WOOCOMMERCE INTEGRATION KIT GUIDE

XLS PADLOCK - EXCEL PROTECTION SOFTWARE

G.D.G. SOFTWARE

Table of Contents

1. About this guide

XLS Padlock WooCommerce Integration Kit Guide

Version 2021
Copyright © G.D.G. Software 2018-2020

Last Revision: December 2020

For XLS Padlock version 2021 and further

Description

2. Description

Online activation and validation features of XLS Padlock let you **control user access to your Excel workbooks remotely**. For instance, you can sell subscriptions for your workbooks and block access of customers if their subscription expired for non payment.

The workbook applications compiled with XLS Padlock can communicate through a web application powered by this integration kit and your WooCommerce-powered website in order to manage activations automatically.

The following guide contains instructions to set up the web application, bind it to your WooCommerce website and your Excel workbooks. After that, online activation and optionally validation will be functional. You will be able to manage access of your customers directly from WooCommerce.
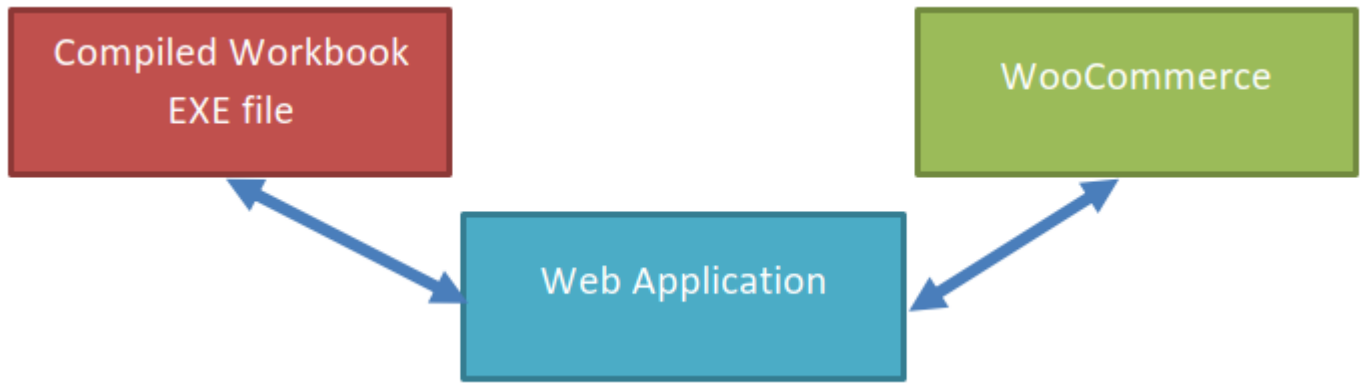
This integration kit does not require any special WordPress plug-in: it uses RESTful APIs.

Server requirements for the WooCommerce integration kit

3. Server requirements for the WooCommerce integration kit

The web application is based on the Fat-Free Framework (https://fatfreeframework.com) so they share the same server requirements: please refer to https://fatfreeframework.com/3.7/system-requirements

- ➢ PHP 5.6 or greater

- ➢ PHP CURL extension enabled

- ➢ PHP Archive PHAR must be enabled (by default it is)

- ➢ Your server must handle the PUT HTTP Method. Otherwise, please ask your webhost to enable it.

- ➢ mod_rewrite and mod_headers enabled (Apache)

- ➢ WooCommerce 3.5+

- ➢ WordPress 5+

Technical Support and license

4. Technical Support and license

Do not hesitate to send us your feedback, to report bugs or to share your modifications with us:

info@xlspadlock.com

This WooCommerce integration kit is provided "AS IS" WITHOUT WARRANTY OF ANY KIND, WHETHER EXPRESS, IMPLIED, OR STATUTORY, INCLUDING, BUT NOT LIMITED TO, THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THIS PRODUCT IS WITH YOU. SHOULD THIS PRODUCT PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR, OR CORRECTION.

By installing and using this WooCommerce integration kit, you agree to be bound by the license terms available from the LICENSE file in the folder named "**xlspadlock-woocom**".

Please note that, depending on your requests, we will <u>not</u> provide you with free technical support for this WooCommerce integration kit. You should have server and PHP knowledge at least to deal with the WooCommerce integration kit.

Step 1: enable REST API in WooCommerce and obtain API keys

5. Step 1: enable REST API in WooCommerce and obtain API keys

The web application uses RESTful to communicate with your WooCommerce website. REST support must then be enabled in WordPress/WooCommerce, and two API keys are also required.

→ **Follow instructions from this page to obtain the two REST API keys:**

https://docs.woocommerce.com/document/woocommerce-rest-api/

After following these instructions, before going further, you must have:

1. Enabled REST API in your WooCommerce website.

2. Created a user with READ/WRITE permissions:

3. Generated the two API keys (that begin with ck_ and cs_) and placed them in a text editor for reuse.

Here is a screenshot of the REST API tab of WooCommerce:

General	Products	Shipping	Checkout	Accounts	Emails	**API**

Settings | **Keys/Apps** | Webhooks

Keys/Apps   Add key

Bulk Actions ⌄	Apply			
☐ Description	Consumer key ending in	User		Permissions
☐ **XLS Padlock WooCommerce Helper**	▪️▪️	▪️▪️		Read/Write
☐ Description	Consumer key ending in	User		Permissions

Bulk Actions ⌄	Apply

Copy the files to your server

6. Step 2: install the Web application on your server and configure settings

6.1. Copy the files to your server

In the local folder on your computer where you unpacked the WooCommerce integration kit, you will find a folder named "**xlspadlock-woocom**".

Upload the entire contents of the "**xlspadlock-woocom**" folder to your web server. It can be in the root folder of your WordPress website, but it can be somewhere else too (especially if you get conflicts).

You can then rename the folder to something else if you want, for instance "**activation**" as shown below:
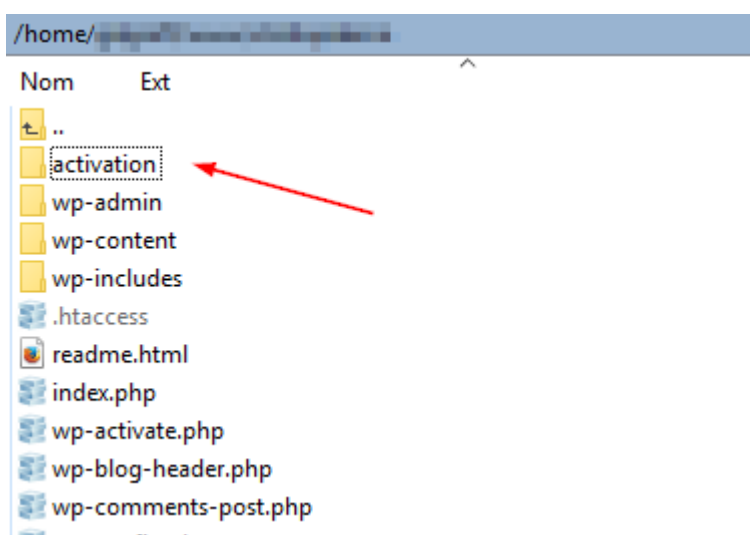
Open the "activation" folder and you should see:
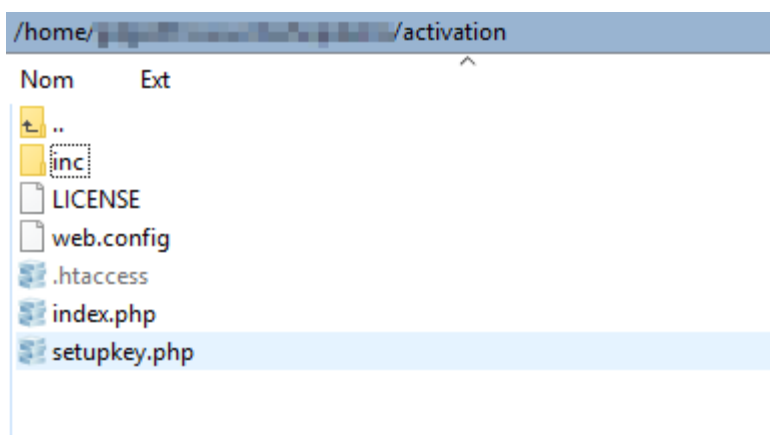
- If you are on Windows IIS, delete the unnecessary .htaccess file.

- If you are on Apache, delete the unnecessary web.config file.

[Base URL for activation](#)

6.2. Base URL for activation

Please ensure the **inc** subfolder is not web accessible (you should get a 403 forbidden error if you try to access it with a web browser): it contains sensitive data and should be protected by the shipped .htaccess file (on Apache) or the web.config file (on Windows IIS).

Write down the public URL that allows access to the "**xlspadlock-woocom**" folder because we will use it later.

In our case, since we renamed the "**xlspadlock-woocom**" to "**activation**", it should be:

> http://www.mywebsite.com/wordpress/**activation**

if you access your WordPress website at http://www.mywebsite.com/wordpress

---

[Set up local encryption key](#)

6.3. Set up local encryption key

Open your Web browser and navigate to the URL above + setupkey.php

With the example above, we would go to:

> http://www.mywebsite.com/wordpress/**activation/setupkey.php**

This PHP script will set up the encryption key used by the web application to exchange with the workbook EXE.

After the execution of the PHP script, your web browser should show:
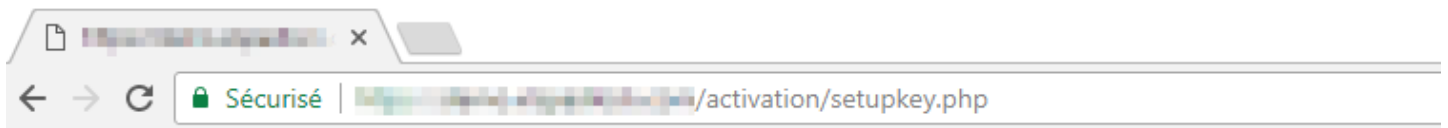
This encryption key is stored in a file at the location:

```
/activation/inc/lib/validationkey.txt
```

Once you are in production (i.e. you deployed your workbook EXE file made with XLS Padlock to customers), do not replace this file, otherwise end users will have to reactivate their workbook.

Finally, for security purpose, remove the setupkey.php file from your web server.
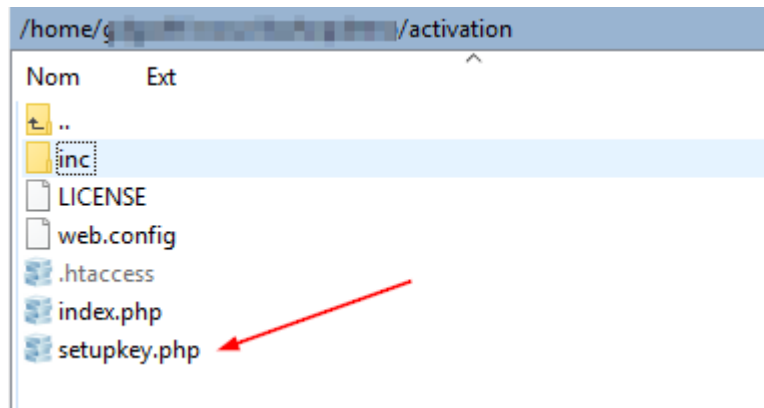
Step 3: configure mandatory settings in config.ini

7. Step 3: configure mandatory settings in config.ini

1. Navigate to the "**inc**" subfolder and edit the **config.ini** file stored on your server.
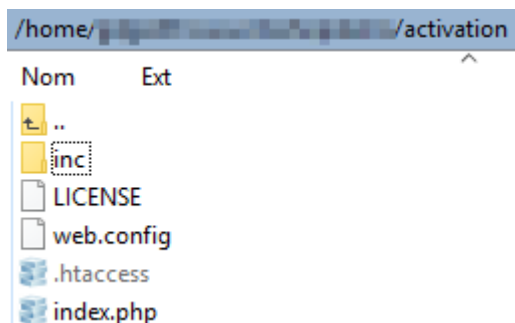
Once again, check that the config.ini file cannot be accessed publicly (the .htaccess prevents end users from accessing it) because this file contains sensitive data for generating activation keys.

