**

Default value: false

Places the window above all non-topmost windows. The window maintains its topmost position even when it is deactivated.

### **minimize\_to\_tray**

Default value: false

Whether to minimize to system tray - taskbar icon will be hidden.

### **minimize\_to\_tray\_message**

Default value: "Minimized to tray"

Message that will be displayed in a Balloon Notification after being minimized.

### **save\_position**

Default value: false

Not yet implemented. Whether to save window size & position and restore it on next launch of the application..

---

## **popup\_window**

### **icon**

Default value: ""

You can set a different icon for popup windows. If set to empty, main window icon will be used.

### **fixed\_title**

Default value: ""

The title for the popup window is set according to meta `<title>` tag on a webpage, however you can set a fixed title for all popup windows, in this case meta title will be ignored.

### **center\_relative\_to\_parent**

Default value: true

Center popup window relative to the parent window.

## **default\_size**

Default value: [1024, 768]

The default size used when there is no size specified for the popup window (no width/height parameters passed to window.open).

---

## **web\_server**

### **listen\_on**

Default value: ["127.0.0.1", 0]

Ip address and port to listen on. It is recommended for the port to be in range 49152–65535. When set to 0, OS will assign a random free port. It is recommended to set it to 0 to avoid port conflicts.

Note that if you're using HTML5 persistent storage such as local storage, indexed db or web sql, that lives on between application launches, then you need to set a fixed port, eg. ["127.0.0.1", 54007]. for more details.

To listen on all available IP addresses (local and internet public ip addresses) set the IP address to "\*" and also set a fixed port (if you need a fixed web server url that doesn't change with each app relaunch). For example in settings.json set this: "listen\_on": ["\*", 54007]. Mongoose web server will listen on all network cards, but it is still up to you to detect computer's public IP address. URL address launched in phpdesktop browser is still local "http://127.0.0.1:xx/". Note that you may still need to configure router port forwarding for it to work for an internet IP address. Note also that when "listen\_on" is set to "\*" then the SERVER\_NAME environment variable will still be set to 127.0.0.1 - this may cause issues in PHP scripts that depend on that env variable. To fix it add this code at the top of all your PHP scripts: `<?php $_SERVER["SERVER_NAME"] = $_SERVER["HTTP_HOST"]; ?>`. Same for the \$\_ENV global variable. This will set SERVER\_NAME to an ip address that the web server is being accessed from. For example when accessing from 127.0.0.1 it will be set to that. When accessing from 192.168.0.2 then it will be 192.xxx in that case. Same for an internet IP address.

### **www\_directory**

Default value: "www"

Document root directory for your PHP scripts.

## **index\_files**

Default value: ["index.html", "index.php"]

List of files to be treated as directory index. A maximum of 32 files is allowed.

## **cgi\_interpreter**

Default value: "php/php-cgi.exe"

CGI interpreter for all CGI scripts regardless script extension.

From Mongoose documentation: "Mongoose decides which interpreter to use by looking at the first line of a CGI script" - this is probably not true after we've set cgi extensions.

## **cgi\_extensions**

Default value: ["php"]

Files with these extension are treated as CGI. A maximum of 32 extensions is allowed.

## **cgi\_temp\_dir**

Default value: ""

The temp directory for the CGI environment. This will set three env variables: TMP, TEMP, TMPDIR. It is recommended to leave this string empty, read below.

If set to empty string PHP Desktop will set a temp directory for you automatically. If the temp directory returned by OS contains unicode characters (PHP doesn't play well with unicode), then it will choose another temp directory for you that will not cause problems with PHP.

## **404\_handler**

Default value: "/pretty-urls.php"

Http 404 not found errors will call the provided php script. You can use this to enable support for pretty urls in application. This is an equivalent of using Apache's mod\_rewrite using this set of rules:

```
<IfModule mod_rewrite.c>
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule ^(.*)$ pretty-urls.php/$1 [L]
</IfModule>
```

To know what pretty url was accessed check `$_SERVER["PATH_INFO"]` or `$_SERVER["REQUEST_URI"]`. See [pretty-urls.php](#) script for an example.

## hide\_files

Default value: []

Patterns for the files to hide. Files that match the pattern will not show up in directory listing and return 404 Not Found if requested. Pattern must be for a file name only, not including directory name. Maximum number of patterns is 100.

Special characters that can be used in patterns:

- \* (matches multiple characters)
- ? (matches one character)

Example values: [".\*", "secret.txt"] - hides all files starting with dot and additionally "secret.txt".

