Using Feeds

There are various times where it may be easier to perform adding content or updating content in bulk. For instance, if a resource page links out to another site for majority of the information (such as a list of your statutes), but the site changes – it may be easier to create a quick CSV file with the modifications, and upload this way. Maybe there is an RSS feed for news articles from another agency you'd like to have added to your site, or a twitter feed.

In all these cases the **Feeds** module is a wonderful module to perform and use these functions.

The Scenario

You're an agency which lists all your statutes and rules on a page for citizens as a resource, but you must keep the links and statutes updated in relating to your agency. Currently, you're hand coding these links in. Which was fine, up until the source website had been completely restructured and now all the links are different (but still follow a pattern). Now you must update by hand, but you have about 50 statutes where you need to find the URLs for and update the links on the statute page.

You have figured that if this has happened once, there's a good chance it will happen again. After all, technology improves daily, and design trends also change. Why not be a little proactive and make this easy in the future too?

In your brainstorm of possible solutions, you've settled on using a module called **Feeds**.

Prerequisites

Prior to using the **Feeds** module, you will need to ensure that the following has been completed:

- If planning to import new pages of information, or other information feeds - you've created a content type relating to the feed, with as many necessary fields needed.

- Any taxonomy vocabularies have been created if necessary.

- Create/Find an RSS feed or CSV file with the information needed.

For this scenario –
We've created one content type called **Statute.** We've named the *Title* field "Statute Number"; changed the *Body* field to "Regarding" as a name, and plain text input without a summary; created a *Link* field labeled as "URL", with no title option for the URL selected; created two

Taxonomy references, one for "Chapter" and one for "Article" (of which, two vocabularies were also created to link back to).

Our source is a CSV file where the headers consist of: title, regarding, article, chapter, url.

Required Modules

The following modules are required when using Feeds:

- Feeds (https://www.drupal.org/project/feeds)
- Feed Import (https://www.drupal.org/project/feed_import)
- Chaos Tools (required for Feeds) (https://www.drupal.org/project/ctools)
- Job Scheduler (required for Feeds) (https://www.drupal.org/project/job_scheduler)

Additional modules that could be useful for Feeds:

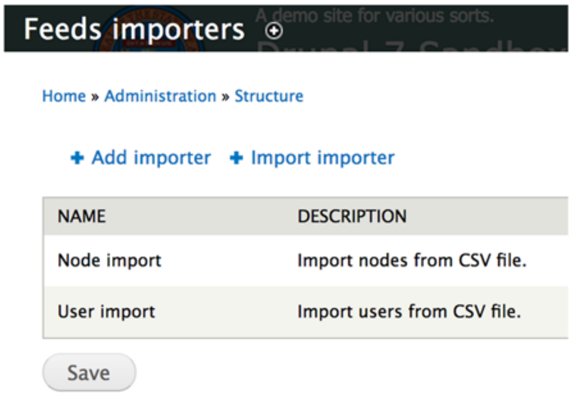
- Feeds XPath Parser (https://www.drupal.org/project/feeds_xpathparser)
- Feeds Tamper (https://www.drupal.org/project/feeds_tamper)

Configuring Feeds

1) Verify that any Content Types or Taxonomy items needed are created beforehand. *For this scenario, we have generated two Taxonomy vocabularies, one for Statute Article, and one for Statute Chapter. As well, we have generated a basic Content Type which consists of fields for: **Statute Number** (title), URL (link field), Regarding (plain text block), Chapter (referencing to taxonomy), Article (referencing to taxonomy).*

2) Enable the **Feeds**, **Feeds Admin UI**, **Feed Import**, **Chaos Tools**, and **Job Scheduler** modules from the Modules page of the website.

3) Navigate to **Structure > Feed Importers**. You may click Feed Importers from the menu and will see a settings page for adding a new feed importer; you may also use it's child menu if using an Admin Toolbar which allows showing of sub menus to select your next choice as well.