The important part of the config.ini file looks like this:

[globals]

```
; ====== Modify the following parameters for your workbook ======
; ** Your XLS Padlock user ID **
xlspadlockuserid = "Your USER ID"

; ** Default maximum number of activations per order allowed (we also take account
of order quantity).
; For instance, if the customer bought a quantity of 2 products and with
defmaxactivperorder = 2, we would allow 4 activations.
defmaxactivperorder = 2

; ** Base URL to the WordPress/WooCommerce website - WARNING: END SLASH / MUST BE
INCLUDED
; To enable REST support for your WooCommerce install, please follow steps from
https://docs.woocommerce.com/document/woocommerce-rest-api/
wprestbaseurl = "http://mywebsite.com/wordpress/"

; ** WooCommerce REST API Consumer Key - see
https://docs.woocommerce.com/document/woocommerce-rest-api/
wprestconsumer_key = "ck_XXX"

; ** WooCommerce REST API Consumer Secret - see
https://docs.woocommerce.com/document/woocommerce-rest-api/
wpconsumer_secret = "cs_XXX"
```

2. Change the value of the entry named **xlspadlockuserid** to your XLS Padlock user ID. This user ID is available in the registration email that you received when you purchased XLS Padlock. **Important: remember that the user ID begins with xps-**

3. Enter the full base URL to your WordPress/WooCommerce website **between quotes** for the **wprestbaseurl** entry.

For instance, if your Wordpress website is publicly accessible at http://www.mywebsite.com/wordpress, you should then enter:

```
wprestbaseurl = "http://www.mywebsite.com/wordpress/"
```

4. Finally, provide the two API keys you generated from your WooCommerce Settings page at step 1: WooCommerce REST API Consumer Key in **wprestconsumer_key** and WooCommerce REST API Consumer Secret in **wpconsumer_secret**.

```
wprestconsumer_key = "ck_1254212121…"

wpconsumer_secret = "cs_4545452121…"
```

5. Save modifications to the **config.ini** file.

The web application is now ready to receive data from the compiled workbook EXE files and communicate with your WordPress/WooCommerce website.

Step 4: associate your compiled workbook EXE files with the web application

8. Step 4: associate your compiled workbook EXE files with the web application

The integration kit was designed to support multiple Excel workbooks at once. So, you must install the kit only once on your server even to manage multiple workbooks.

All the parameters of your Excel workbooks must be configured in a file called **workbooks.json** in JSON format.

You will find a template of the file in the "**inc**" directory of the application. We will see how to configure it in the following steps.

> ⚠️ Be careful, your file must respect the JSON format, otherwise it will not be loaded correctly by the application. You can test the validity of your file using online tools such as https://jsonlint.com/

1. Navigate to the "**inc**" subfolder and edit the **workbooks.json** file stored on your server.

Once again, check that the workbooks.json file cannot be accessed publicly (the .htaccess prevents end users from accessing it) because this file contains sensitive data for generating activation keys!

2. Here is the content of the workbooks.json file:

```
{
    "products": {
        "31": {
            "title": "My workbook title",
            "xlspadlockprivatekey": "{a04deb75-5288-4e0d-9254-58d44b63ecd4}",
            "xlspadlockmasterkey": "765DEB91C5886835F7E8740E72",
            "variations": {
                "32": {
                    "usehardwarelocking": 1,
                    "maxactivperorder": 2,
                    "keymaxdays": 180,
                    "shownagscreen": 0
                },
                "33": {
                    "usehardwarelocking": 1,
                    "keymaxdays": 10,
                    "maxactivperorder": 1,
                    "shownagscreen": 1
                }
            }
        }
    }
}
```

"products" is a mandatory section. Do not change the value. The schema used is the following:

```
{
    "products": {
        "product ID": {
            …
            "variations": {
            "variation ID 1": {
```

It is quite possible to define several products and therefore several Excel workbooks identified by their **product ID** (value

automatically assigned by WooCommerce when you add a product).

Similarly, **for each product, you can have several variations** and configure the WooCommerce Integration Kit to create activation keys dedicated to each product variation.

For example, you can have a "trial version" variation with a nag screen and a "registered version" variation for the same workbook that only lasts one year before mandatory renewal.

Many possibilities are available.

[Configuring the classical product in workbooks.json](#)

9. Associate to Classical Orders

9.1. Configuring the classical product in workbooks.json

The steps are for products which lead to classical orders in WooCommerce. They are not related to subscriptions.

1. Edit the **workbooks.json** file stored on your server.

2. If not done, go to the **WooCommerce admin page** and create the product that will be associated to your workbook. If you do not know how to do that, please see the WooCommerce documentation at https://docs.woocommerce.com/document/managing-products/

3. In the WooCommerce admin page, choose Products and place the cursor over the product you want. The ID will be displayed. On the screenshot below, it would be 68:
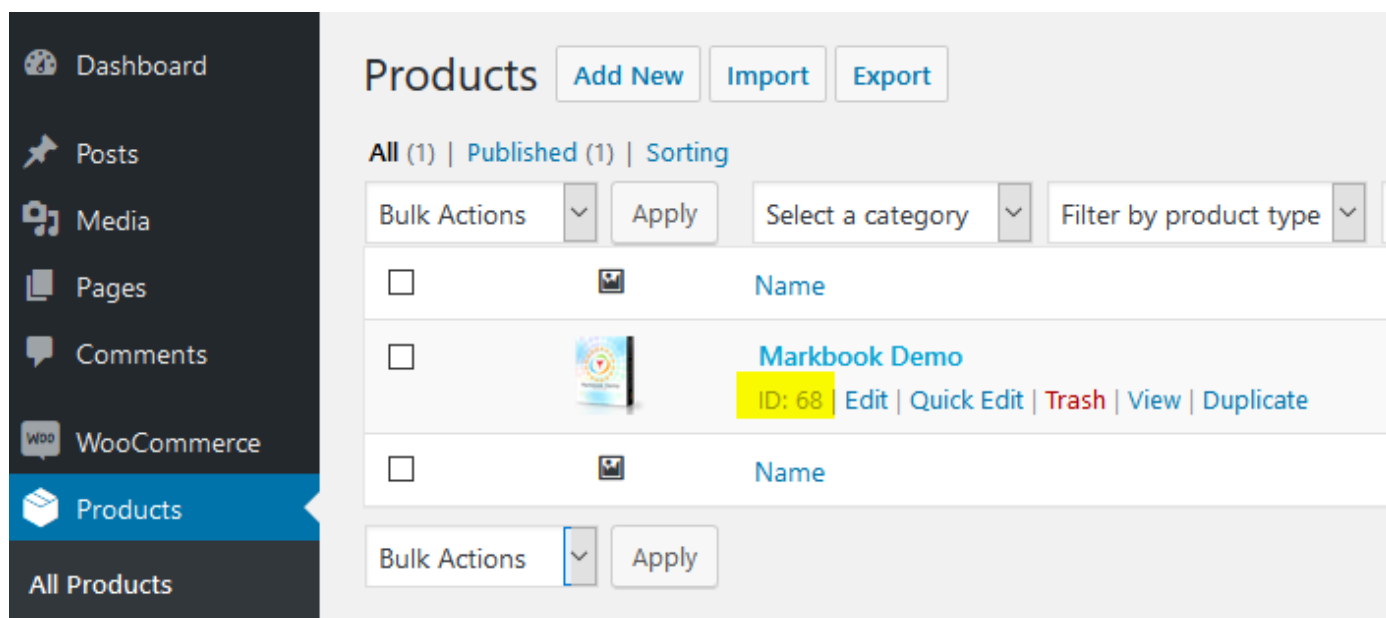
4. In the workbooks.json file, replace "product ID" with the product ID you obtained. In our case above, we would write "68":

```
{
    "products": {
        "68": {
                …
                "variations": {
                "variation ID 1": {
```

5. You can then give the title you want to your workbook. This is purely informative and can be useful to help you navigate when configuring multiple workbooks:

```
{
    "products": {
        "68": {
            "title": "My workbook title",
```

6. Start Excel, open your Excel workbook and XLS Padlock.

Navigate to the "**Activation Keys / Activation Settings**" page. Click the "**Copy**" button near the "**Application Master Key**" field:
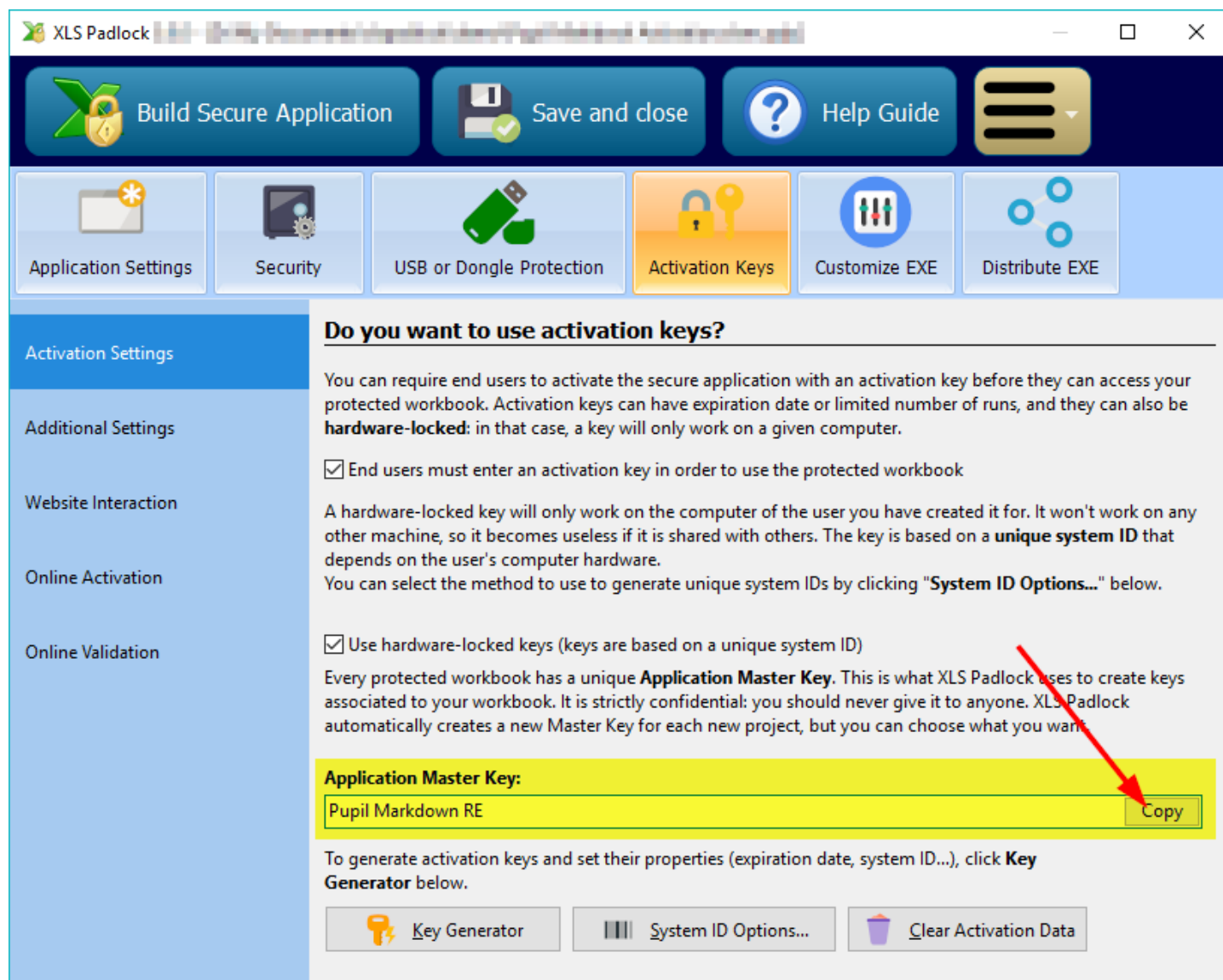
Paste this into workbooks.json for the value of the **xlspadlockmasterkey** entry. Based on the screenshot above, this would give:

```
{
    "products": {
        "68": {
            "title": "My workbook title",
            "xlspadlockmasterkey": "Pupil Markdown RE",
            "variations": {
                …
```

