

# User manual (browser extension)

This page documents the usage of Haktilo WebExtension. For a documentation of Haktilo proxy's usage, see [here](#).

## Installation

Instructions for different browsers have been put on their respective pages:

- [Installation instructions \(browser extension, Mozilla\)](#)
- [Installation instructions \(browser extension, Chromium\)](#)

Users who want to install Haktilo from source can also visit [Building the extension](#) page.

## Understanding the concepts

### Script blocking and injection

Haktilo combines features of a user script manager and a content blocker. Out of the box, it can be used to block site's JavaScript, similarly to how NoScript (for example) does it. Once you import custom scripts into Haktilo (either from a Hydrilla repository server or by typing code in a form in the settings page), it can also inject them into pages.

Script blocking and injection is configured using [URL patterns](#). Patterns have different specificity. More specific patterns will override the settings of the less specific ones. In short, for every visited page, Haktilo performs the following steps:

1. Check if the page is privileged (e.g. it is its own settings page or a directory listing of local filesystem). If it is, don't attempt to block nor inject anything on this page and don't perform the remaining steps.
2. Try to find a script blocking/allowing/injection rule with a pattern matching page's URL. If found, apply the rule and don't perform the next step.
  - If multiple rules match, pick the one with the most specific pattern.
3. Check whether the default policy is to block scripts or allow them. Act accordingly.

In the end, Haktilo's action shall be one of the following:

- block page's own scripts
- allow page's own scripts to execute (i.e. don't do anything)
- inject the supplied user script **and** block page's own scripts

We can see that Haktilo's concepts are different from those of most user script managers. GreaseMonkey, for instance, executes user scripts alongside page's original scripts. Haktilo effectively **replaces** page's scripts with the user-supplied ones.

## Mappings and Resources

To make handling JavaScript libraries more manageable, we introduced our own concept of packages. Currently, Haktilo understands 2 different types of items (i.e. packages):

- Resource - It defines a set of scripts that can be injected together into a page. It can also name other resources as its dependencies. When injecting scripts of a given resource into some page, Haktilo will first inject scripts of all resources depended on.
- Mapping - It associates URL patterns with resources in a 1:1 fashion. If pattern `https://example.com/**` is associated with resource `my-sample-res` it means the scripts from `my-sample-res` should be injected into all HTTPS pages under the `example.com` domain.

For simple cases, this may be overly complex. Because of that, Haktilo's settings page contains a simple form that can be used to quickly define a script payload for a set of URL patterns.

Mappings and resources can also be installed from Hydrilla repository which serves both simple and complex (i.e. multi-resource) payloads. Defining more complex mappings and resources within Haktilo itself is not (yet) supported.

## Operating the popup window

While browsing with Haktilo installed, a small Haktilo icon will be present in the extensions panel (usually located to the right of the

url bar). Clicking on it will open the extension's popup.

haketilo\_popup.png

At the very bottom of the popup there is a convenience shortcut - a button which, when clicked, opens Haketilo's [settings page](#). At the top is page's status report, containing information on how Haketilo has modified the currently viewed page. For the above website there is no custom policy set. As a result Haketilo simply applies the default policy of blocking scripts. In cases a policy is set for a matching [URL pattern](#), it is used to modify the page and its details can be viewed in the popup.

haketilo\_popup\_payload.png

Sometimes you might wish to look for custom scripts to use on sites or for ethical fixes others made to enable browsing without running nonfree JavaScript sites serve. Such search can be performed from the popup. Clicking the "Search for custom resources" button below status report will bring up the repository query view.

haketilo\_popup\_search\_for\_custom.png

Repository view contains a list of all installed repositories. Each entry has a "Show results" button which, when clicked, will make Haketilo query the repository for payloads that can be used on the viewed website. Haketilo comes with [https://hydrilla.koszko.org/api\\_v1/](https://hydrilla.koszko.org/api_v1/) configured as the default repository to use. This can of course be changed in the settings page.

haketilo\_popup\_show\_results.png

Currently, there are very few custom scripts listed in the main Hydrilla instance; if you are lucky enough to find a match, however, click the "More..." button on the right to install the policy.

haketilo\_popup\_more.png

In the view that appears you can see the components of the policy you want to install. You'll often see 2 entries with the same name. One represents a resource which defines a set of scripts. The other one represents a mapping which associates resource(s) with URL patterns. If, by chance, some components are already installed (for example because they were required by some other policy you installed earlier), they won't be listed. There is an exception - components with a newer version available. Those will be listed and proceeding with the installation will cause them to be updated.

When you're sure you want to commit the installation, just click "Install".

haketilo\_popup\_install.png

Voilà! The installed items (mapping and resource(s)) will be used the next time you load a matching page. They can also be viewed and managed under the appropriate tabs on the [settings page](#).