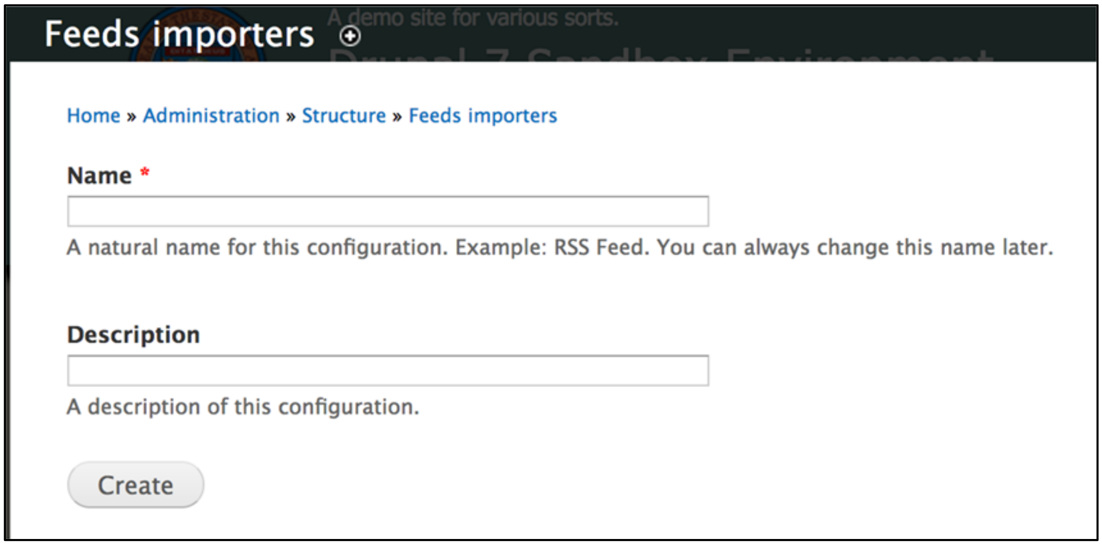

Feeds importers
A demo site for various sorts.
Home » Administration » Structure
✚ Add importer ✚ Import importer
Save

NAME	DESCRIPTION
Node import	Import nodes from CSV file.
User import	Import users from CSV file.

4) Click **Add importer**.

5) Provide a name of this feed in the **Name** field.

6) [OPTIONAL] Provide a general summary in **Description**, if necessary.


Feeds importers
A demo site for various sorts.
Home » Administration » Structure » Feeds importers
Create

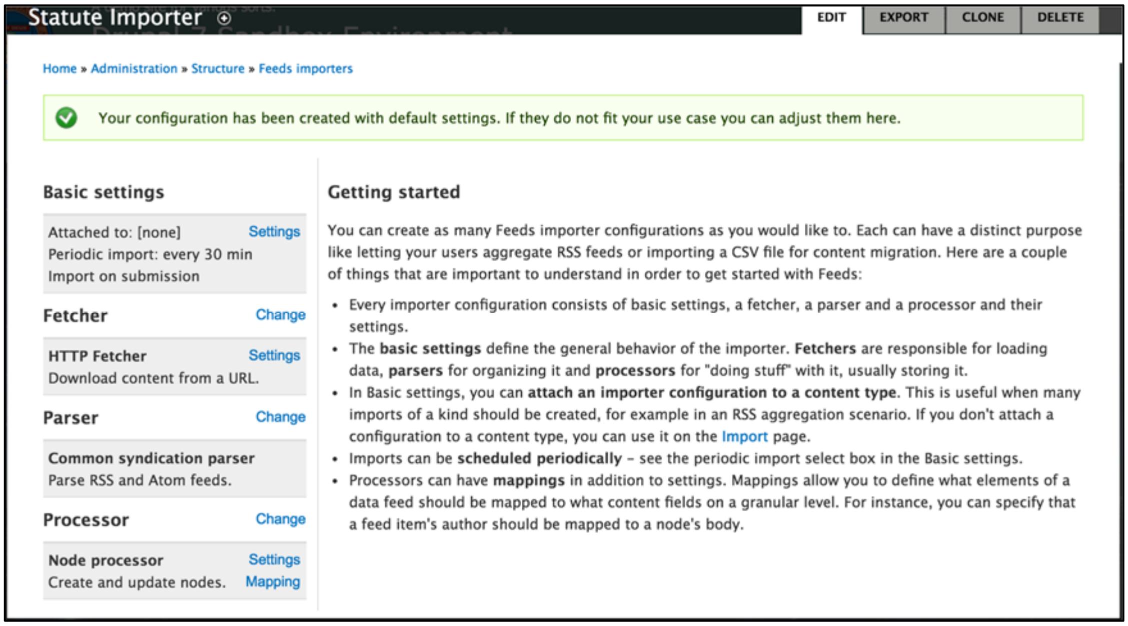
Name *

A natural name for this configuration. Example: RSS Feed. You can always change this name later.

Description

A description of this configuration.

7) Click **Create**; you will be provided a new window in which we are able to configure settings for.



Your configuration has been created with default settings. If they do not fit your use case you can adjust them here.

Basic settings

Attached to: [none] Settings
Periodic import: every 30 min
Import on submission

Fetcher Change

HTTP Fetcher Settings
Download content from a URL.

Parser Change

Common syndication parser
Parse RSS and Atom feeds.