7. Back to XLS Padlock, choose **Online Activation** and click the "**Copy**" button near the "**Security Private Key**" field:
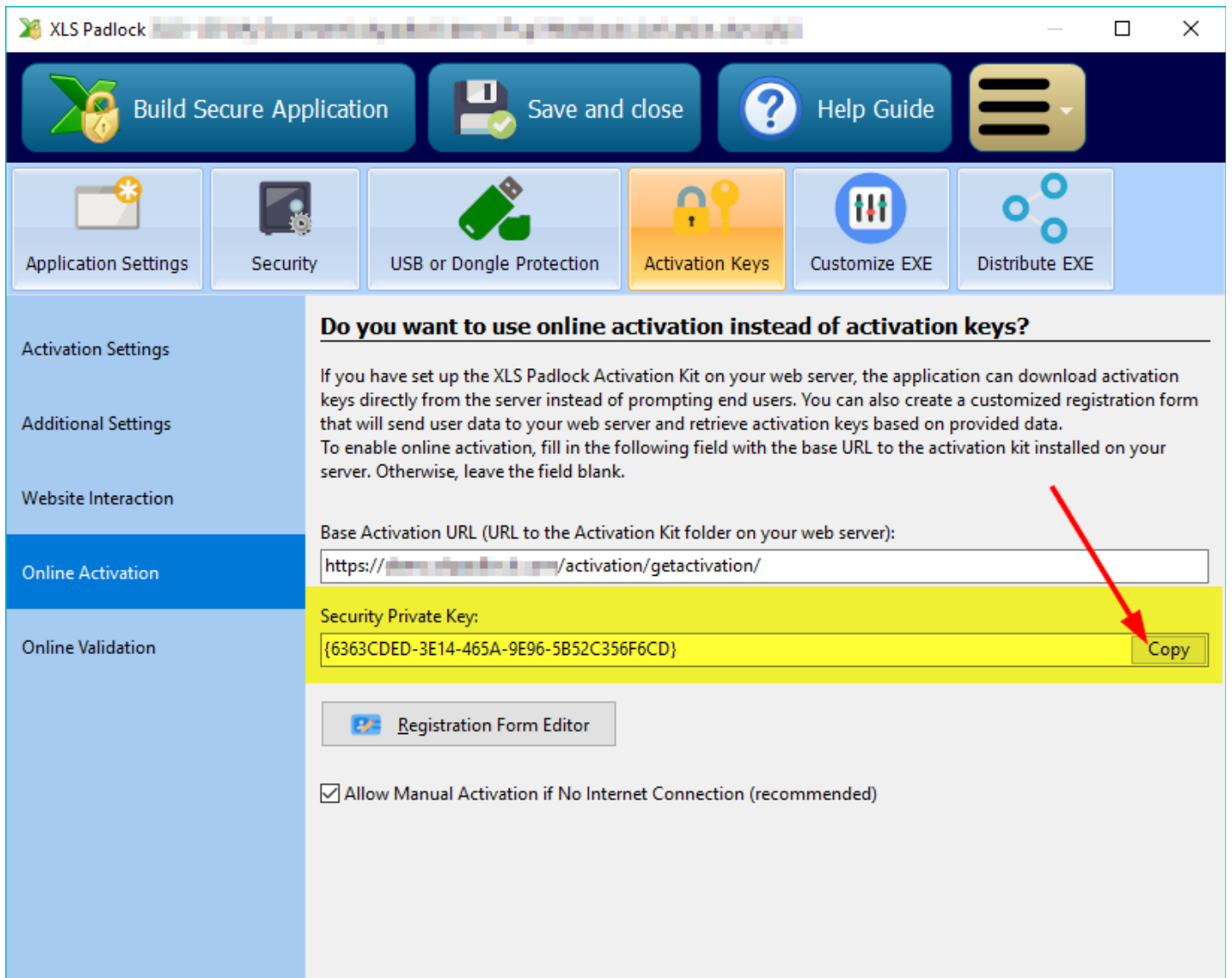
Paste this into workbooks.json for the value of the **xlspadlockprivatekey** entry. Based on the screenshot above, this would give:

```
{
    "products": {
        "31": {
            "title": "My workbook title",
            "xlspadlockprivatekey": "{6363CDED-3E14-465A-9E96-5B52C356F6CD}",
            "xlspadlockmasterkey": "Pupil Markdown RE",
            "variations": {
                    …
```

8. Now we will deal with the variations:

   ➢ if your product does not contain any variations, enter "0" as the variation ID.

   ➢ Otherwise, you will have to create a "variation" block for each variation present in your product and configure the corresponding properties.

To obtain the variation ID, in your WooCommerce admin, choose your product in "Products". Then, in "Product Data", click the "Variations" tab and the variation ID are listed after each hash, as you can see them highlighted on the screenshot below:
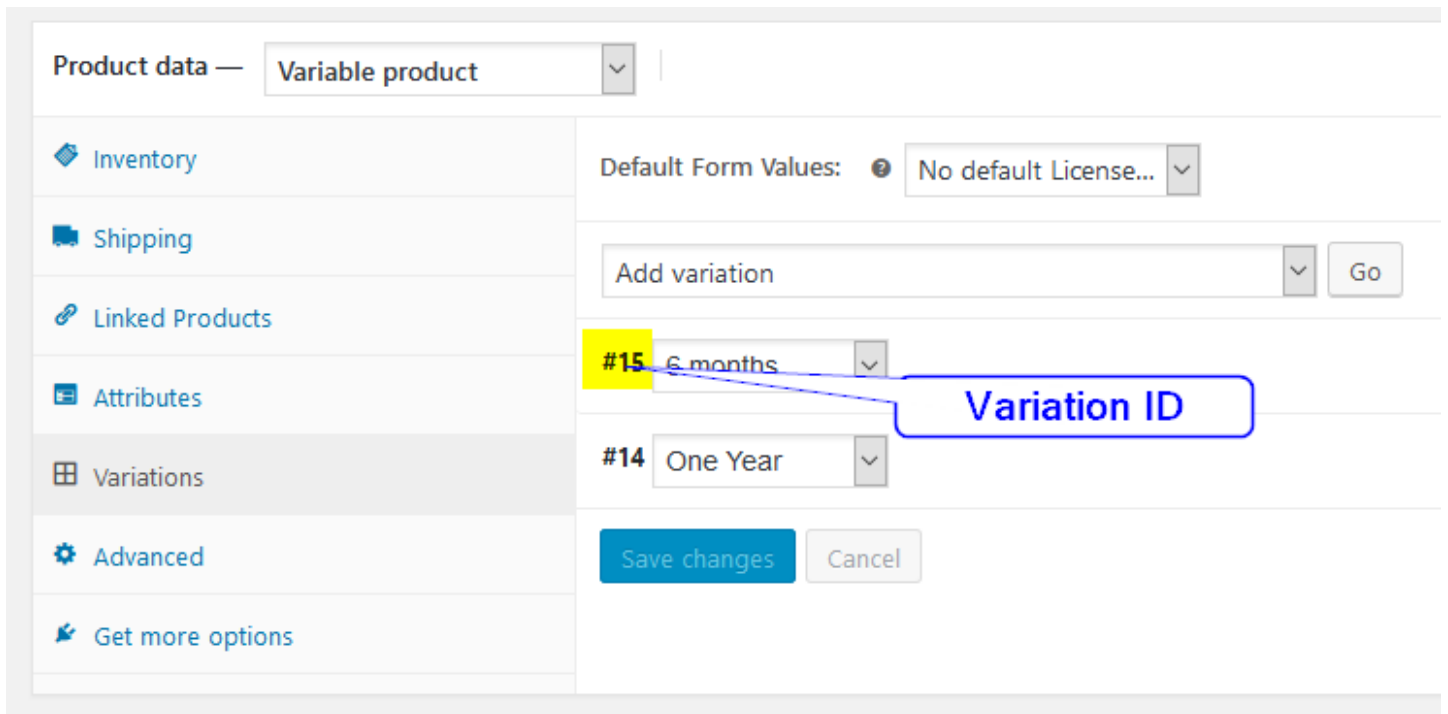
Go to the workbooks.json and locate the "variation" block. Enter your variation ID for the first variation block. If your product has no variation defined in WooCommerce, enter "0" as the ID. In our sample, we would enter 15 as below:

```
{
    "products": {
        "31": {
            "title": "My workbook title",
            "xlspadlockprivatekey": "{6363CDED-3E14-465A-9E96-5B52C356F6CD}",
            "xlspadlockmasterkey": "Pupil Markdown RE",
            "variations": {
                15: {
                    "usehardwarelocking": 1,
                    "maxactivperorder": 2,
                    "keymaxdays": 180,
                    "shownagscreen": 0
                },
    …
```

9. Now we can configure the different properties for that variation:

    ➢ If you do not want to generate hardware-locked activation keys (NOT RECOMMENDED), change the **usehardwarelocking** entry from 1 to 0.

    ➢ Define the number of activations per order allowed thanks to the **maxactivperorder** entry. This lets you define on how many machines an end user (for instance who purchased your compiled workbook EXE) can activate the workbook. By default, it's the value defined in the config.ini (see above).

Note: the order quantity is also taken in account. For instance, if the customer bought a quantity of 2 products and with maxactivperorder = 2, we would allow 4 activations.

    a. If you want to generate activation keys that would display a nag screen (useful for trials), change the **shownagscreen** entry from 0 to 1.

    b. Finally, if you want to set an expiration date or maximum number of runs / days, it's possible. Remember that XLS Padlock only allows one expiration mode: date, number of runs or number of days per activation key.

        i. For a maximum number of runs, define the entry named "**keymaxruns**" and set its value to the

number of runs you want to allow.

    ii. For a maximum number of days, define the entry named "**keymaxdays**" and set its value to the number of days you want to allow.

    iii.   For a fixed expiration date, define the entry named "**keyexpiredate**" and set its value to the date with the following mandatory format: YYYY/MM/DD

10. Repeat the same steps for all variations in your product. For instance, below you can see that we defined two variations (one with a fixed expiration date and another with a number of allowed days):

```json
{
    "products": {
        "31": {
            "title": "My workbook title",
            "xlspadlockprivatekey": "{6363CDED-3E14-465A-9E96-5B52C356F6CD}",
            "xlspadlockmasterkey": "Pupil Markdown RE",
            "variations": {
            15: {
                    "usehardwarelocking": 1,
                    "maxactivperorder": 2,
                    "keymaxdays": 180,
                    "shownagscreen": 0
                },
            16: {
                    "usehardwarelocking": 1,
                    "maxactivperorder": 1,
                    "keyexpiredate": "2020/10/04",
                    "shownagscreen": 1
                },
            }
    …
```

Configuring the Excel workbook

9.2. Configuring the Excel workbook

1. In your XLS Padlock project, navigate to the "**Activation Keys / Online Activation**" page.

In the **Base Activation URL** field, enter the base URL as defined previously (see Base URL for activation) on your web server followed by **/getactivation/{product ID}** (a final slash / is not mandatory) and replace **{product ID}** with the product ID associated to your workbook (and defined in workbooks.json).

In our case started earlier, it would be:

```
https://www.mywebsite.com/wordpress/activation/getactivation/68
```

If the product ID is omitted, the first product defined in workbooks.json will be used.

XLS Padlock can handle TLS/SSL connections: if your web server has a valid SSL certificate, you can use secure URLs that begin with **HTTPS**.

2. Click **Registration Form Editor**. XLS Padlock shows a basic HTML editor and a preview of the dialog's text. You can define your own HTML text but at least an HTML field named **token is required**.

> ⚠️ To identify the order in the WooCommerce database, the web application expects the compiled workbook EXE to send a **unique order identifier** called **token**. For your end users, it's an activation code but internally we will name it the "token".

For instance, we can use the following HTML code for the registration form editor:

```
<FONT size="10"><B>Activation Required</B></FONT><br><br>
Welcome to this sample application made with Excel and XLS Padlock. <A
href="https://demo.xlspadlock.com">Learn more about this application</A><br><br>To
access this application, please enter
the activation code that you received after your order and press <B>Activate</B>.
If your computer has no Internet connection,
choose <B>Manual Activation</B>.<br><br>
Your Activation Code:<br><CONTROL TYPE="EDIT" WIDTH="280" VALUE="" ID="token">
```

Only this part is mandatory because it defines the HTML field that lets the user enter his activation token:

```
<CONTROL TYPE="EDIT" WIDTH="280" VALUE="" ID="token">
```

Save your changes and close the editor.

