Gipter UI description

version: 4.+ for Java 8 & 11+ distribution

author: Paweł Gawędzki

Don't you worry!
My minions will take care of your copyright items!

Table of Contents

Important

This is no an instruction how to set up the Gipter!!! It is a description of UI. Instruction can be found on the github.

Requirements

You need Java 8 installed on your computer. You can download it here.

Gipter is also available for Java 11+. There is a separate distribution for that.

Launching in UI mode

UI mode is set as default. Just double click on the Gipter.jar and that's it. If you have any problems then use the right click option and pick '*Open with*' then choose '*Java(TM) Platform SE binary*'.

Launching in command line (CLI) mode

Open PowerShell, go to Gipter home and then use this command:

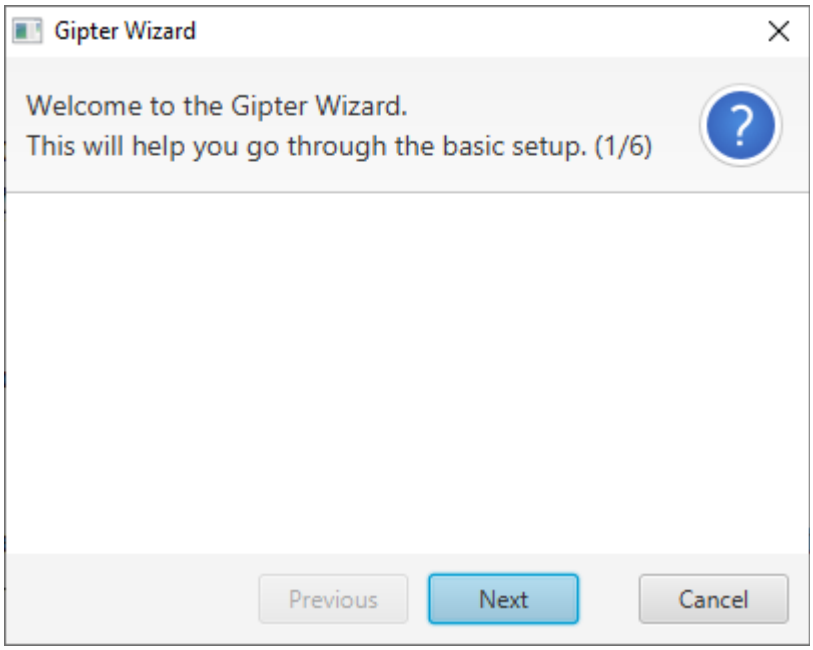
```
java -jar Gipter.jar useUI=N
```

or create a file with extension *.cmd and copy-paste the above instruction. Place the file in the Gipter home directory and double-click it. You will launch the command window in which Gipter will be executed.

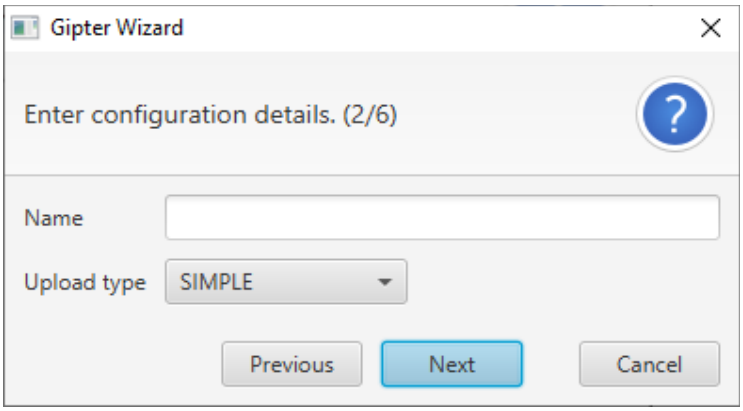
Wizard

If you executing the application for the first time you will find short wizard that will help you got through the basic configuration. Depend on which options you pick the number of screens could be varies. Here you can see all the screens you can encounter in order:

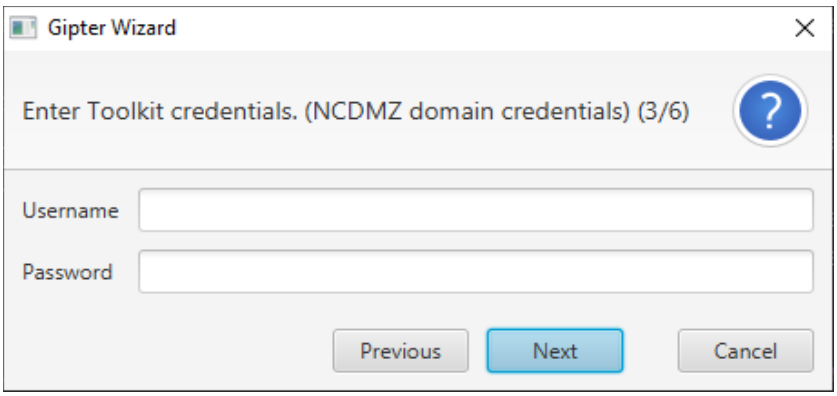
Welcome screen – with the surprise that you need to discover by your self ;)