Processor Change

Node processor Settings
Create and update nodes. Mapping

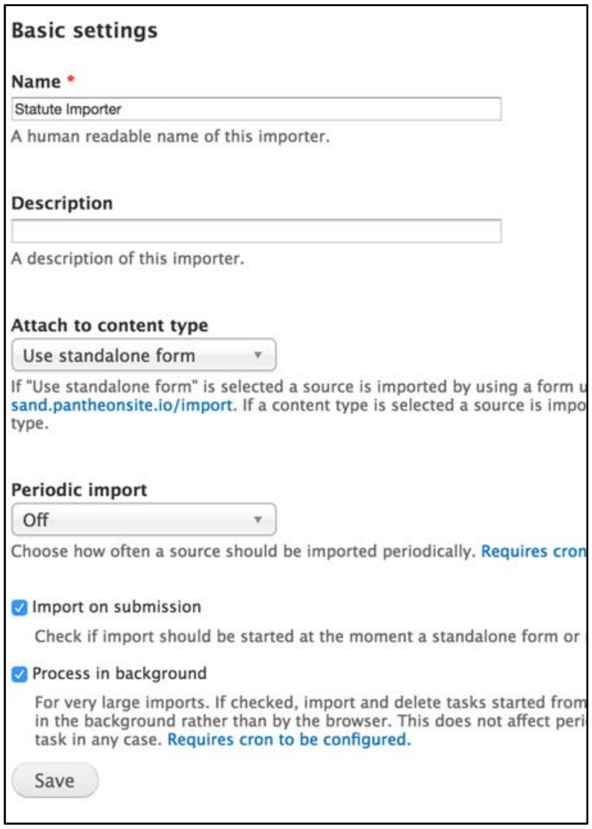
Getting started

You can create as many Feeds importer configurations as you would like to. Each can have a distinct purpose like letting your users aggregate RSS feeds or importing a CSV file for content migration. Here are a couple of things that are important to understand in order to get started with Feeds:

- Every importer configuration consists of basic settings, a fetcher, a parser and a processor and their settings.
- The **basic settings** define the general behavior of the importer. **Fetchers** are responsible for loading data, **parsers** for organizing it and **processors** for "doing stuff" with it, usually storing it.
- In Basic settings, you can **attach an importer configuration to a content type**. This is useful when many imports of a kind should be created, for example in an RSS aggregation scenario. If you don't attach a configuration to a content type, you can use it on the Import page.
- Imports can be **scheduled periodically** – see the periodic import select box in the Basic settings.
- Processors can have **mappings** in addition to settings. Mappings allow you to define what elements of a data feed should be mapped to what content fields on a granular level. For instance, you can specify that a feed item's author should be mapped to a node's body.

8) Under **Basic Settings**, click **Settings** for *Attached to:*. This allows you to change any general settings such as name or description when necessary.

9) Adjust the following settings as needed:



Basic settings

Name *

Statute Importer

A human readable name of this importer.

Description

A description of this importer.

Attach to content type

Use standalone form ▾

If "Use standalone form" is selected a source is imported by using a form u sand.pantheonsite.io/import. If a content type is selected a source is impo type.

Periodic import

Off ▾

Choose how often a source should be imported periodically. Requires cron

☑ Import on submission

Check if import should be started at the moment a standalone form or

☑ Process in background

For very large imports. If checked, import and delete tasks started from in the background rather than by the browser. This does not affect peri task in any case. Requires cron to be configured.

a. **Attach to content type** – by default, you're generally going to use the option *Use standalone form*. If you select any other item in the list (which pulls content types), the form will be attached to that specified content type when generating a new page from *Content*.

b. **Periodic import** – If you plan to manually upload the items, select *Off*. This is just a timer (which the function is provided by the job scheduler module) in case that the feed is always updated and you wish for it to automatically apply the updates.

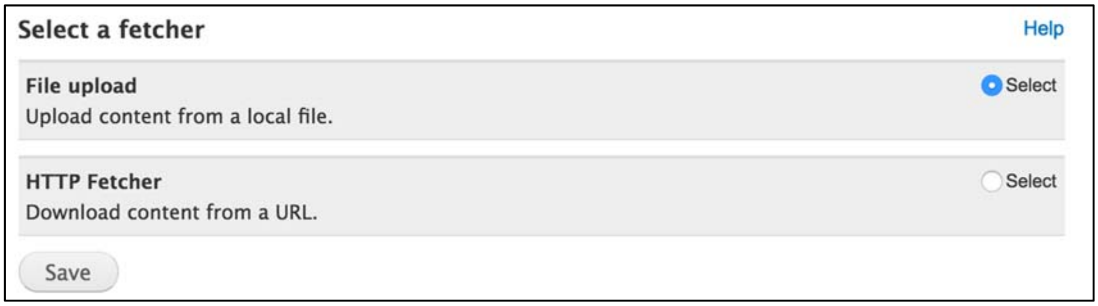
c. **Import on submission** – Allows the import start immediately when the form is submitted.

d. **Process in background** – If a large import, this is generally good to check. This will break down the import in the background rather than wait on the browser to respond.

10) Once configured, click **Save**.

11) For **Fetcher,** click **Change**.

12) Fetcher selects the option of how to receive the import by source. Using things like an RSS feed or something that provides a URL, then the *HTTP Fetcher* would be useful here. However, for this scenario, we will be uploading a CSV file that we have modified specifically for this situation. Because it is a local file, we will be selecting **File Upload.**



Select a fetcher

Help

File upload ● Select

Upload content from a local file.