3. Compile your workbook EXE application. It is now associated to the web application.

9.3. Description of the activation process

To determine the number of allowed activations for your compiled workbook EXE, the web application stores custom data into WooCommerce orders thanks to the Custom Fields feature.
When a customer purchases your compiled workbook EXE through WooCommerce, an order is generated and appears in the Orders list:

You can also test your activation process by creating manual orders with the button **Add order** above.

Activation and validation processes managed by the web application use the **activation token** that identifies the order in the WooCommerce database (see Configuring the Excel workbook).

All activation tokens expected by the web application must be in the following format:
CUSTOMER EMAIL-ORDER ID

For instance, info@gdgsoft.com-76 is a valid activation token. The web application will use the number after the - character to identify the order ID in the WooCommerce database. Then, it will check whether the recorded email address

suits the email provided in the activation token before the - character.

This activation token must be sent to the customer who purchased your workbook. This token lets him activate the workbook. Some WooCommerce plug-ins allow sending emails to customers after purchase (see Sending activation token to customers): be sure to include the activation token in the email.

When a customer runs the compiled workbook EXE file, a dialog box appears telling him that an activation is required. As said before, the dialog box must at least ask for the activation token (you can name it what you want but the associated HTML field ID must be "token", see Configuring the Excel workbook).
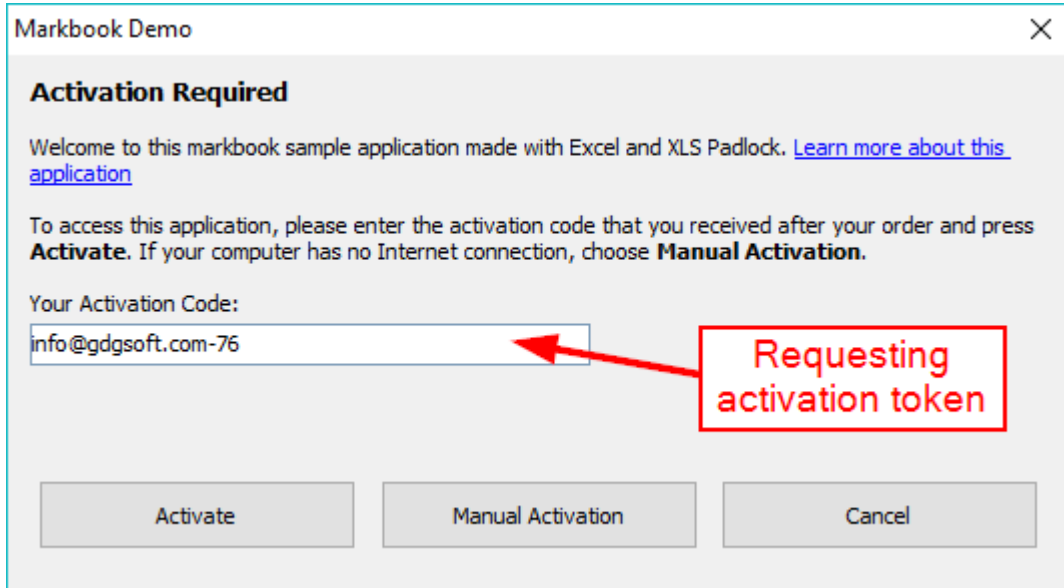
When the customer clicks Activate, data is sent to the activation server:
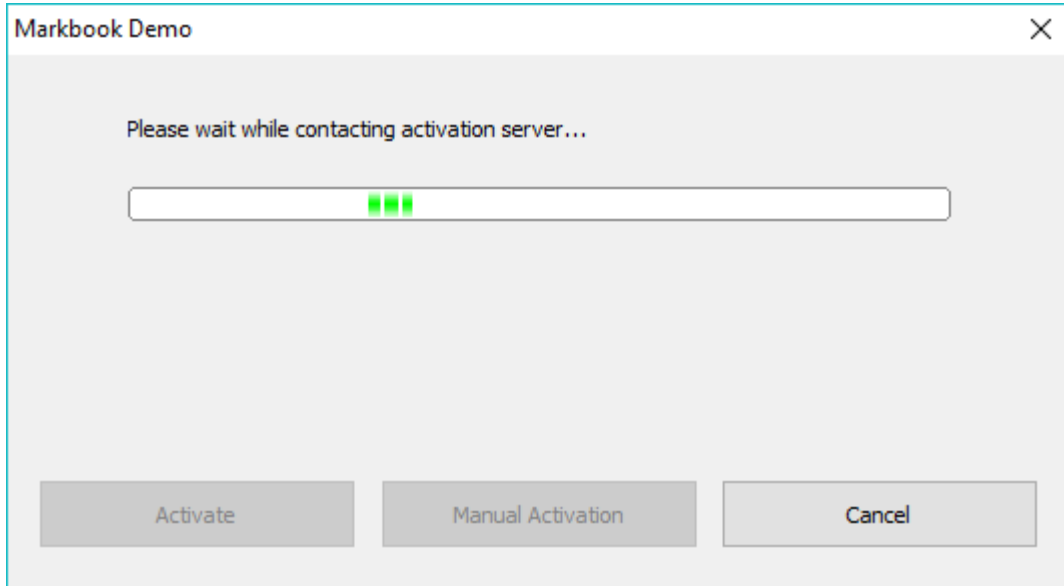
The WooCommerce kit on the activation server will check the order's status and different settings.
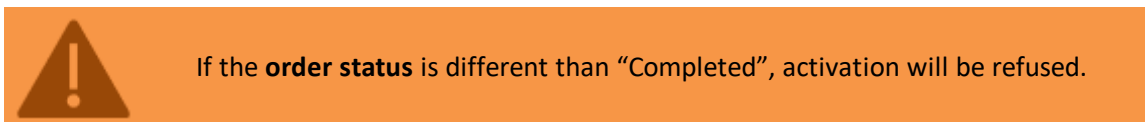
⚠️ If the **order status** is different than "Completed", activation will be refused.

It will be refused if the status is pending, processing, on-hold, cancelled, refunded or failed. This ensures that only paid customers can activate your workbook.

After a successful activation, a confirmation message is shown, and the compiled workbook EXE is restarted. If an error

occurs, the corresponding message box is shown, and customer can try to activate again.

9.4. Sending activation token to customers

As explained in the description of the activation process, the activation token **CUSTOMER EMAIL-ORDER ID** must be sent to the customer who purchased your workbook. This token lets him activate the workbook.

Some free WooCommerce plug-ins allow sending emails to customers after purchase.

For instance, Woo Custom Emails available at https://wordpress.org/plugins/woo-custom-emails-per-product/

To insert the the activation token in the email template, use:

$$\{order\_billing\_email\}-\{order\_number\}$$

Choose Order Status
(Choose order statuses when changed this email should fire.)

Pending payment ▼	Completed ▼	± :
Processing ▼	Completed ▼	± :
On hold ▼	Completed ▼	± :
Cancelled ▼	Completed ▼	± :
Refunded ▼	Completed ▼	± :
Failed ▼	Completed ▼	± :

Template
(Use these tags to to print them in email. -
{order_date}, {order_number},
{woocommerce_email_order_meta},
{order_billing_name},
{email_order_items_table},
{email_order_total_footer},
{order_billing_email},
{order_billing_phone}, {email_addresses}
)

Add Media   Add Form

Paragraph ▼ B I ≔ ≔ 66 ≡ ≡ ≡ 𝒫 ▦ ⤢ ⌨
ABE — A ▼ 📋 ⬦ Ω ⏛ ⏩ ↶ ↷ ❓

Your activation token is {order_billing_email}–{order_number}

9.5. Manage existing activations / block customers

9.5.1. Using WooCommerce custom fields

Use the WooCommerce **Orders** section and then **Edit order** to manage activations of a given order:
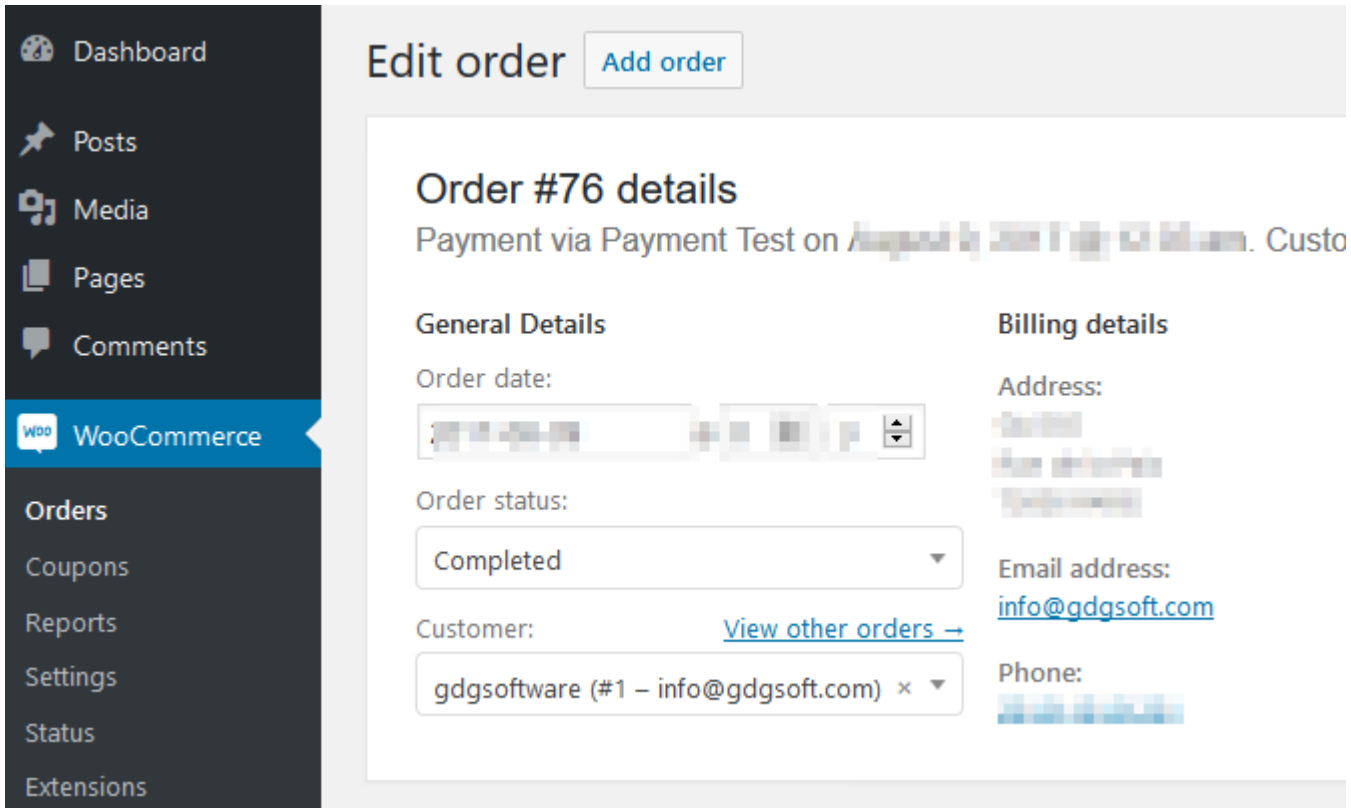
When the customer activates the compiled workbook EXE, the web application modifies the corresponding WooCommerce order by adding a custom field named **xlspadlock_activations:**
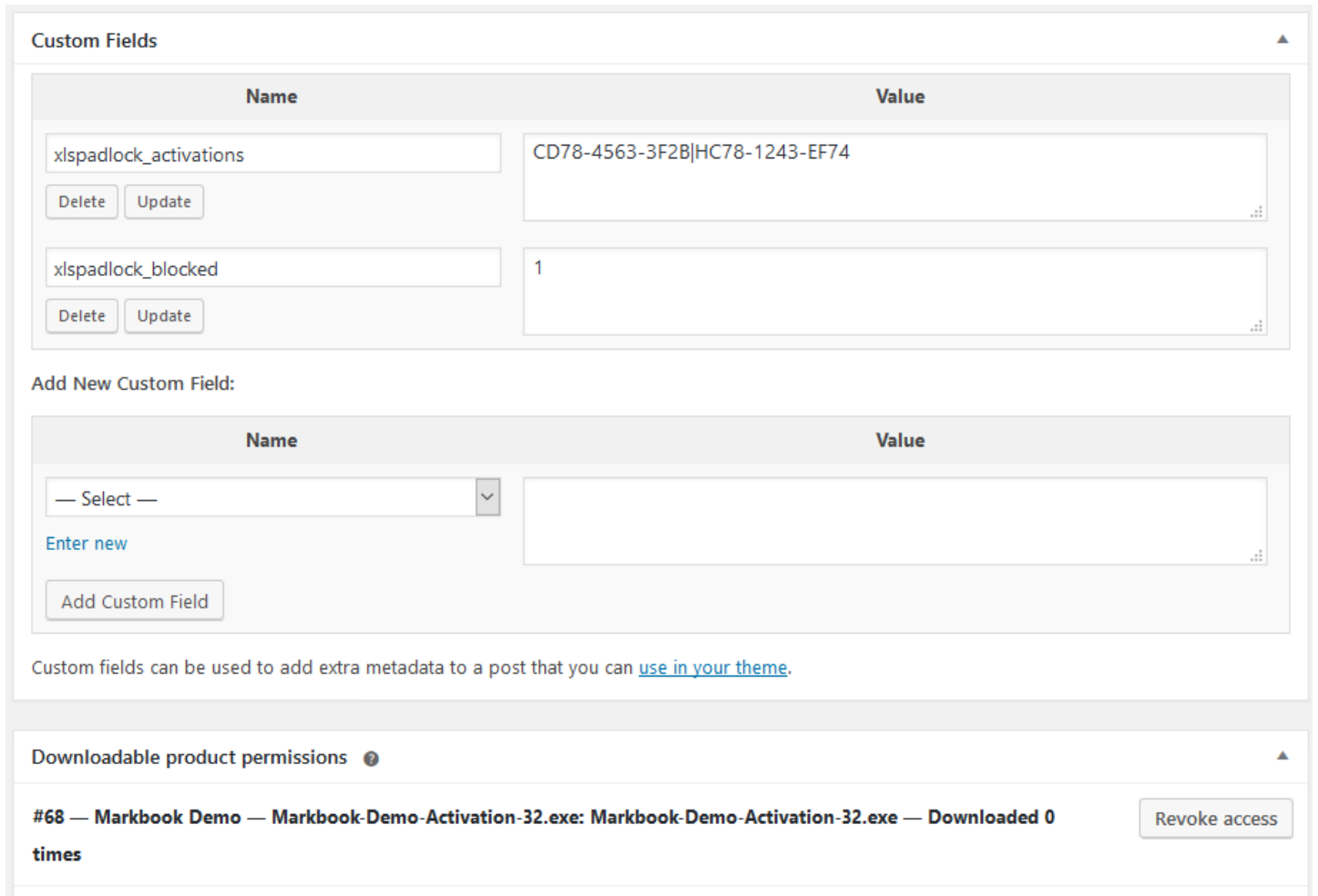
The value of the **xlspadlock_activations** custom field contains a **list of system IDs**. These unique system IDs are the ones of the computers the customer activated the compiled workbook EXE on. System IDs are separated by the | character.