## Chrome settings

### cache\_path

Default value: "webcache"

The directory where cache data will be stored on disk. If set to empty string, an in-memory cache will be used and your cookies and other data (like HTML5 databases) will not persist across app launch sessions. Chrome flushes cache data to disk every 30 seconds or so.

### command\_line\_switches

Default value: {}

There are many settings that can be customized through Chrome command line switches. These switches can only be set programmatically in the settings.json file, by passing a dictionary of switches, with switch name as a key. The switch name should not contain the "-" or "--" prefix, otherwise

it will be ignored. Switch value may be an empty string, if the switch doesn't require a value. These switches are set only for the main browser process.

See the peter.sh website for an assembled list of all Chrome switches: <http://peter.sh/experiments/chromium-command-line-switches/>

See also CEF switches in the [cef\\_switches.cc](#) file.

Example code in settings.json:

```
command_line_switches: {  
  "lang": "fr",  
  "enable-media-stream": "",  
  "enable-system-flash": ""  
},
```

Example switches:

- {"lang": "fr"} - The language file that we want to try to open. Of the form language[-country] where language is the 2 letter code from ISO-639. See locales/ directory for a full list of supported languages.
- {"enable-media-stream": ""} - Enables media streaming (WebRTC audio/video).
- {"enable-system-flash": ""} - Enables Flash. Additionally you need to [install PPAPI system-wide plugin](#).
- {"enable-speech-input": ""} - Enable speech input (x-webkit-speech).
- {"disable-web-security": ""} - Don't enforce the [same-origin policy](#).
- {"proxy-server": "socks5://127.0.0.1:8888"} - To set custom proxy. See also [Proxy Resolution](#) in upstream CEF.
- {"no-proxy-server": ""} - By default Chromium uses the IE proxy settings (set in Internet Explorer options). To disable use of that proxy set the "no-proxy-server" Chromium switch. See also [Proxy Resolution](#) in upstream CEF.

These switches can also be set from command line when launching the phpdesktop executable, for example "phpdesktop-chrome.exe --enable-media-stream". This can be useful to quickly test some of the switches without being forced to edit the settings.json file. Also when combined with the ability to [pass CGI environment variables as command line arguments](#) you are capable of creating multiple phpdesktop browser windows with a mix of different settings to fulfill your application specific needs.

## **context\_menu**

Default value: {}

The mouse context menu is configurable using the following options:

- enable\_menu - whether to enable context menu.

- navigation - whether to show the "Back", "Forward" and "Reload" options.
- print - whether to show the "Print..." option.
- view\_source - whether to show the "View source" option.
- open\_in\_external\_browser - whether to show the "Open page in external browser" and "Open frame in external browser" options.
- devtools - whether to show the "Show Developer Tools" option. See also the "devtools\_F12" option. You have to set the "remote\_debugging\_port" option for the developer tools to be available.

## **devtools\_F12**

Default value: true

Developer tools will open when pressing the F12 key. See also the ["context\_menu"]["devtools"] option. You have to set the "remote\_debugging\_port" option for the developer tools to be available.

## **external\_drag**

Default value: true

Whether to allow external drag events to enter the browser window. For example an url is dragged and navigation occurs, or a file is dragged and its contents are displayed.

There are methods for detecting what type of element is dragged into the window, in future we might allow filtering and allowing some type of the drag events,.

BUG NOTICE: This also controls internal drag and drop events on websites,.

## **external\_navigation**

Default value: true

Whether to allow navigation to external websites inside PHP Desktop browser. When set to false external links will be opened in OS default browser. This behavior applies in general to external navigation, so this gets fired not only when clicking a link, but also when doing a redirect (whether it is initiated from php or from javascript using the window.location object).

## **log\_file**

Default value: "debug.log"

The directory and file name to use for the debug log. If empty, the default name of "debug.log" will be used and the file will be written to the application directory.

### **log\_severity**

Default value: "default"

The log severity. Only messages of this severity level or higher will be logged. Also configurable using the "log-severity" command-line switch with a value of "verbose", "info", "warning", "error", "error-report" or "disable". The default is "info".

### **reload\_page\_F5**

Default value: true

Reloading page with cache ignored will be available under the F5 key. See also the ["context\_menu"]["reload\_page"] option.

### **remote\_debugging\_port**

Default value: 0

Port for remote debugging. A value between 1024 and 65535 is accepted. A value of 0 will generate a random port. A value of -1 will disable remote debugging.