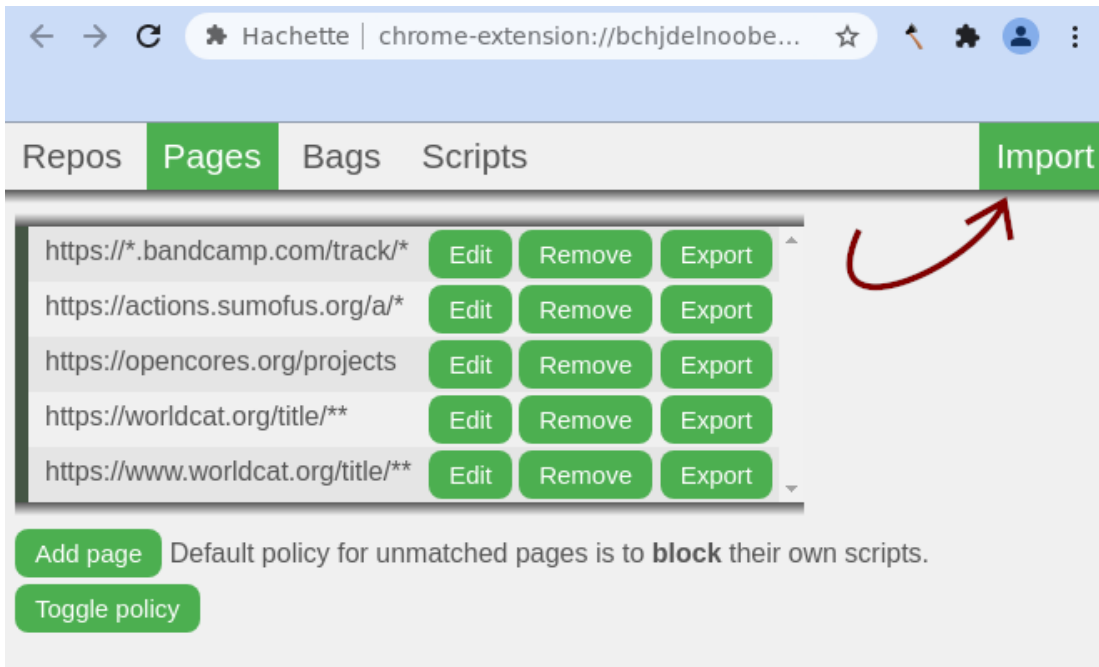
Above all else, Haketilo is designed to put you in control of your web browsing. Like Haketilo itself, all scripts distributed as defaults with the extension and published in the Hydrilla repository are free software. This means you can read, modify and distribute them without undue restrictions in either compiled form or as source code.

## Manually importing custom scripts

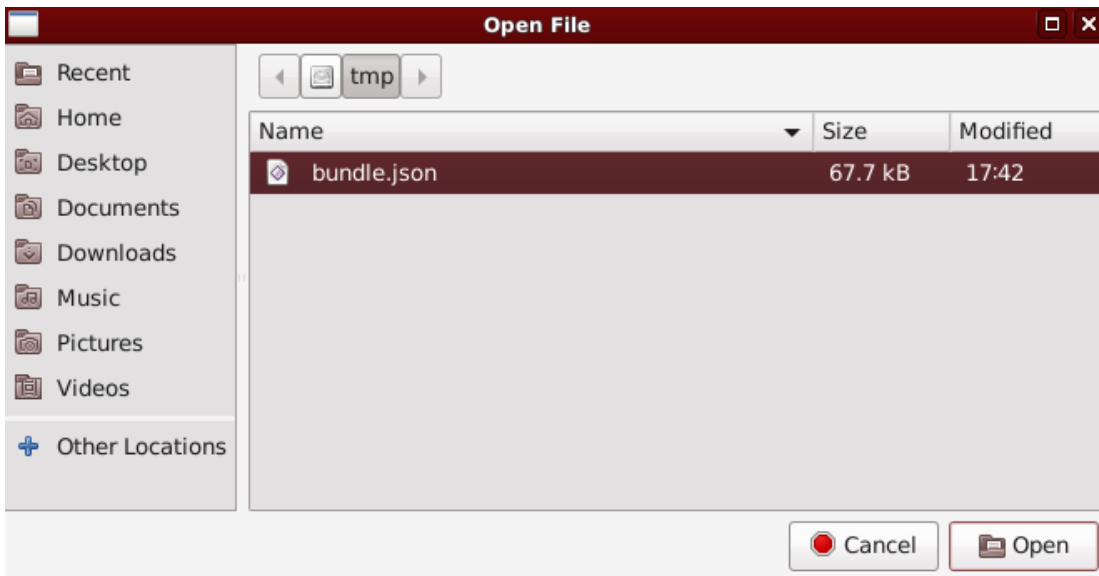
**Note: this section of the manual corresponds to old Haketilo version 0.1 and is awaiting an update**