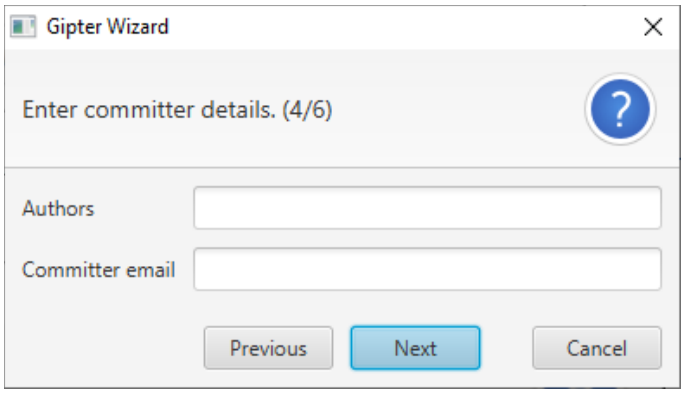
Configuration details screen – this is screen number 2 whre you need to enter configuration name and upload type. If you do not enter the name then default value for it will be set. Default value is *'wizard-config'.*



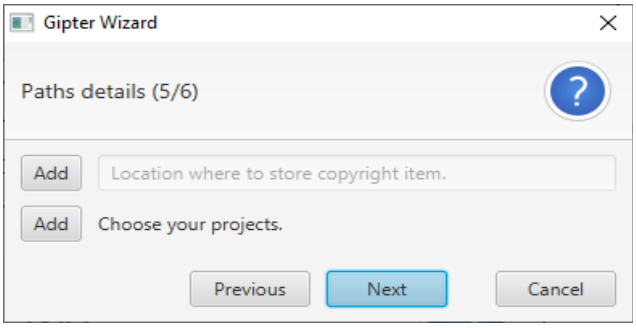
Toolkit credentials screen – screen number 3 where you need to enter you credentials to Toolkit. Just remember it is NCDMZ domain login and password. By default the username is going to be taken from system settings.



Author screen – screen number 4 where you need to enter author of the item. In simple words if you are using git repository then author is git *user.name* from git config and committer email is *user.email* from git config.



Path details screen – screen number 5 where you need to choose project, from which you will generate you copyright item and choose the location where generate copyright item will be stored on your computer.

