

3.1.1 Pages Module



Pages module is very advanced input module that can work with the following output modules: Archiver, WordPress and Sitemap manager.

Pages module concept:

Pages module can process the following types of premade content that you want to add to your website using our “drip” technology:

- **static pages** generated with the help of third party content generators with absolutely any extension such as .html, .htm, .php, .csv, .shtml, .txt, and you can even come up with your own extensions.
- **articles** in the form of html pages or txt files
- **any other static data** in the form of files that you think is useful to add to your website.

Pages module does not generate content! But it can work with premade content by “feeding” it slowly to your website!

First what you need to do is to create a folder on you server, lets call it “**contentpages**” and set its permissions to 777 (CHMOD 777).

Then upload your html pages or txt articles into it.

After that, in pages module specify a relative path to your content pages directory. If you created that folder in root directory it should be just **/contentpages** without closing slash.

File extensions for processing – here you can specify what files in that folder you would like to be processed by the script. Very convenient feature if you have for example txt, html, php files all in one folder in only want html files to be processed.

SGW[®] engi

Home Output Modules Input Modules Test & Log Modules Setup Options Logout [View site](#)

Pages RSS Wizard

Relative path to content pages folder ?

File extensions for processing (php, html, etc) ?

Scan all content in the folder: 10 files in base

Clear database: This will clear MySQL database, containing information about your content pages. 10 files in base

Number of hours to change file's date and timestamp: ?

The quantity of files for each processing: - ?

The method of adding content to your blog or main page: All in one post All in separate posts

Post title will be taken from the following tag ?

Text will be taken from the following tag ?

The number of symbols of blog post text: All text within tag Random number of symbols -

Allow HTML tags: ?

Insert links to pages into blog posts: No Before post text After post text

Html tag for links anchor text: ?

Links anchor text will be as: Only link text Link text + Additional phrase: Phrase before link text Phrase after link text

The links will be pointed to subdomain: ?

After you are done with extensions, click **“Scan”** button. SGW will scan your content folder and will add information about your files to the database.

In case you did something wrong and want to redo the process, you can clear the database by pressing the **“Clean”** button.

Number of hours to change file's date and timestamp: Let's say you select **5**. When the cron job runs, SGW will pick random time within 5 hours of current time and will change processing files date and timestamp to that time.

I know it sounds confusing, but it is very powerful feature. It will create an impression that current file was just created and uploaded to your server even though it was uploaded long time ago. This will help to eliminate footprints for the search engines and even make your machine generated content look like human content.

The quantity of files for each processing: Let's say you select from **1** to **10**. For each script execution by the cron job, it will be processing random number of files from that interval.

The method of adding content to your blog or main page: If you want Pages module to post snippets of text with links from your content pages to your blog, here you have two options: all content from currently processing pages into one single post or separate posts.

Post title will be taken from the following tag: Here you can specify where you would like the title for your blog post or newly generated page to be pulled from. You need to specify the tag, without brackets. **MG WAY CORP - All rights reserved ©**

your blog post or newly generated page to be pulled from. You need to specify the tag, without brackets.

For example: title

That means that SGW will look for `<title></title>` tag in your processed page and will make it a title of your blog post (or newly generated page, if you select archiver as your output module).

Text will be taken from the following tag: The same thing is with text. SGW will look for that tag in you processed page and will pull content within that text for your blog post or newly generated page in archiver. That way you can make post to your blog or have completely new pages generated in archiver, based on templates and snippets of text from your premade content pages.

The number of symbols of blog post text: Specifies how much text should be taken from your snippet:

- **all text** or

- **random number of symbols.**

Allow HTML tags: Specifies what html tags should be left in that snippet of text. Should be separated by |. All other tags will be automatically removed.

Next (blue) section (*see image above*) will be responsible for links.

Insert links to pages into blog posts: You need to specify where backlinks to your content pages will be inserted in your blog posts or newly generated pages:

- **before post text** or

- **after post text**

Html tag for links anchor text: Specify any tag where you would like you link's anchor text to be pulled from. If you put: **title**, then each link to your content page will have title of that page as its anchor text.

Links anchor text will be as: Here you can add an additional phrase to your link text if you need to. For example: *Please read more about...*

You can put it before actual link text or after.

The links will be pointed to subdomain: If this feature is needed, enter URL of your subddomain. (For example, your blog is located at your main domain, but your content pages are located at subdomain and you want your links to be pointed to 'http://subdomain.domain.com' instead of http://domain.com/subdomain). Enter without closing / slash. For example: http://subdomain.domain.com

Next (grey) section will allow you to specify **Output module** for Pages module:

CONNECTION WITH OTHER MODULES

Content from this module should be sent to: Wordpress module Archiver module

Connection of Module **Pages** with Module **Wordpress**

Choose Wordpress blog category: WordPress Module setup is needed before setting this section!

Blog user: WordPress Module setup is needed before setting this section!

Connection of Module **Pages** with Module **Pinger**

Ping remote services for this module with updates: Yes No

Click to update settings:

You can choose either **Wordpress module** to make posts to your blog or **Archiver module** to generate new pages on a random basis based on predefined templates.

If you choose WordPress module, you can also choose **WP category** to make posts to and **Blog user** to make posts from.

(Please, note that you need to specify settings for WordPress Module before using this feature.)

Ping remote services for this module with updates: Will activate pinger module to ping remote websites for all Pages module updates *(Please see Chapter 3.2.4. for more info.)*

Don't forget to click on "Update settings" after you do any changes.

Conclusion: So if you pair Pages Module (input) and WP Module (output), they will take random number of pages out of your content pages folder, change their timestamp, post snippets of text from these pages along with backlinks to your blog, updating your sitemap as well *(please see how to setup WordPress Module in Chapter 3.2.3).*

Search engines robots love blogs, so they will be able to index your content pages fast and very easy.

Then if you pair Pages Module (input) with Archiver Module (output), instead of posting to blog they will generate new pages based on set of predefined templates and snippets of text from your content pages, updating your sitemap as well *(please see how to setup Archiver Module in Chapter 3.2.2).*