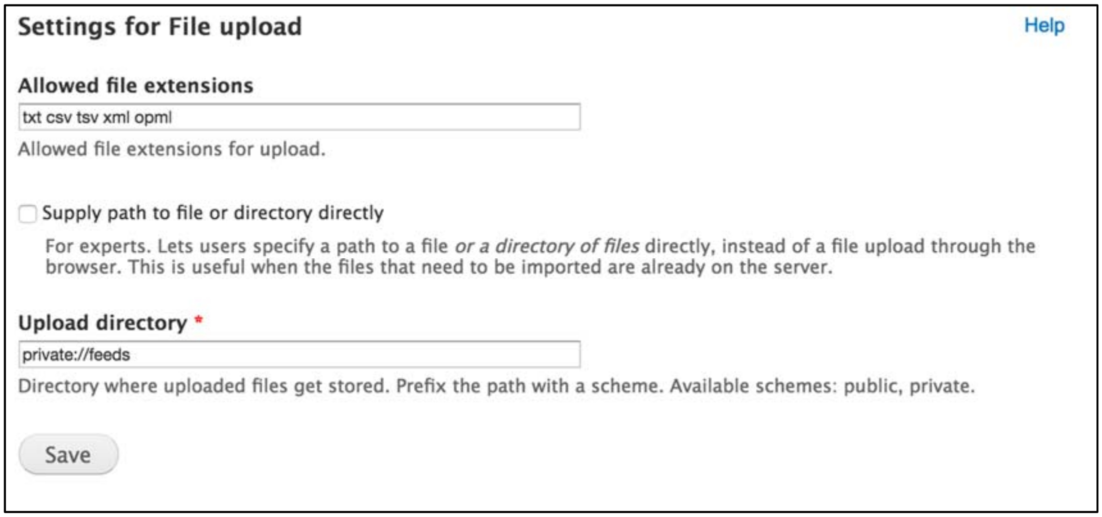
HTTP Fetcher ○ Select

Download content from a URL.

Save

13) Click **Save**.

14) For **File Upload** (which is now listed on the left), select **Settings**.



Settings for File upload

Help

Allowed file extensions

txt csv tsv xml opml

Allowed file extensions for upload.

□ Supply path to file or directory directly

For experts. Lets users specify a path to a file *or a directory of files* directly, instead of a file upload through the browser. This is useful when the files that need to be imported are already on the server.

Upload directory *

private://feeds

Directory where uploaded files get stored. Prefix the path with a scheme. Available schemes: public, private.

Save

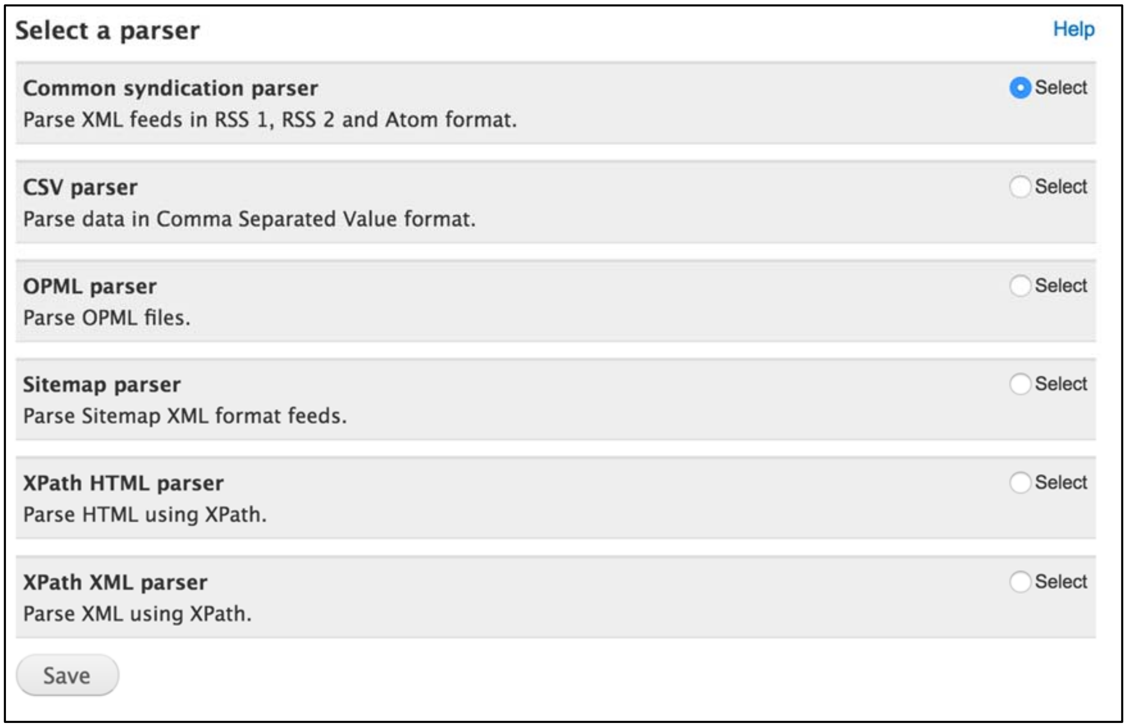
15) Verify the settings are accurate, then click **Save** when completed.
For this scenario, we will be leaving this as is.

a. **Allowed file extensions** – if your feed/file is a specific file type, ensure that it's extension is also listed here so that uploading it will not provide an error.

b. **Supply path to file or directory directly** – Typically you will not be using this, but in the case that you've uploaded the files already to the server, this allows you to provide that pathway.

c. **Upload Directory** – When uploading a file to this feed importer, where does this save to? It is ALWAYS recommended to keep *private://* prior to any folder additions. This ensures your upload is safe.