For instance, "CD78-4563-3F2B|HC78-1243-EF74" indicates that the compiled workbook EXE was activated on two different computers. If you configured the web application to allow 2 activations only, next time an error will be triggered (no more activations left).

You can manually clear existing activations by removing the corresponding system ID.

Note: each time the customer activates the compiled workbook EXE file, a note is added to the WooCommerce order, as you can see here:

XLS Padlock Notification: Workbook
Activated

added on August 16, 2017 at 1:03 pm **Delete note**

Blocking a customer

9.5.2. Blocking a customer

You can block activations if your customer for instance asked for a refund.

The WooCommerce Kit automatically blocks any activation if:

- The order status is different than "Completed". If the status is pending, processing, on-hold, cancelled, refunded or failed, activation and validation will be refused.

- If the "xlspadlock_blocked" custom field is set to "1".

This can be done by adding a custom field to the WooCommerce order with the following properties:

- Name: xlspadlock_blocked

- Value: 1

Custom Fields ▲

Name	Value
xlspadlock_activations [Delete] [Update]	CD78-4563-3F2B|HC78-1243-EF74
xlspadlock_blocked [Delete] [Update]	1

Add New Custom Field:

Name	Value
— Select — ▾ Enter new [Add Custom Field]	

Custom fields can be used to add extra metadata to a post that you can use in your theme.

Downloadable product permissions ⊘ ▲

#68 — Markbook Demo — Markbook-Demo-Activation-32.exe: Markbook-Demo-Activation-32.exe — Downloaded 0 times   [Revoke access]

To unblock the order, set the value to 0 or simply remove the custom field "xlspadlock_blocked".

Description of the validation process

9.6. Description of the validation process

The WooCommerce Integration Kit supports the validation feature of XLS Padlock: you can configure your compiled workbook EXE to **check the validity of the activation regularly**.

> 💡 **This feature is only available if online activation is also used**.

Possible uses:
- you sell Excel workbooks and you want to block access to them if end users ask for refunds.
- you want to control who can access Excel workbooks regularly.

To activate the validation feature, fill in the base validation URL field.
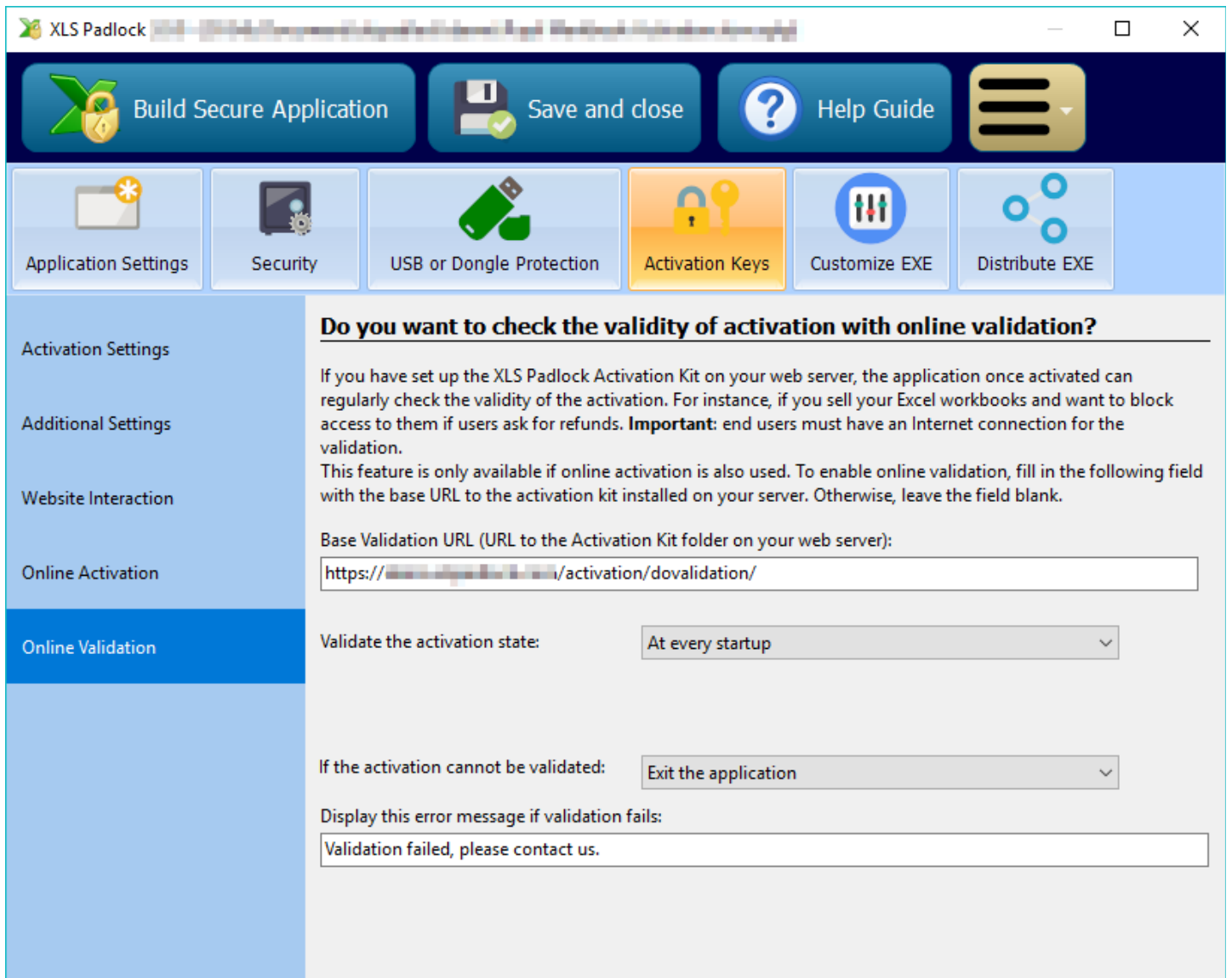
In the **Base Validation URL** field, enter the base URL as defined previously (see Base URL for activation) on your web server **followed by /dovalidation/{product ID}** (a final slash / is not mandatory) and replace **{product ID}** with the product ID associated to your workbook (and defined in workbooks.json).

In our case, it would be:

```
https://www.mywebsite.com/wordpress/activation/dovalidation/68
```

Then, define the frequency of validation and what should happen if the validation fails. Please refer to the XLS Padlock User Guide for further details about each setting.

Recompile your workbook EXE with XLS Padlock.

The compiled workbook EXE can now perform validation at startup:
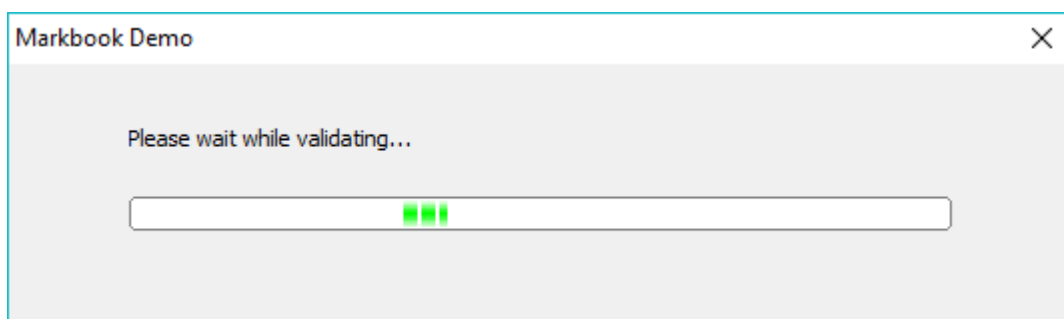
Suppose that you blocked the access of your customer (see Blocking a customer), they will get this error (returned by the web application):
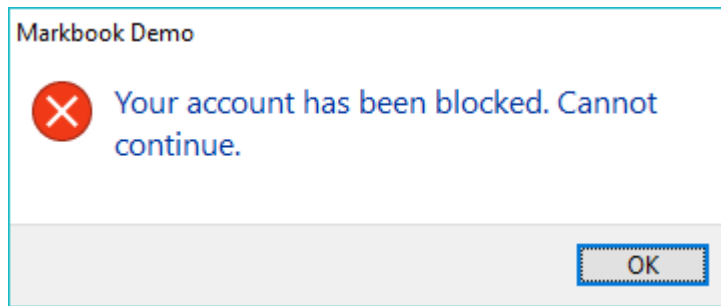
Otherwise, if validation is successful, the compiled workbook EXE continues its execution.

During validation, the WooCommerce kit also checks the order's status.
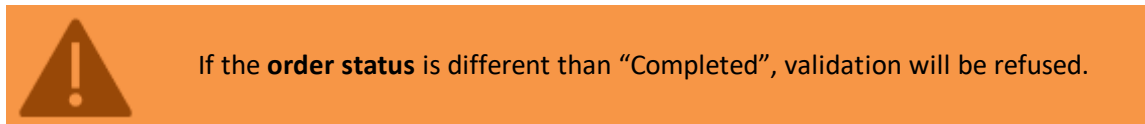
> ⚠️ If the **order status** is different than "Completed", validation will be refused.