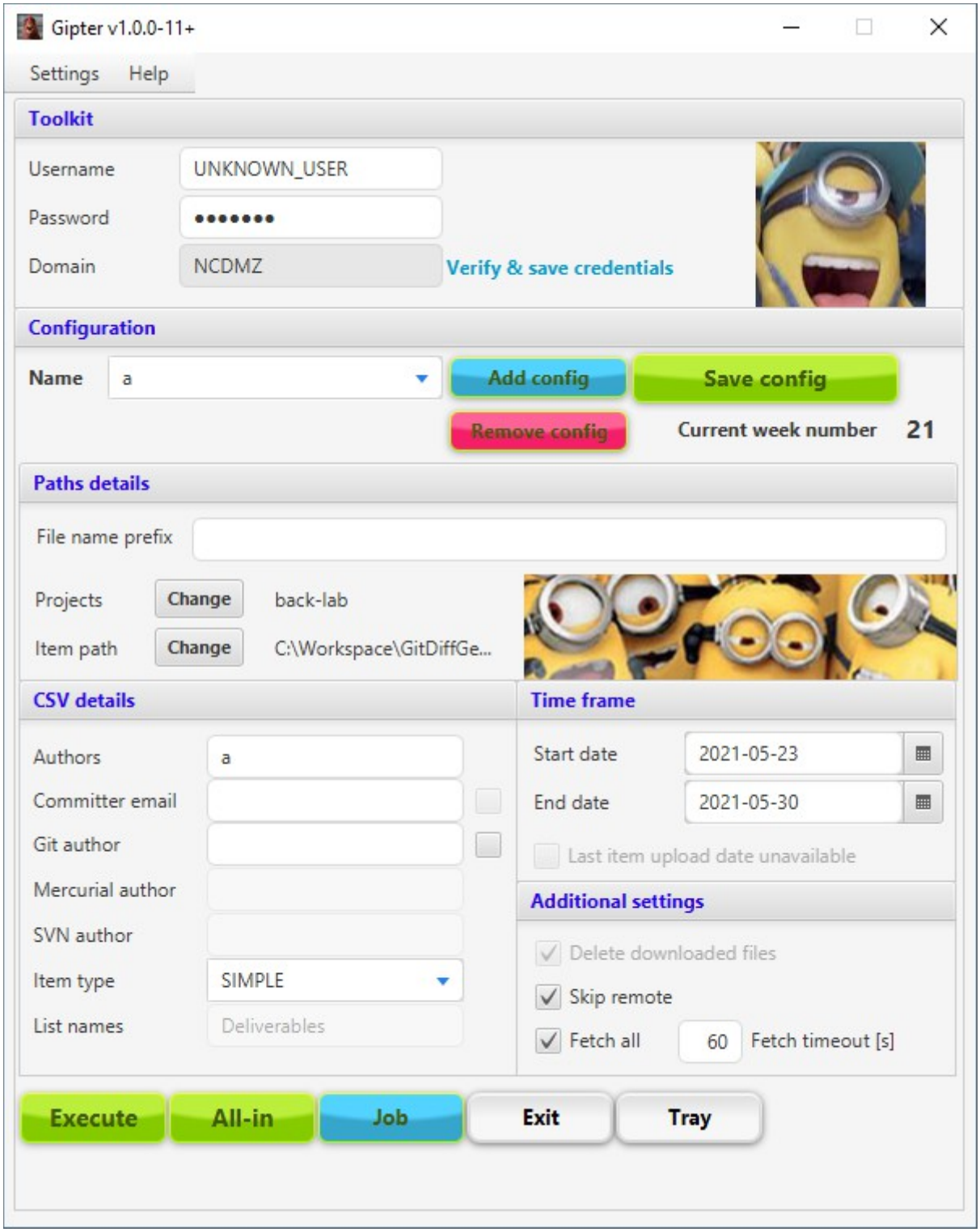


The last screen is final where you will find nothing more but Finish button.

As I mentioned at the beginning of this paragraph the wizard is launched by default when you execute the application for the first time, but also the wizard can be launched from *Help* menu manually. You will find the description of that further in this instruction.

Description of main window

All parameters are described here.



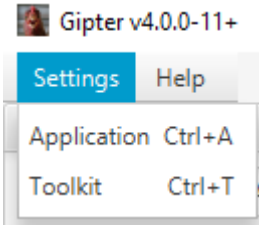
This is the reflection of all parameters that are defined in the Gipter application. You should know how to set them. What's important here is toolkit credentials. Use NCDMZ credentials. Once you enter the credentials you can verify them by pressing the link **Verify & save credentials**.

Menu

On main window you have the access to menu. There are two different menus: Settings and Help.

Settings

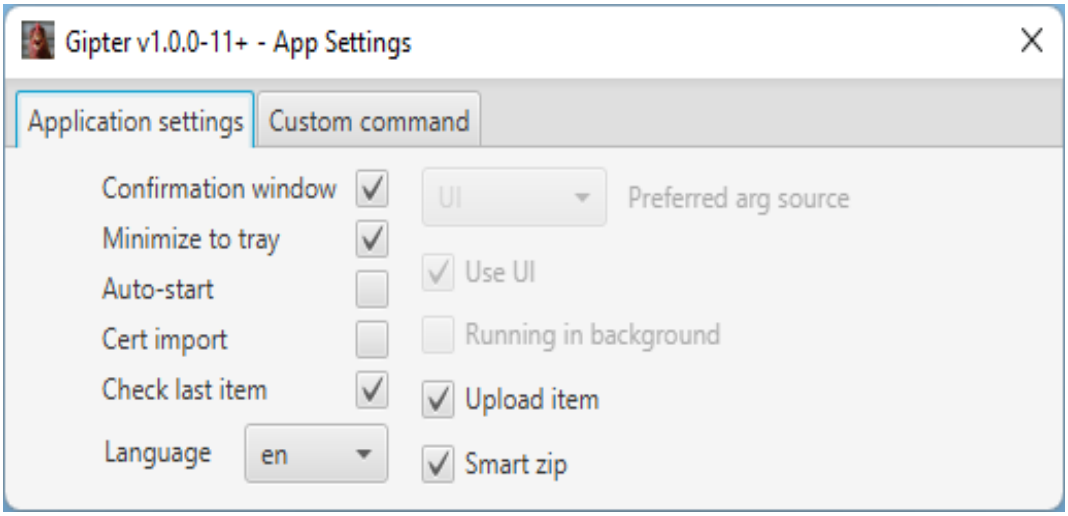
Under 'Settings' you have two different type of menu items:



All of them have the accelerators set.

Settings → Application

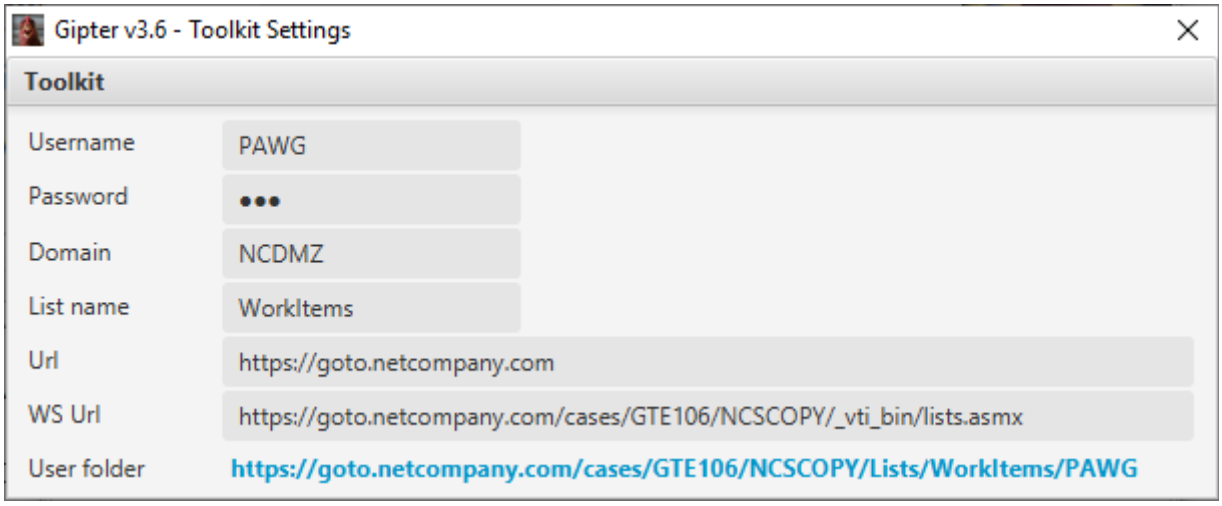
displays window where you can change the behaviour of the application:



You can manipulate these settings as you want. There is an accelerator set for this window. If you press *ESCAPE* then window will be closed and settings saved.