16) For **Parser**, select **Change**.



Select a parser

Help

Common syndication parser ● Select
Parse XML feeds in RSS 1, RSS 2 and Atom format.

CSV parser ○ Select
Parse data in Comma Separated Value format.

OPML parser ○ Select
Parse OPML files.

Sitemap parser ○ Select
Parse Sitemap XML format feeds.

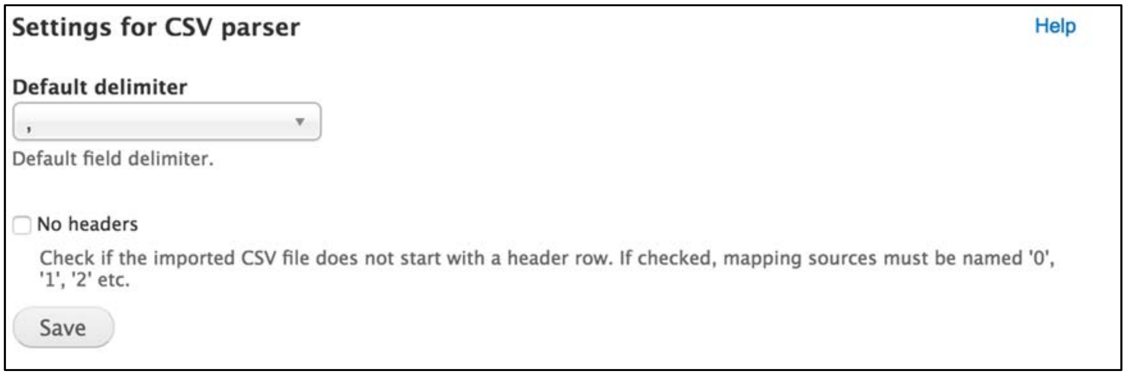
XPath HTML parser ○ Select
Parse HTML using XPath.

XPath XML parser ○ Select
Parse XML using XPath.

Save

17) More often than enough, you will be using just the **CSV parser**, select this option and click **Save**. This is just the type of the source file you will be taking your data from. It will change some setting options depending on what is selected here as far as assigning fields to data goes.

18) Under **CSV Parser**, select **Settings**.



Settings for CSV parser

Help

Default delimiter
, ▾
Default field delimiter.

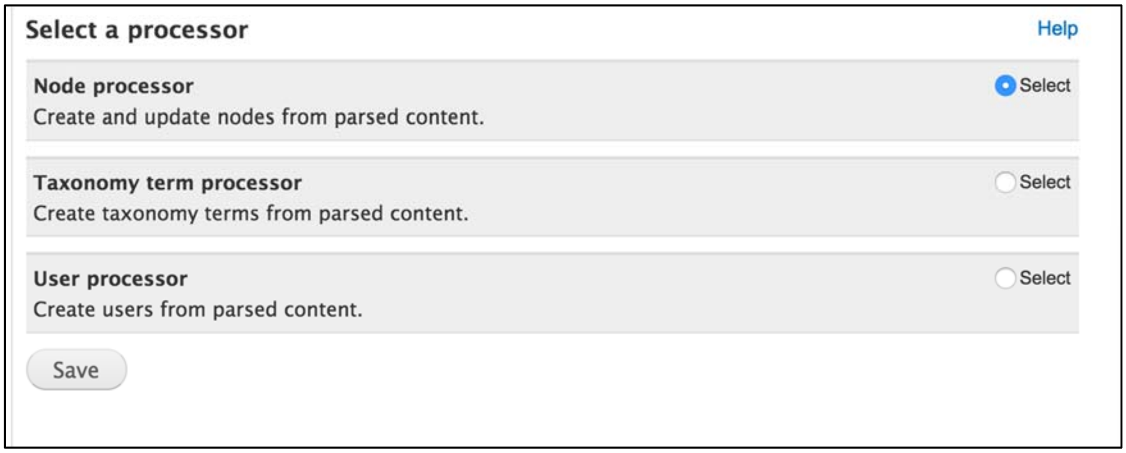
□ No headers
Check if the imported CSV file does not start with a header row. If checked, mapping sources must be named '0', '1', '2' etc.

Save

19) Based on how your CSV file has been exported, change any options necessary. In this case, we will be leaving the *Default delimiter* as is, and will leave *No headers* unchecked.