It will be refused if the status is pending, processing, on-hold, cancelled, refunded or failed. This ensures that only paid customers are validated.

Configuring the subscription product in workbooks.json

10. Associate to Subscriptions

10.1. Configuring the subscription product in workbooks.json

The steps below are for subscription products.

> ⚠️ The [WooCommerce Subscription Add-in](#) is required.

1. Edit the **workbooks.json** file stored on your server.

2. If not done, go to the **WooCommerce admin page** and create the subscription product that will be associated to your workbook. If you do not know how to do that, please see the WooCommerce documentation at https://docs.woocommerce.com/document/subscriptions/

3. In the WooCommerce admin page, choose Products and place the cursor over the subscription product you want. The ID will be displayed. On the screenshot below, it would be 13:
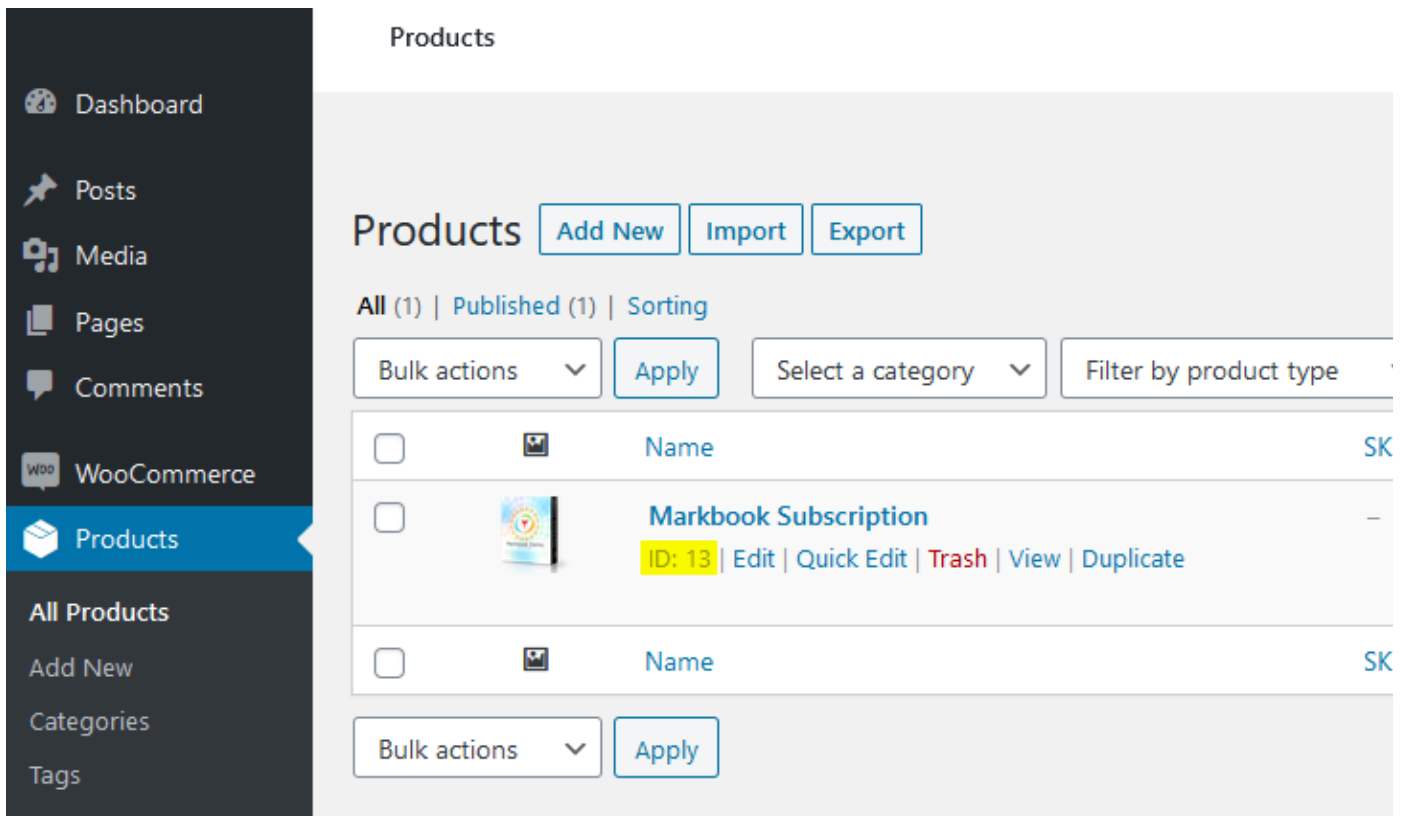
4. In the workbooks.json file, replace "product ID" with the product ID you obtained. In our case above, we would write "13":

```
{
    "products": {
        "13": {
            …
            "variations": {
            "variation ID 1": {
```

5. You can then give the title you want to your workbook. This is purely informative and can be useful to help you navigate when configuring multiple workbooks:

```
{
      "products": {
            "13": {
                  "title": "My workbook title",
```

6. Start Excel, open your Excel workbook and XLS Padlock.

Navigate to the "**Activation Keys / Activation Settings**" page. Click the "**Copy**" button near the "**Application Master Key**" field:
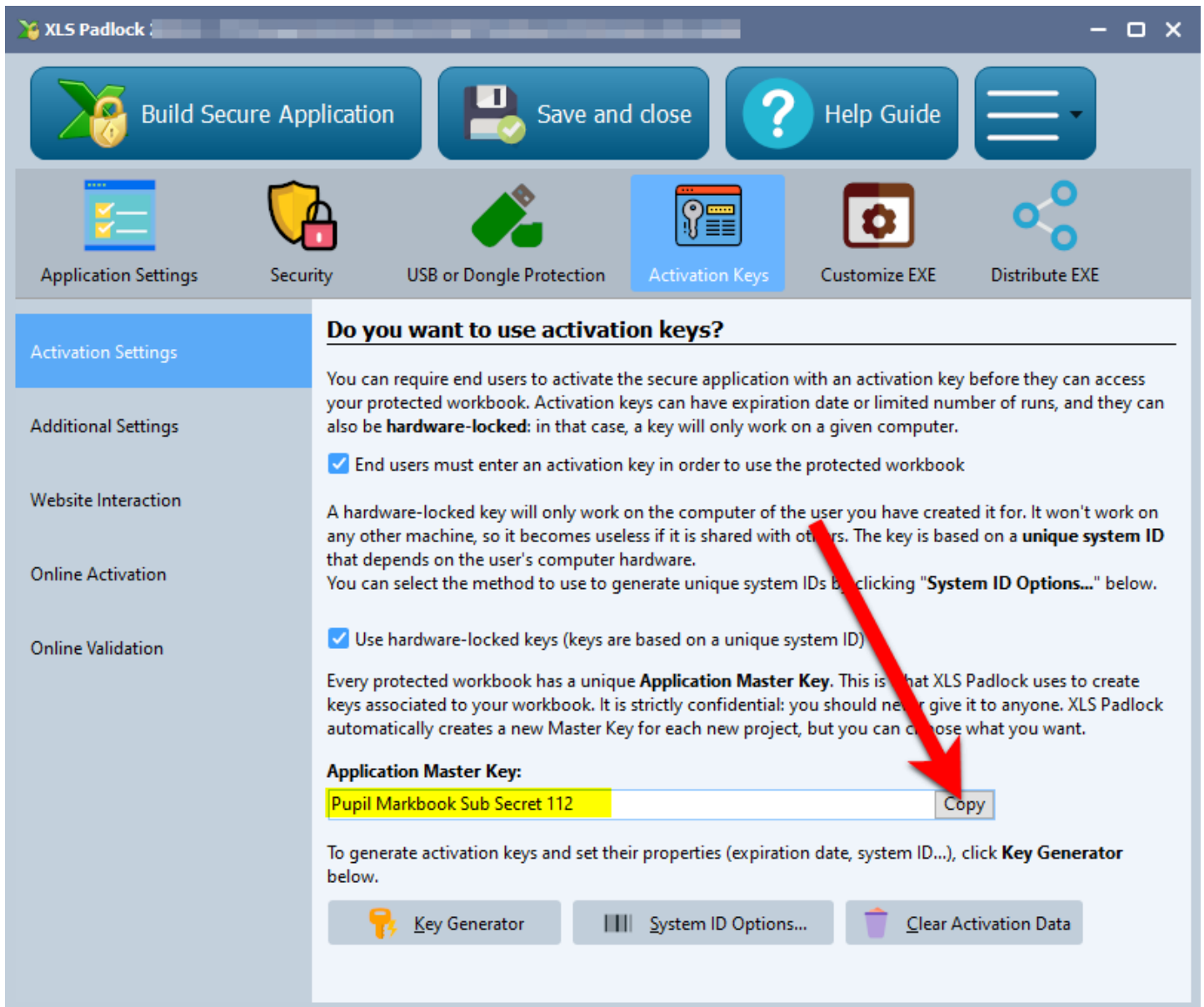
Paste this into workbooks.json for the value of the **xlspadlockmasterkey** entry. Based on the screenshot above, this would give:

```
{
      "products": {
            "13": {
                  "title": "My workbook title",
                  "xlspadlockmasterkey": "Pupil Markbook Sub Secret 112",
                  "variations": {
                        …
```

7. Back to XLS Padlock, choose **Online Activation** and click the "**Copy**" button near the "**Security Private Key**" field:
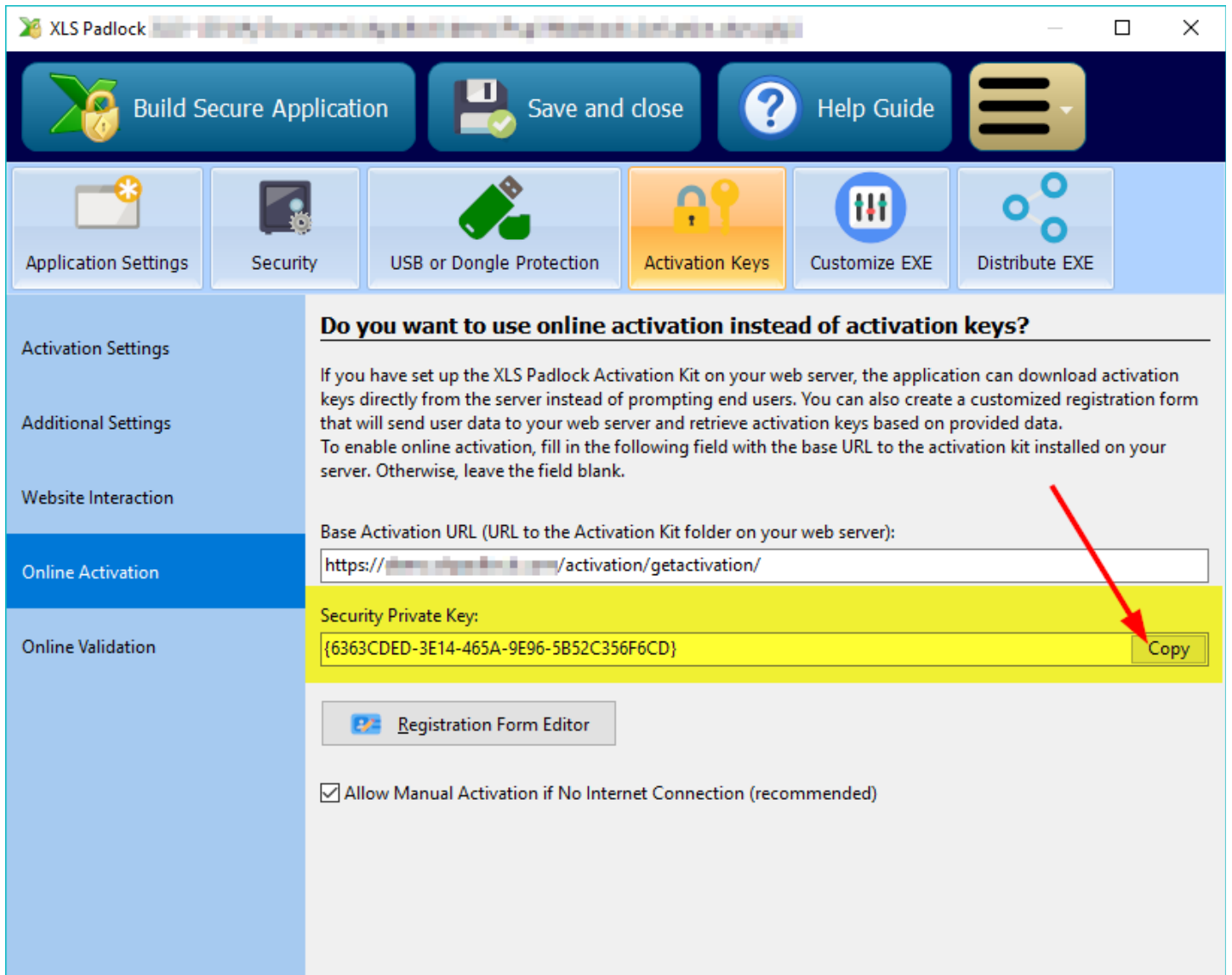
Paste this into workbooks.json for the value of the **xlspadlockprivatekey** entry. Based on the screenshot above, this would give:

```
{
    "products": {
        "31": {
            "title": "My workbook title",
            "xlspadlockprivatekey": "{6363CDED-3E14-465A-9E96-5B52C356F6CD}",
            "xlspadlockmasterkey": "Pupil Markbook Sub Secret 112",
            "variations": {
                …
```

8. Now we will deal with the variations:

➢ if your subscription product does not contain any variations, enter "0" as the variation ID.