Although installation of site fixes and custom content is meant to be convenient through the use of a repository, one can also export and import such payloads to and from JSON files. In fact, all the scripts currently served by the default Hydrilla repository can be downloaded in that format from <https://hachette-hydrilla.org/>. Most are fixes for js-encumbered websites, but there are also some alternative interfaces for already-functional sites. You can download and install particular scripts one by one or go crazy and just import it all at once from the "All-in-one bundle".

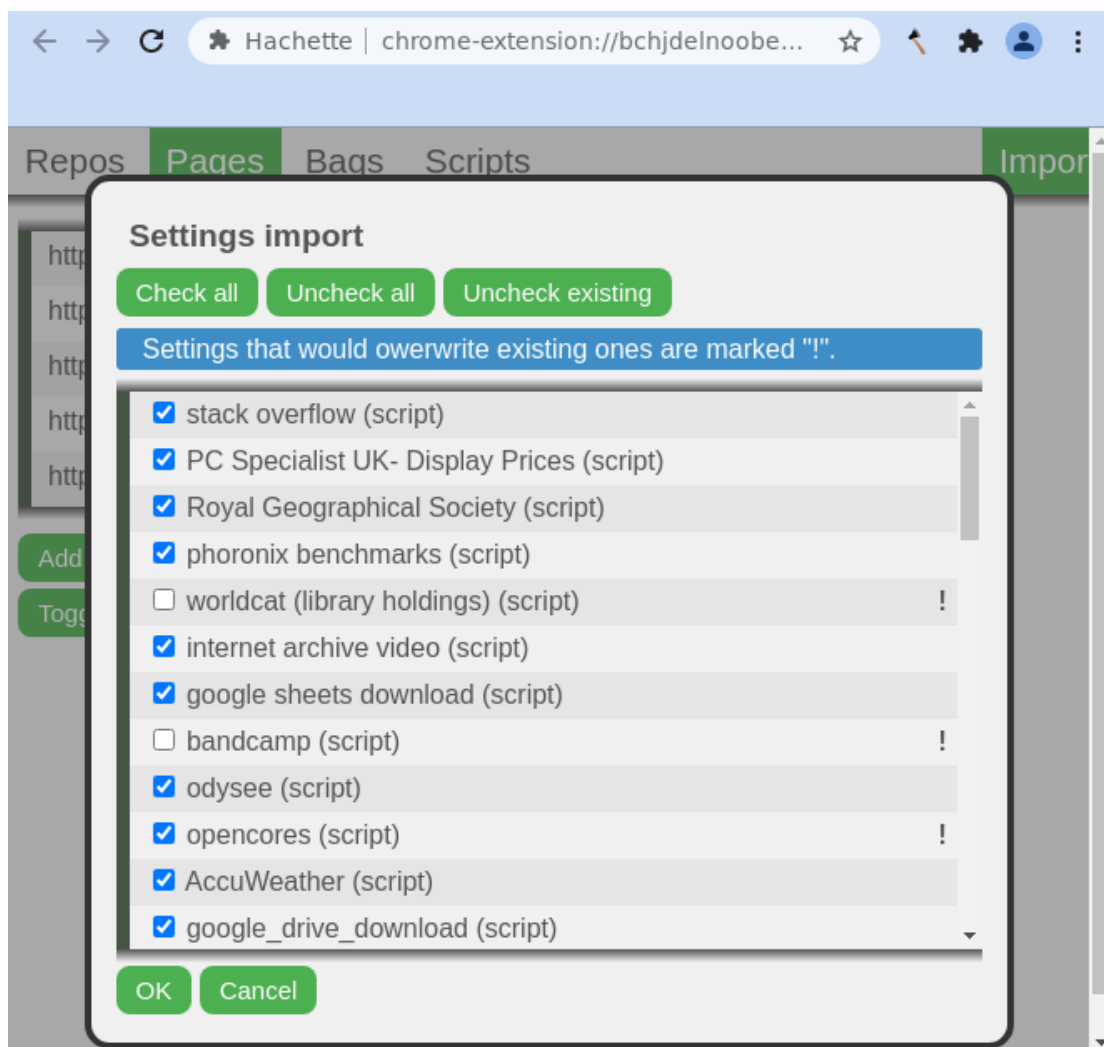
After you've downloaded the right .json file, go to Haketilo's settings page (reachable by clicking the button at the bottom of the popup window) and click "Import".



Now, find and choose the .json file with scripts.