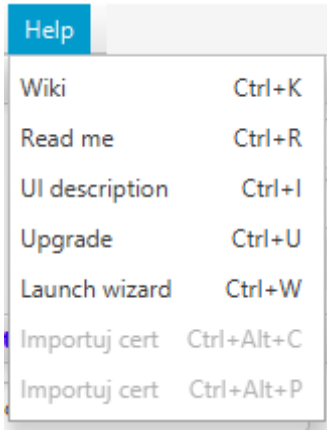
Settings → Toolkit

displays window with toolkit settings. At this window you can not set anything. What you can do here is you can go to your copyright items in toolkit. To do so just press blue link.



Help

The second menu item is '**Help**'. I can imagine that you may have some problems with the application so here you have the place where you can find some answers.



Wiki

Takes you to the wiki page on the Github.

Read me

Takes you to the page on the Github, where you can find description of all parameters.

UI description

Opens this document, if it is located in Gipter's home directory.

Upgrade

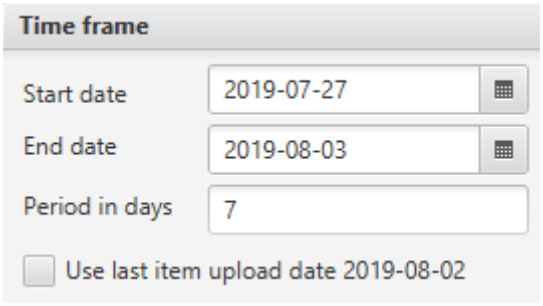
This option is available when new version of the application is available. When you press it the automatic upgrade will start.

Launch wizard

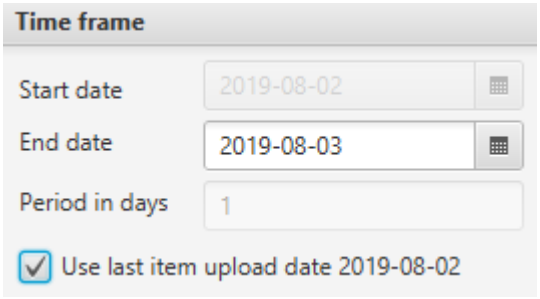
This option launches the wizard. So you can add your configuration straight from main window or by using wizard.

Checkbox on main window

There is one very useful checkbox in 'Time frame' section.



The last item upload date is the date taken from Toolkit. This date is the date of the last copyright item uploaded to Toolkit. This date is downloaded automatically on start-up and each refresh of main window. Of course this date will be downloaded if you properly set your Toolkit credentials. Why do you even want to use this option? If you have the last item upload date, then you can select the checkbox, save the configuration and execute new upload. Start date of the new upload will be the last item upload date. See below picture:



So if you do not use built-in jobs, but execute uploads manually, then this option allows you to generate the copyright items from your last upload until now.

Buttons on main window

Here is a short description of the buttons.

Add config – adds new configuration. You are allowed to add as many configurations as you want. To use that button you need to set configuration name first.

Remove config– removes selected configuration. When selected configuration is removed then next from the list is taken. If there is no configuration left then default values are displayed.

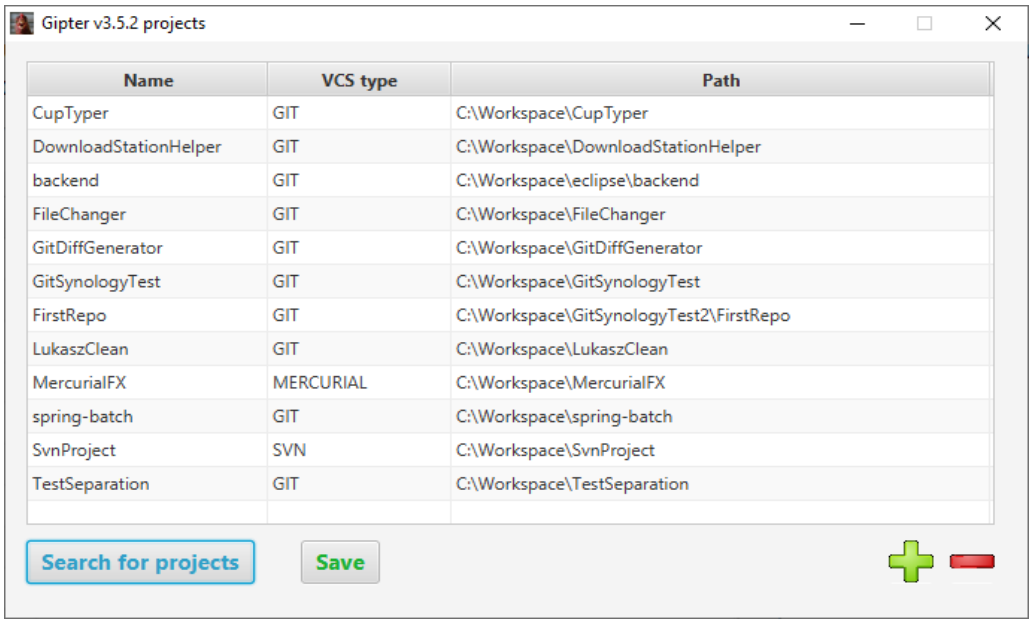
Save current configuration – saving or overriding current settings under the given configuration name.