20) Click **Save**.

21) Select **Settings** from the **Processor** section.



Select a processor

Help

Node processor
Create and update nodes from parsed content.
● Select

Taxonomy term processor
Create taxonomy terms from parsed content.
○ Select

User processor
Create users from parsed content.
○ Select

Save

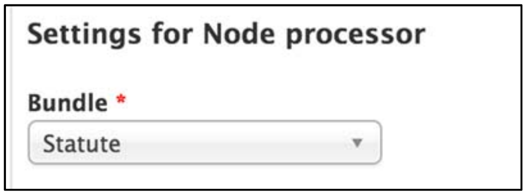
22) This section will help filter where this data will be applied to. You can import data to users, taxonomies, or nodes/content types. For this scenario, select **Node processor**.

23) Click **Save**.

24) Under *Node processor* on the left column, select **Settings**.

25) Modify the following settings:
For this scenario, please use the selections provided in the example image.

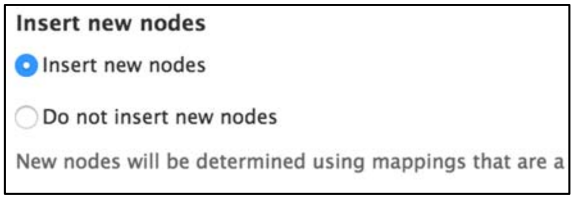
a. **Bundle** – Select the content type where the imports will be added as.



Settings for Node processor

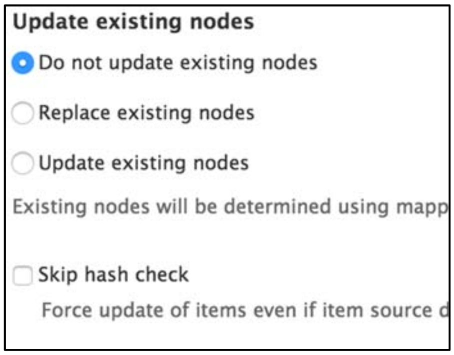
Bundle *
Statute ▾

b. **Insert new nodes** – This allows the importer to determine when to add new nodes if "unique targeting" is configured.



Insert new nodes
● Insert new nodes
○ Do not insert new nodes
New nodes will be determined using mappings that are a

c. **Update existing nodes** – This will allow the importer to determine if to update an existing node, replace it, or to do nothing when configured for "unique targeting".



Update existing nodes

● Do not update existing nodes

○ Replace existing nodes

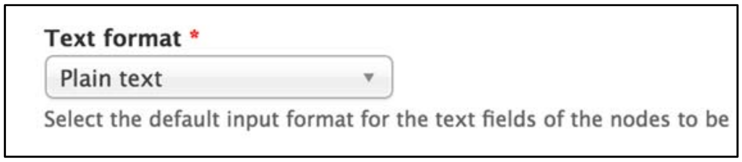
○ Update existing nodes

Existing nodes will be determined using mapp

□ Skip hash check

Force update of items even if item source d

d. **Text format –** Traditionally, *Plain text* is selected here. It assigns the default input format for text fields.

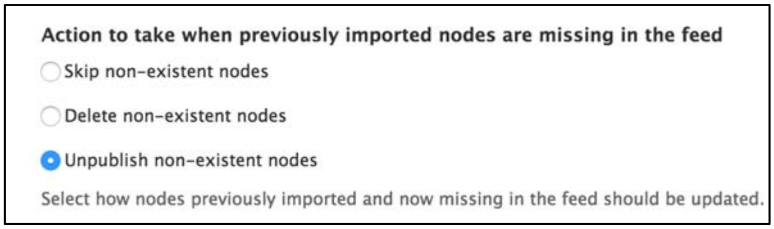


Text format *

Plain text ▾

Select the default input format for the text fields of the nodes to be

e. **Action to take when previously imported nodes are missing in the feed –** Informs how to handle when an updated import contains missing items that were there from a previous import.



Action to take when previously imported nodes are missing in the feed

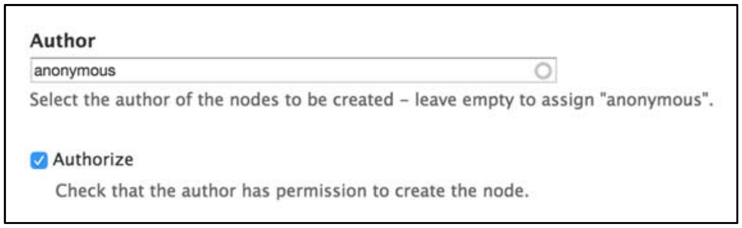
○ Skip non-existent nodes

○ Delete non-existent nodes

● Unpublish non-existent nodes

Select how nodes previously imported and now missing in the feed should be updated.

f. **Author –** Allows you to define who should be the author of the imported nodes. Typically, most people use *Anonymous* here.