In the frame that appears you can select which components you want to install. Note that components might behave improperly if you disable other ones they depend on. As of version 0.1, Haketilo will do nothing to stop you from messing things up ;)



If you click "Ok", scripts and settings are imported. Voilà! They can be viewed and managed under the appropriate tabs on the [settings page](#).

## Using the settings

**Note: this section of the manual corresponds to old Haketilo version 0.1 and is awaiting an update**

As well as allowing you to [manually import scripts from JSON files](#), the settings page provides access to all configurable options available in Haketilo. These are arranged under four tabs, one for each type of item- repository, page (pattern), bag, and script.

haketilo\_bags\_tab.png

Each tab is arranged similarly, with a list of items followed by an "Add ..." button (which opens an empty new item for editing). Besides every item is an edit button, a remove button, and- except for on repositories- an export button, which exports the item and its dependencies to JSON format for download.

Editing an item will open up a form with appropriate fields. **Changes are not autosaved**; they must be manually committed using the "save" button at the bottom of each form.

haktilo\_repo\_edit.png

Repositories and scripts currently employ relatively simple forms. However, to allow ample space for editing the source code of scripts, the form overflows the page, which requires scrolling down to find the "save" button and avoid losing changes.

haketilo\_script\_edit.png

Bags can contain scripts and other bags, and opening one to edit it will present alongside the name a list of items contained within. To add new items, click on the "Add scripts" button towards the bottom of the form, select which new items, and click "Ok".

haketilo\_bag\_edit.png

haketilo\_bag\_popup.png

Editing a page is slightly different to other items, even if the form works the same way. Instead of a name, pages have a field for a [URL pattern](#) that they apply to and a field for a payload to inject into them. This payload can be changed by clicking on the "Choose payload" button, which will bring up a popup similar to the one for bags except that only one item is selected at a time. Injecting a payload **will block all scripts that are sent with a page natively**.

haketilo\_page\_edit.png

haketilo\_page\_popup.png

As well as a list of all available scripts and bags, at the very bottom of the payloads popup there is the option "(None)" for no payload. When selected, this enables an additional checkbox in the editing form, which controls whether or not a the scripts sent with a page natively should be blocked. Script-blocking behavior on URLs without a corresponding page policy listed can also be set, at any time, using the "Toggle policy" button at the bottom of the pages tab.

haketilo\_edit\_script\_blocking.png

## What to do next

Please [REPORT BUGS](#). This is incredibly important. If nobody reports them, they likely won't get fixed.

You might want to learn about [current limitations](#) of Haketilo. If despite these you like the extension, please spread the word. We'll be also happy to receive some feedback or - if you're a programmer - code contributions. Consider [creating an account](#) on our issue tracker or writing to [koszko@koszko.org](mailto:koszko@koszko.org) :)

### Files

hachette_popup_injected_scripts.png	68.9 KB	09/10/2021	koszko
hachette_popup_install_scripts.png	67.5 KB	09/10/2021	koszko
hachette_popup_install_scripts_import_frame.png	72 KB	09/10/2021	koszko
hachette_popup_opencores.png	105 KB	09/10/2021	koszko
hachette_popup_possible_patterns.png	114 KB	09/10/2021	koszko
hachette_popup.png	54.5 KB	09/11/2021	koszko
hachette_settings_import_but.png	63.3 KB	09/11/2021	koszko
navigate_scripts_bundle.png	33.4 KB	09/11/2021	koszko
hachette_settings_import_frame.png	84.8 KB	09/11/2021	koszko
haketilo_bag_edit.png	39.2 KB	09/15/2021	jahoti
haketilo_bag_popup.png	57.8 KB	09/15/2021	jahoti
haketilo_bags_tab.png	30.4 KB	09/15/2021	jahoti
haketilo_bags_tab.png	30.4 KB	09/15/2021	jahoti
haketilo_edit_script_blocking.png	67.7 KB	09/15/2021	jahoti
haketilo_page_edit.png	70.5 KB	09/15/2021	jahoti
haketilo_page_popup.png	76.3 KB	09/15/2021	jahoti
haketilo_repo_edit.png	27.4 KB	09/15/2021	jahoti
haketilo_script_edit.png	62.8 KB	09/15/2021	jahoti
haketilo_popup_payload.png	83.4 KB	03/12/2022	koszko
haketilo_popup_search_for_custom.png	81.3 KB	03/12/2022	koszko
haketilo_popup_show_results.png	61.3 KB	03/12/2022	koszko
haketilo_popup_more.png	74.2 KB	03/12/2022	koszko
haketilo_popup_install.png	62.2 KB	03/12/2022	koszko
haketilo_popup.png	59.3 KB	03/12/2022	koszko