Execute – executes diff generation and upload to toolkit for chosen configuration name.

All-in – executes diff generation for all configurations**.**

Tray – Closes the main window and create Gipter icon in system tray. This button is enabled when tray is supported by operating system.

Change – for '*Project paths*' allows you to pick up all projects, that you want to combine and use as copyright item. You can either replace previously chosen or add a new one. To do so Gipter will open new window to manage the projects:

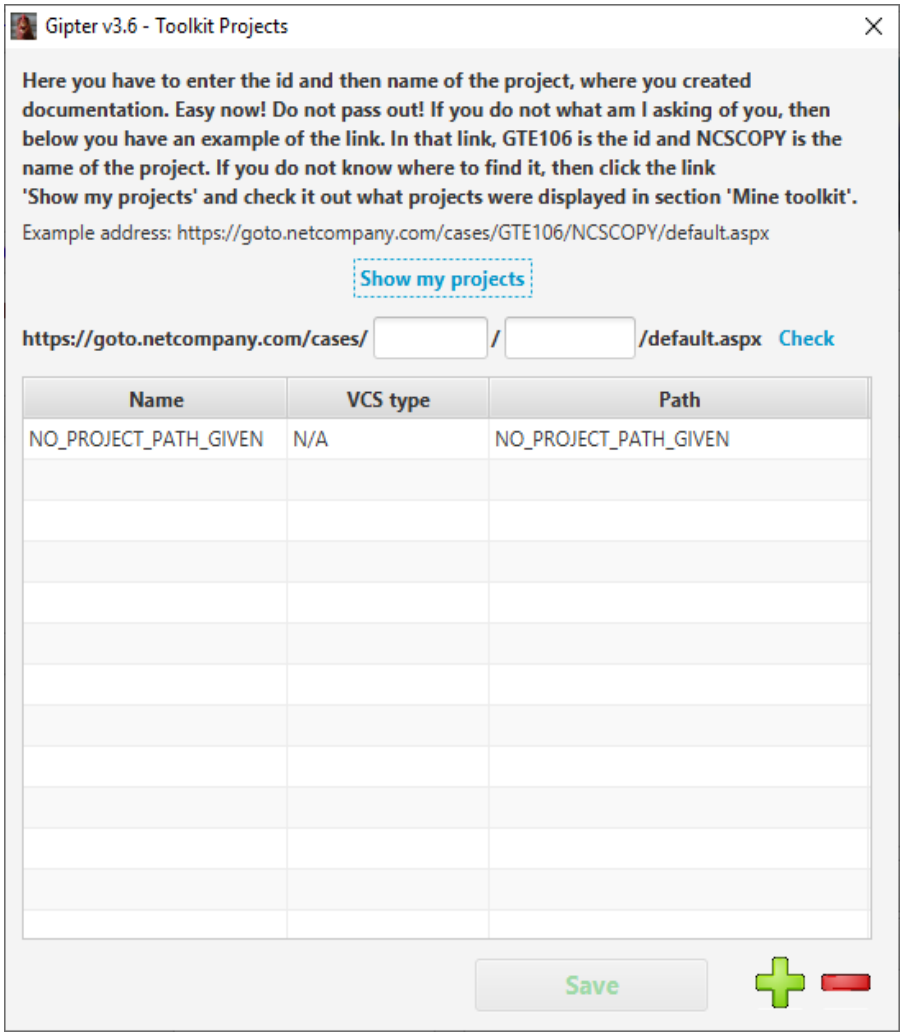


Interface is simple. If you want to add new project then press , if you want to remove then select project(s) and press . If you want to Gipter find all your repositories because manual adding is too boring then use '**Search for projects**' button and select parent folder with you projects. You can use this feature multiple times. It will add new projects to existing. To save changes you made just press button '**Save**'.

If you set upload type as TOOKLIT_DOCS then different window with projects will be displayed. And why would you do that? Because Gipter produces diff from the documents that you created. So for the documents you need to specify different kind of projects. To be more specific you need to add Toolkit projects. Under these projects your documentation is kept. But don't

you worry. If you set toolkit credentials at the main window of application, then Gipter will download your project automatically for you and display it in the table. You can always check if Gipter downloaded proper projects, by filling in two fields in the link '**Check**' and press that link. It should redirect you to toolkit. There is another link there 'Show my project' that will redirect you to the page with all projects that are available for you. That window contains also the description what to do :)

The way to add and remove projects from the table is exactly the same as with the regular projects.