Author

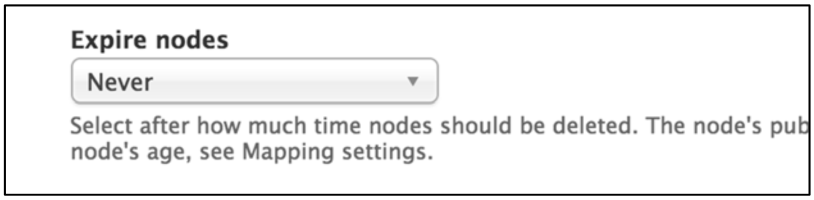
anonymous

Select the author of the nodes to be created - leave empty to assign "anonymous".

☑ Authorize

Check that the author has permission to create the node.

g. **Expire nodes –** Allows you to set a time limit for when the nodes should be deleted.



Expire nodes

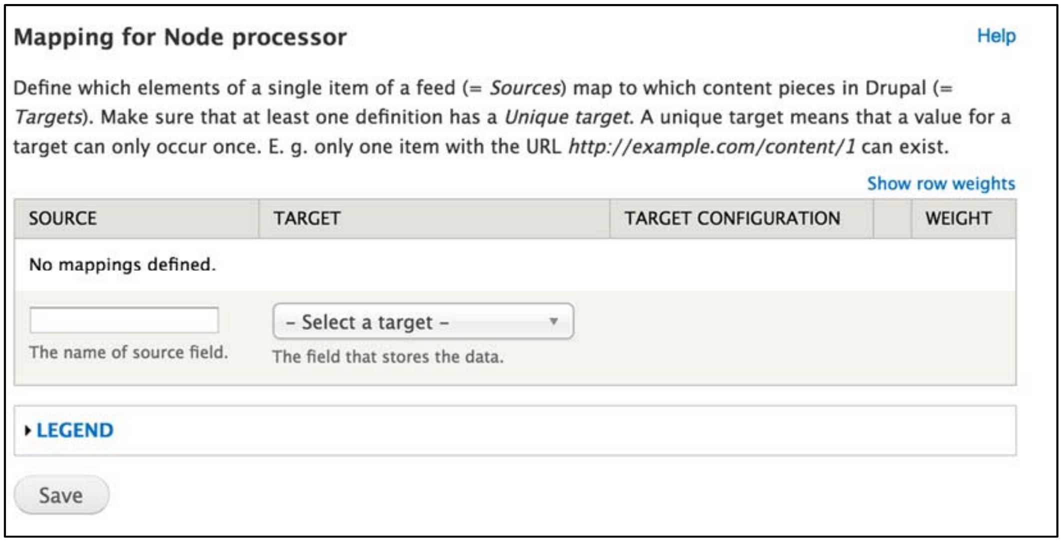
Never ▾

Select after how much time nodes should be deleted. The node's pub

node's age, see Mapping settings.

26) Select **Save**.

27) Under *Create and update nodes*, select **Mapping**. With this settings page, we will be mapping the columns from the CSV file to fields of the content type.



Mapping for Node processor

Define which elements of a single item of a feed (= *Sources*) map to which content pieces in Drupal (= *Targets*). Make sure that at least one definition has a *Unique target*. A unique target means that a value for a target can only occur once. E. g. only one item with the URL *http://example.com/content/1* can exist.

SOURCE	TARGET	TARGET CONFIGURATION		WEIGHT
No mappings defined.				
The name of source field.	– Select a target – ▾ The field that stores the data.			

28) To add a mapping from the file to the site, follow the below. Please keep in mind, this must be exact for naming conventions. If you accidently mistype and do not catch this, you can always remove the entry. You can only add one at a time.

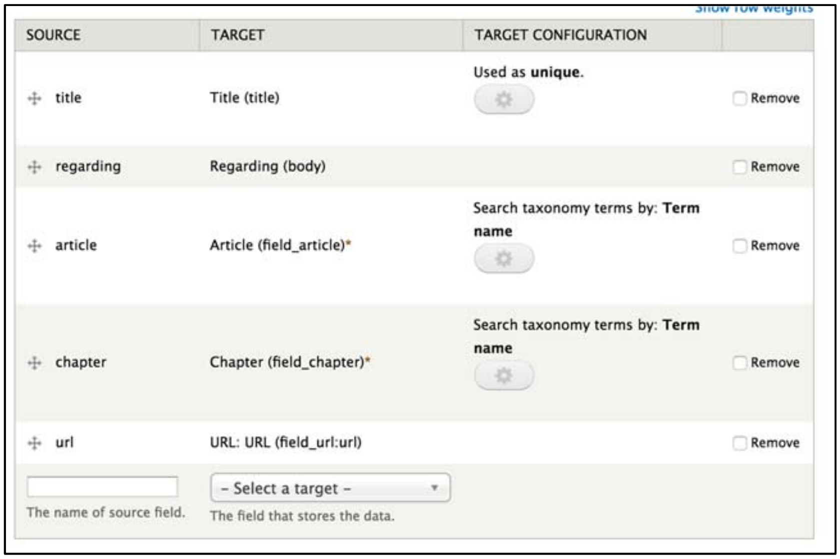
    a. In the empty text box, enter the header of the cell (usually based on columns) from the CSV file.

    b. In the drop down box, select which field this should be linked to.

    c. Click **Save**.

    d. If the field is unique, and will not be used any other time then the one instance, click the gear icon if presented. Check **Unique**, and select **Update**. This type of setting configured will allow updating nodes if something were to change later. Other additional settings may appear in this same column based on which fields are being used.

e. Reorder, Add additional, or Remove if necessary, and click **Save** when finished.



