

## Important

This is not an instruction on how to set up the Gipter!!! It is a description of the UI. Instructions can be found on the [github](#).

## Requirements

You need Java 8 installed on your computer. You can download it [here](#).

## 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 use this command:

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

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

## Description of UI

All parameters are described [here](#).

The screenshot shows the Gipter v3.5.2 application window with the following configuration details:

CSV details	Launch details
Authors: NO_AUTHORS_GIVEN	Preferred arg source: UI
Committer email: [empty]	Confirmation window: <input checked="" type="checkbox"/>
Git author: [empty]	Use UI: <input checked="" type="checkbox"/>
Mercurial author: [empty]	Tray active: <input checked="" type="checkbox"/>
SVN author: [empty]	Auto-start: <input type="checkbox"/>
Skip remote: <input checked="" type="checkbox"/>	Save current configuration: <input type="button" value="Save"/>
Upload type: SIMPLE	

Time frame	
Start date	2019-04-01
End date	2019-04-27
Period in days	7

Toolkit details	
Username: PAWG	List names: Deliverables
Password: [masked]	Delete downloaded files: <input checked="" type="checkbox"/>
Domain: NCDMZ	Upload as html: <input type="checkbox"/>
List name: WorkItems	
Url: https://goto.netcompany.com	
WS Url: https://goto.netcompany.com/cases/GTE106/NCSCOPY/_vti_bin/lists.asmx	
User folder: https://goto.netcompany.com/cases/GTE106/NCSCOPY/Lists/WorkItems/PAWG	

Paths details	
File name prefix: PAWG_DOC	<input type="checkbox"/> Use prefix as file name
Project paths: /cases/GTE440/TOEDNLD	<input type="button" value="Change"/>
Item path: C:\Workspace\GitDiffGenerator\out\artifacts\Gipter	<input type="button" value="Change"/>

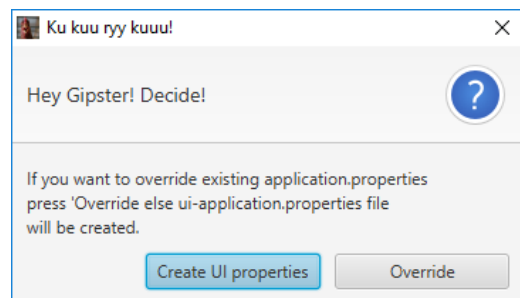
  

en  ☒ Finished with SUCCESS

Done

This is the reflection of all parameters that are defined in Gipter application. You should know how to set them. What's important here is toolkit credentials. Use NCDMZ credentials. Now here is a short description of buttons:

**Save** – saving current settings. You will decide if existing '*application.properties*' should be overridden or '*ui-application.properties*' should be created. To decide you will see below window



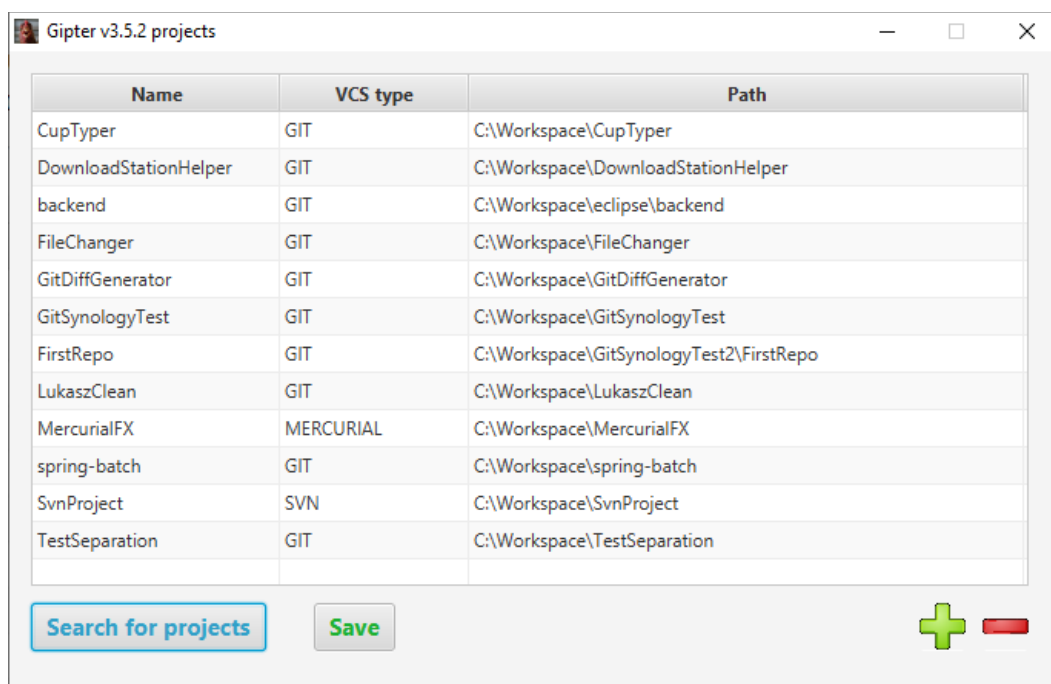
Why? Because *application.properties* is a source of truth for CLI mode. And from some reason maybe you would like to distinguish parameters of UI mode from CLI mode.