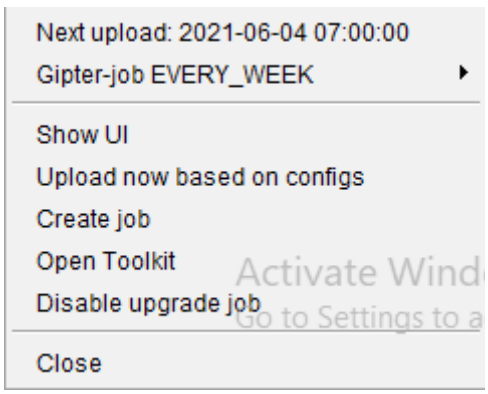
Change – for '*Item path*' allows you to choose where to store your copyright items.

Exit – terminates the program.

You can also change the language to Polish.

Tray description

When you look at the tray you will see a new icon there like below:



Yes, yes! It’s a chicken ;) When you right-click that chicken in your tray then you will see the menu. Below is a short description of all available options.

If the program have been executed at least once, then **Last upload … -** shows when was the last upload of item and status of it (success or fail). Following options are visible all the time.

Show UI – brings back the main window with application parameters.

Upload now based on configs – uploads copyright items for all configurations. This actions is equal to **All-in** button from the main view.

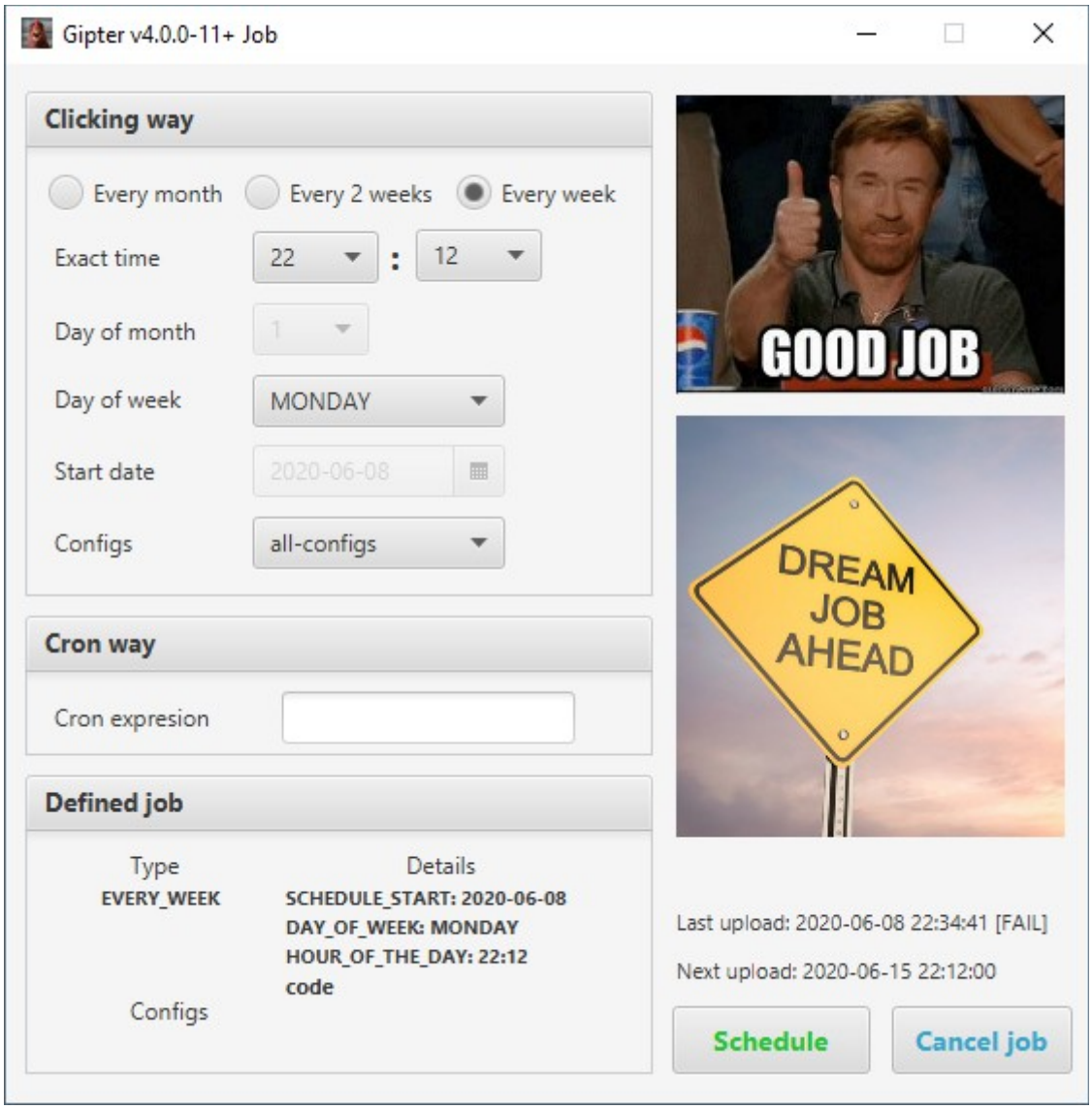
Open Toolkit – this menu item is available when toolkit credentials are set. After click on it it will take the user to the Toolkit where the application uploads the copyright items.