➢ Otherwise, if you have a variable subscription, you will have to create a "variation" block for each variation present in your product and configure the corresponding properties.

To obtain the variation ID, in your WooCommerce admin, choose your product in "Products". Then, in "Product Data", click the "Variations" tab and the variation ID are listed after each hash, as you can see them highlighted on the screenshot below:

Go to the workbooks.json and locate the "variation" block. Enter your variation ID for the first variation block. If your subscription product has no variation defined in WooCommerce, enter "**0**" as the ID. In our sample, we would enter 21 as below:

```json
{
    "products": {
        "13": {
            "title": "My workbook title",
            "xlspadlockprivatekey": "{6363CDED-3E14-465A-9E96-5B52C356F6CD}",
            "xlspadlockmasterkey": "Pupil Markbook Sub Secret 112",
            "variations": {
                21: {
                    "usehardwarelocking": 1,
                    "maxactivperorder": 2,
                    "keymaxdays": 180,
                    "shownagscreen": 0
                },
    …
```

9. Now we can configure the different properties for that variation:

   ➢ If you do not want to generate hardware-locked activation keys (NOT RECOMMENDED), change the **usehardwarelocking** entry from 1 to 0.

   ➢ Define the number of activations per order allowed thanks to the **maxactivperorder** entry. This lets you define on how many machines an end user (for instance who purchased your compiled workbook EXE) can activate the workbook. By default, it's the value defined in the config.ini (see above).

Note: the order quantity is also taken in account. For instance, if the customer bought a quantity of 2 products and with maxactivperorder = 2, we would allow 4 activations.

   a. If you want to generate activation keys that would display a nag screen (useful for trials), change the **shownagscreen** entry from 0 to 1.

   b. Finally, if you want to set an expiration date or maximum number of runs / days, it's possible. Remember that XLS Padlock only allows one expiration mode: date, number of runs or number of days per activation key.

      i. For a maximum number of runs, define the entry named "**keymaxruns**" and set its value to the number of runs you want to allow.

      ii. For a maximum number of days, define the entry named "**keymaxdays**" and set its value to the

number of days you want to allow.

    iii.   For a fixed expiration date, define the entry named "**keyexpiredate**" and set its value to the date with the following mandatory format: YYYY/MM/DD

10. Repeat the same steps for all variations in your product. For instance, below you can see that we defined two variations (one with a fixed expiration date and another with a number of allowed days):

```json
{
    "products": {
        "13": {
            "title": "My workbook title",
            "xlspadlockprivatekey": "{6363CDED-3E14-465A-9E96-5B52C356F6CD}",
            "xlspadlockmasterkey": "Pupil Markbook Sub Secret 112",
            "variations": {
            21: {
                    "usehardwarelocking": 1,
                    "maxactivperorder": 2,
                    "keymaxdays": 180,
                    "shownagscreen": 0
                },
            22: {
                    "usehardwarelocking": 1,
                    "maxactivperorder": 4,
                    "shownagscreen": 0
                },
            }
        …
```

Configuring the Excel workbook for subscriptions

10.2. Configuring the Excel workbook for subscriptions

◁ ▷

1. In your XLS Padlock project, navigate to the "**Activation Keys / Online Activation**" page.

In the **Base Activation URL** field, enter the base URL as defined previously (see Base URL for activation) on your web server followed by **/getsubactivation/{product ID}** (a final slash / is not mandatory) and replace **{product ID}** with the product ID associated to your workbook (and defined in workbooks.json).

In our case started earlier, it would be:

```
https://www.mywebsite.com/wordpress/activation/getsubactivation/13
```

If the product ID is omitted, the first product defined in workbooks.json will be used.

XLS Padlock can handle TLS/SSL connections: if your web server has a valid SSL certificate, you can use secure URLs that begin with **HTTPS**.

2. Click **Registration Form Editor**. XLS Padlock shows a basic HTML editor and a preview of the dialog's text. You can define your own HTML text but at least an HTML field named **token is required**.

> ⚠️ To identify the order in the WooCommerce database, the web application expects the compiled workbook EXE to send a **unique order identifier** called **token**. For your end users, it's an activation code but internally we will name it the "token".

For instance, we can use the following HTML code for the registration form editor:

```
<FONT size="10"><B>Activation Required</B></FONT><br><br>
Welcome to this sample application made with Excel and XLS Padlock. <A
href="https://demo.xlspadlock.com">Learn more about this application</A><br><br>To
access this application, please enter
the activation code that you received after your order and press <B>Activate</B>.
If your computer has no Internet connection,
choose <B>Manual Activation</B>.<br><br>
Your Activation Code:<br><CONTROL TYPE="EDIT" WIDTH="280" VALUE="" ID="token">
```

Only this part is mandatory because it defines the HTML field that lets the user enter his activation token:

```
<CONTROL TYPE="EDIT" WIDTH="280" VALUE="" ID="token">
```

Save your changes and close the editor.

3. Compile your workbook EXE application. It is now associated to the web application.

10.3. Description of the activation process (subscriptions)

To determine the number of allowed activations for your compiled workbook EXE, the web application stores custom data into WooCommerce orders thanks to the Custom Fields feature.

When a customer purchases **a subscription for your compiled workbook EXE** through WooCommerce, an initial order is generated and appears in the Orders list. This initial order which created the subscription is named the **parent order**.

For a given subscription, you can find the parent order by editing the subscription and in the Related Orders section, the Parent Order is indicated:

We expect **each subscription to have a parent order** where to store activation data. If you create subscriptions manually, be sure to associate a parent order to them!

Activation and validation processes **for a subscription** managed by the web application use the **activation token** that identifies the subscription in the WooCommerce database (see Configuring the Excel workbook for subscriptions).

All activation tokens expected by the web application for subscriptions must be in the following format:
**CUSTOMER EMAIL-SUBSCRIPTION ID**

For instance, info@gdgsoft.com-16 is a valid activation token. The web application will use the number after the - character to identify the subscription ID in the WooCommerce database. Then, it will check whether the email address stored in the subscription's parent order suits the email provided in the activation token before the - character.

This activation token must be sent to the customer who purchased your workbook. This token lets him activate the workbook. Some WooCommerce plug-ins allow sending emails to customers after purchase (see Sending activation token to customers in case of subscriptions): be sure to include the activation token in the email.
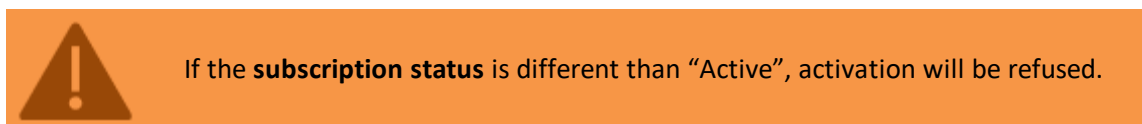
When a customer runs the compiled workbook EXE file, a dialog box appears telling him that an activation is required. As said before, the dialog box must at least ask for the activation token (you can name it what you want but the associated HTML field ID must be "token", see onfiguring the Excel workbook for subscriptions).
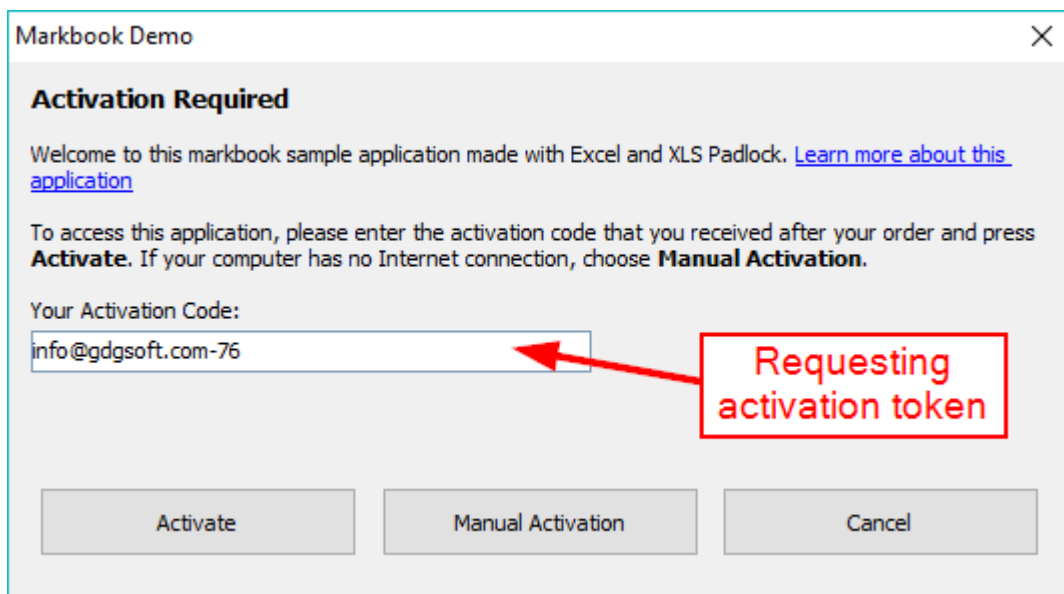
When the customer clicks Activate, data is sent to the activation server:
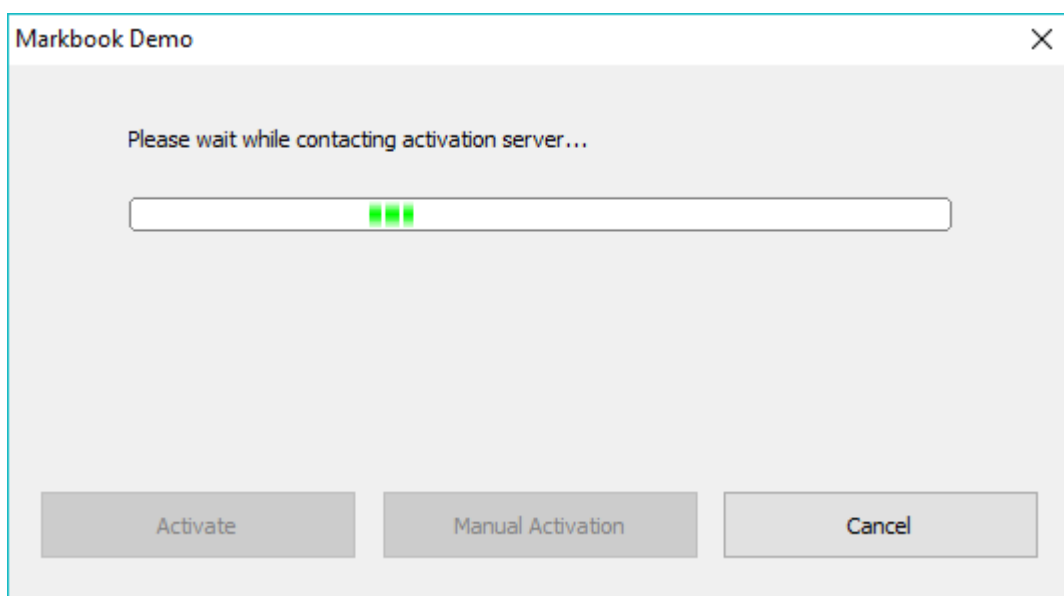
The WooCommerce kit on the activation server will check the subscription's status and different settings.