**Execute** – executes diff generation and upload to toolkit with parameters set in UI.

**Daemon** – minimize application to tray and saves current UI parameters to *ui-application.properties*.

If you choose '*No*' nothing will happen. If '*Yes*' then window with job details will be displayed, but this is described later.

**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**'.

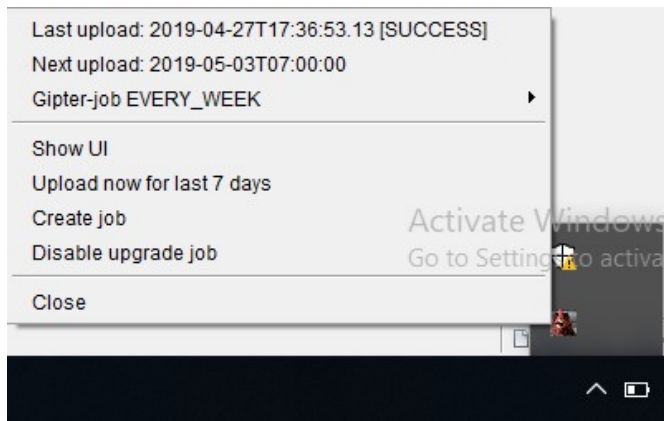
**Change** – for '*Item path*' allows you to choose where to store your copyright items.

**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 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).

Below options are visible all the time.

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

**Upload now for the last {number-of-days} days**—uploads copyright item for the last amount of days. Number-of-days is equal to '*period-in-days*' from the main window. This value can be updated dynamically. You just need to change the value of period-in-days on the main window and either save current configuration to ui-application properties or press Deamon to minimize the main window to the tray icon.

**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:

A screenshot of the 'Gipter v3.5.2 job' configuration window. The window is divided into two main sections: 'Clicking way' and 'Cron way'. The 'Clicking way' section has fields for 'Exact time' (7:0), 'Day of month' (1), 'Day of week' (FRIDAY), and 'Start date' (2019-04-27). It also has radio buttons for 'Every month', 'Every 2 weeks', and 'Every week' (selected). The 'Cron way' section has a 'Cron expresion' field. Below these is a 'Defined job' section showing 'Type' as 'EVERY\_WEEK' and 'Details' as 'SCHEDULE\_START: 2019-04-27', 'DAY\_OF\_WEEK: FRIDAY', and 'HOUR\_OF\_THE\_DAY: 7:00'. At the bottom, there are buttons for 'Schedule' and 'Cancel job', and status text for 'Last upload: 2019-04-27T17:36:53.13 [SUCCESS]' and 'Next upload: 2019-05-03T07:00:00'.

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. There are two ways of deletion an existing job. One is simple cancelling by pressing button ‘Candle 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 you choose to the clicking way to define the job then:

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

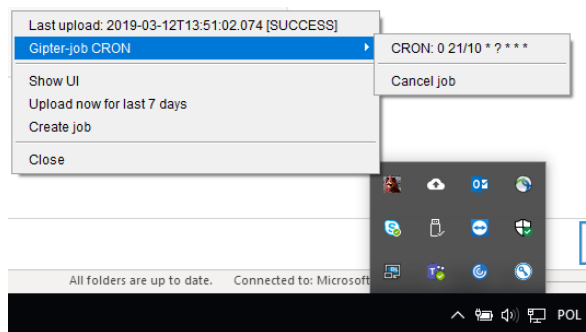
**Day of the month** – the exact day of the month to execute the job.

**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 the difference in the tray:



In 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.

That’s it! Enjoy (I hope)!