Disable upgrade job – switching off the job responsible for checking if the new version of application has been released. When this functionality is disabled, it can be enabled from the same place.

Close – terminates the program.

When you create a job, then **Next upload …** - shows when will be next execution of the job, which means when next upload will be triggered.

Create job – launches the window where you can setup the gipter-job and schedule it. Below is a screenshot with that window:



You can define gipter-job both ways: by clicking in predefined values or specifying CRON expression. CRON expression is well known but if you want to know more read this. If you need with building the CRON expression go here.

If job was defined in the past, its details will be displayed in 'Defined job' section. Remember that when you define job you need to choose the configurations that will be used in that job. By default all configurations are chosen. There are two ways of deletion an existing job. One is simple cancelling by pressing button 'Cancel job'. Job will be forgotten. The second is defining new job by pressing 'Schedule'. The old one will be replaced by the new one.

If from some reason the job was not executed (for instance your computer was not working), then once you run the application you will be asked what to do with missing job. You can execute it or not it's up to you.

If you choose to the clicking way to define the job then:

Configs – configuration to choose.

Exact time – is an hour of the day when the job must be executed.

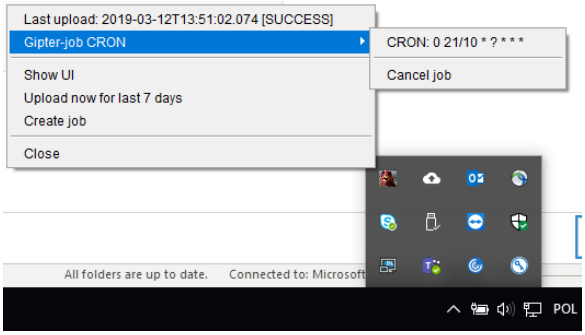
Day of the month – the exact day of the month to execute the job.

Day of week – the day of the week when to execute the job.

Start date – when to schedule the job. At this date above job definition will start to be valid.

Radio buttons are self-explanatory (I think).

Schedule – will create gipter-job and put it into quartz scheduler. Once the job is created you will see the difference in the tray:



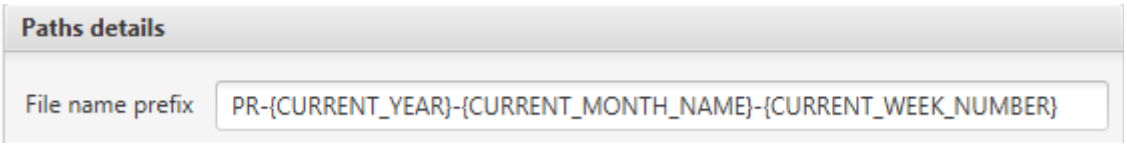
In the tray area now you will see the menu with gipter-job type and details. Also, you can **Cancel job** that job. If you do that this section will disappear from the tray.

You might notice that there is one additional entry on the very top of the tray. This is the info when the last upload was executed end if it was success or failure.

The important thing here is that this job will work only **when Gipter is working**. If you close the application, then the job will be killed. Furthermore if you run application again, then Gipter will reschedule killed job, so it could run again.

Custom item name

You can use some predefine values when you are defining the file name prefix.



When you use predefined values, then these values will be replaced by the real values. In above example *{CURRENT_YEAR}, {CURRENT_MONTH_NAME}, {CURRENT_WEEK_NUMBER}* will be replaced by the real values. When you define file name prefix like that, your copyright item and the name of the file with your work will have the following name:

Lets say that today is 2019-11-25 then the generated name will be: **PR-2019-AUGUST-48**.

You may see that there are values written with capitol letters and non-capitol letters. It means that if you want to have lets say start date month name written with non-capitol letters then you must choose value following value {start_date_month_name}. If you want to have the same value but written with capitol letters then choose {START_DATE_MONTH_NAME}.

Above text field contains smart autocompletion so you will have no problem to use predefined values. Autocompletion is case sensitive.

Session cache

There is one other functionality that is worth attention. Application has session cache available only on the main menu window. The scenario behind that is as follows. Whenever you want to change a bit existing configuration/s without saving it and execute that configuration/s the cache mechanism allows you that. You can change the configuration and do not save it, then switch between configurations and all your changes will be kept in the application cache. If you close the application without saving your changes first, then your changes will be lost, because the cache is terminated along with the application.