⚠️ If the **subscription status** is different than "Active", activation will be refused.

This ensures that only paid customers with an active subscription can activate your workbook.

After a successful activation, a confirmation message is shown, and the compiled workbook EXE is restarted. If an error occurs, the corresponding message box is shown, and customer can try to activate again.

Sending activation token to customers (subscriptions)

10.4. Sending activation token to customers (subscriptions)

As explained in the [description of the activation process](#) for subscriptions, the activation token **CUSTOMER EMAIL-SUBSCRIPTION ID** must be sent to the customer who purchased a subscription for your workbook. This token lets him activate the workbook.

Some free WooCommerce plug-ins allow sending emails to customers after purchase.

For instance, Woo Custom Emails available at https://wordpress.org/plugins/woo-custom-emails-per-product/

To insert the the activation token in the email template, you could use:

<div align="center">

`{order_billing_email}-{subscription_ID}`

</div>

<div align="center">

[Using WooCommerce custom fields (subscription)](#)

</div>

10.5. Manage existing activations / block customers (subscriptions)

10.5.1. Using WooCommerce custom fields (subscription)

For a given subscription, you can manage activations by editing its parent order: in the Related Orders section, the Parent Order is indicated:

Click the **Order Number** to manage activations of a given subscription.

When the customer activates the compiled workbook EXE, the web application modifies the corresponding WooCommerce order by adding a custom field named **xlspadlock_activations_{productid}:**

Custom Fields

Name	Value
is_vat_exempt	no
Delete Update	
░░░░░░░░	░░░
xlspadlock_activations_13	8F░░░░░░
Delete Update	

The value of the **xlspadlock_activations_XX** custom field contains a **list of system IDs**. These unique system IDs are the ones of the computers the customer activated the compiled workbook EXE on. System IDs are separated by the | character.

For instance, "CD78-4563-3F2B|HC78-1243-EF74" indicates that the compiled workbook EXE was activated on two different computers. If you configured the web application to allow 2 activations only, next time an error will be triggered (no more activations left).

You can manually clear existing activations by removing the corresponding system ID.

Note: each time the customer activates the compiled workbook EXE file, a note is added to the WooCommerce order, as you can see here:

[Blocking a customer (subscription)](#)

10.5.2. Blocking a customer (subscription)

You can block activations if your customer for instance asked for a refund.

The WooCommerce Kit automatically blocks any activation if:

- The subscription status is different than "Active": activation and validation will be refused.

- If the "xlspadlock_blocked" custom field is set to "1".

This can be done by adding a custom field to the subscription's parent order with the following properties:

- Name: xlspadlock_blocked

- Value: 1

Custom Fields ▲

Name	Value
xlspadlock_activations Delete Update	CD78-4563-3F2B|HC78-1243-EF74
xlspadlock_blocked Delete Update	1

Add New Custom Field:

Name	Value
— Select — ⌄ Enter new Add Custom Field	

Custom fields can be used to add extra metadata to a post that you can use in your theme.

Downloadable product permissions ❓ ▲

#68 — Markbook Demo — Markbook-Demo-Activation-32.exe: Markbook-Demo-Activation-32.exe — Downloaded 0 times

[Revoke access]

To unblock the order, set the value to 0 or simply remove the custom field "xlspadlock_blocked".

Description of the validation process (subscriptions)

10.6. Description of the validation process (subscriptions)

The WooCommerce Integration Kit supports the validation feature of XLS Padlock: you can configure your compiled workbook EXE to **check the validity of the customer's subscription regularly**.

> 💡 **This feature is only available if online activation is also used**.

Possible uses:

- you sell Excel workbooks and you want to block access to them if end users ask for refunds.

- you want to control who can access Excel workbooks regularly.

To activate the validation feature, fill in the base validation URL field.
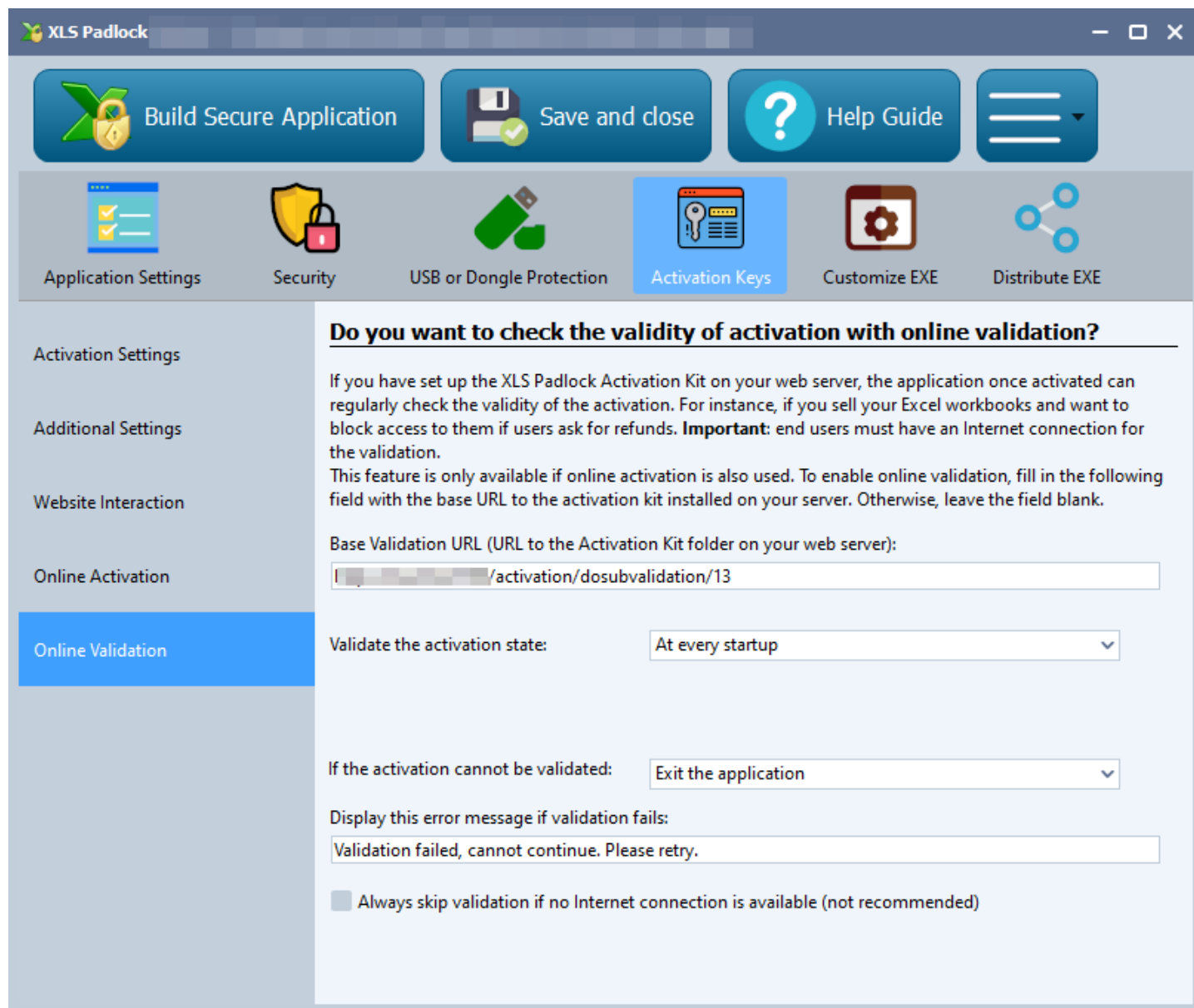
In the **Base Validation URL** field, enter the base URL as defined previously (see Base URL for activation) on your web server **followed by /dosubvalidation/{product ID}** (a final slash / is not mandatory) and replace **{product ID}** with the product ID associated to your workbook (and defined in workbooks.json).

In our case, it would be:

```
https://www.mywebsite.com/wordpress/activation/dosubvalidation/68
```

Then, define the frequency of validation and what should happen if the validation fails. Please refer to the XLS Padlock User Guide for further details about each setting.

Recompile your workbook EXE with XLS Padlock.

The compiled workbook EXE can now perform validation at startup:
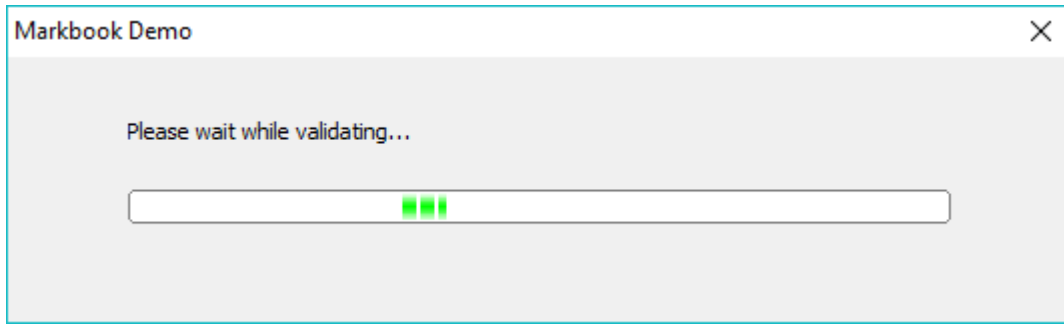
Suppose that you blocked the access of your customer (see Blocking a customer) or the subscription expired, they will get this error (returned by the web application):
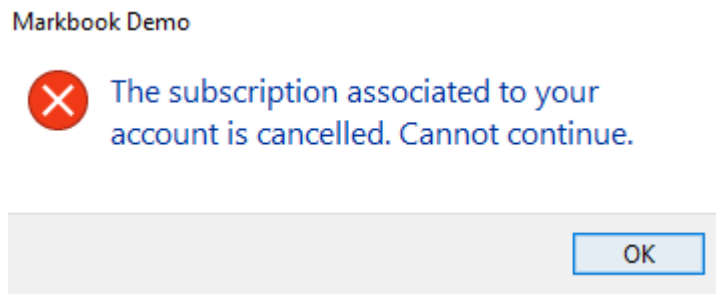
Otherwise, if validation is successful, the compiled workbook EXE continues its execution.

During validation, the WooCommerce kit also checks the subscription's status.

If the **subscription status** is different than "Active", validation will be refused.
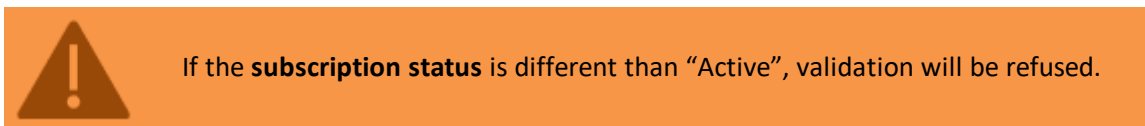
This ensures that only paying customers with active subscriptions are validated, and therefore, may access the workbook.

Upgrading from a previous release

11. Upgrading from a previous release

To upgrade to a new version of the WooCommerce Integration Kit, please follow these recommendations. This supposes you did not make modifications to the core PHP files in the kit.

1. Make a backup of your existing "**xlspadlock-woocom**" folder (see 5.1) on your server.

2. Use your FTP client to download a local copy of the validationkey.txt file normally available at:

   `/activation/inc/lib/validationkey.txt`
   (see 5.3).

3. Use your FTP client to download a local copy of the config.ini file normally available at:

   `/activation/inc/config.ini`
   ([see here](#))

   Use a text compare software or any text editor to compare changes between the new config.ini file available in the updated kit, and your own local copy. In fact, some new parameters may have been added or removed to/from the new config.ini file and you may want to customize them.

4. Use your FTP client to remove the entire "**xlspadlock-woocom**" folder on your server. Then, upload updated files from the new "**xlspadlock-woocom**" local folder as explained in [Copy the files to your server](#).

5. Upload the local validationkey.txt file back to its previous location on your server.

6. Upload the local config.ini file back to its previous location on your server.