Accelerators

Here is the list of all accelerators in the application:

Menu shortcuts:

CTRL + A – open application settings window

CTRL + T – open toolkit settings window,

CTRL + U – upgrade the application,

CTRL + W – open wizard,

Main window shortcuts:

CTRL + ENTER – execute current configuration,

CTRL + SHIFT + ENTER – execute all configurations,

CTRL + J – open job window,

CTRL + ESCAPE – exit the application,

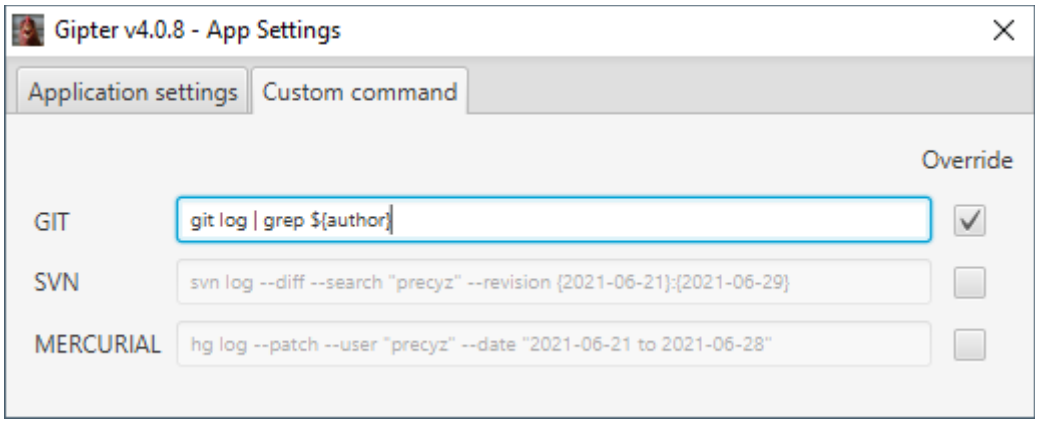
CTRL + M – hide main window,

CTRL + S – save configuration,

Custom commands

You are allowed to use your own command that will produce diff from the repository. To do so go to Settings → Application and choose tab *Custom command.* The screen with default settings looks like the following:

To set your own command select proper checkbox and enter your command. The following picture presents how it can be done:

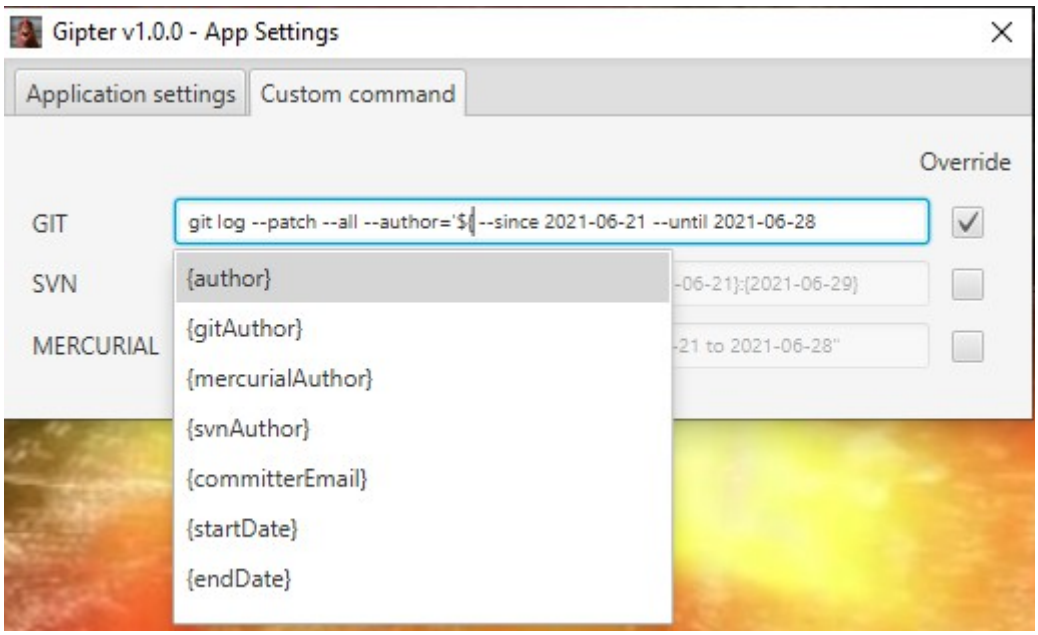


Settings are automatically saved on window close. Above example presents the following command:

git log | grep ${author}

That command is going to be used during item generation from the git repository. For other repositories you need to do the same steps.

One useful hint. You can use placeholders as long as they are exactly the same as the parameters supported by Gipter. You are allowed to use following placeholders: *author, gitAuthor, mercurialAuthor, svnAuthor, committerEmail, startDate, endDate* as long as you surround the placeholder with *${* at the beginning and *}* at the end. The text fields supports autocompletion, it means that when you enter *{* then available option will be displayed to autocomplete. Here is the example visualizing that cool feature:



When predefined value is used, it will be replaced by the actual value on-the-fly. If the predefined value is not available, meaning the application can not find it among the run configuration settings then the custom command will not work properly. Do not worry about *startDate* and *endDate* but if you want to use all others make sure that they are available.

Last item check job

This job is responsible for checking if toolkit contains item that has been uploaded in the current month. If not then popup window with adequate message is shown. Job is executed with following frequency:

```
"checkLastItemJobCronExpression": "0 5 12 20 * ?"
```

Every 20th day of each month at 12:05. The frequency can be changed directly in the *applicationProperties.json* file by simply replacing default cron expression with custom one. The value `"checkLastItemJobCronExpression"` for entry need to be updated.

That’s it! Enjoy (I hope)!