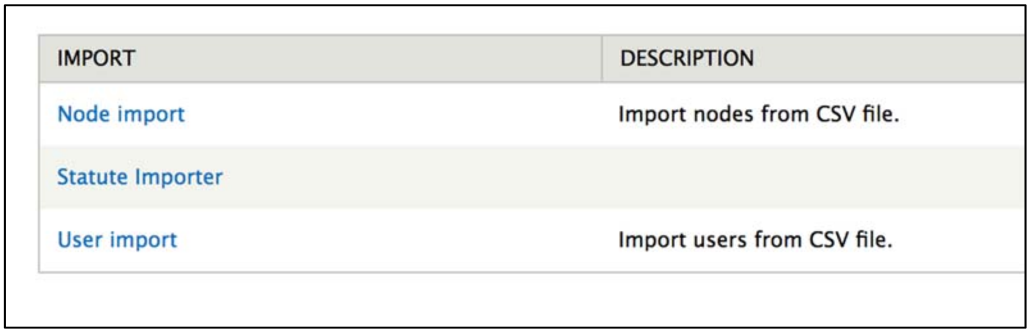
Show row weights

SOURCE	TARGET	TARGET CONFIGURATION	
title	Title (title)	Used as **unique**.	☐ Remove
regarding	Regarding (body)		☐ Remove
article	Article (field_article)*	Search taxonomy terms by: **Term name**	☐ Remove
chapter	Chapter (field_chapter)*	Search taxonomy terms by: **Term name**	☐ Remove
url	URL: URL (field_url:url)		☐ Remove
The name of source field.	– Select a target – ▾ The field that stores the data.		

Import Your Data

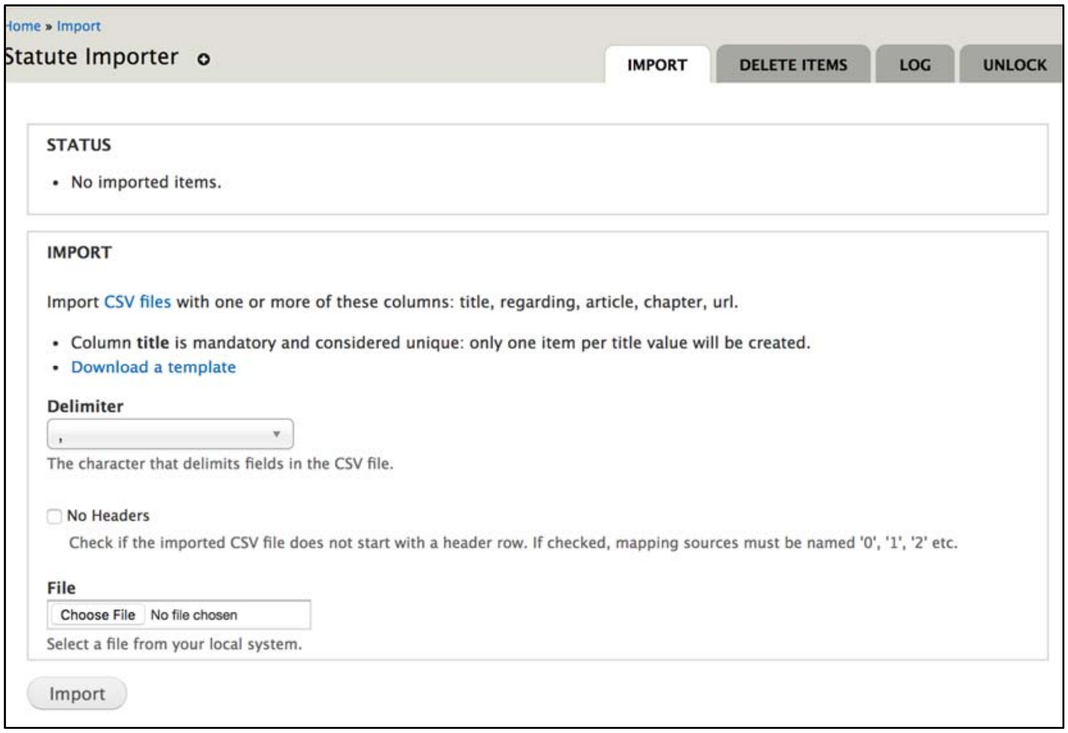
1) Navigate to the import page by adding **/import** to the end of your *root* url. You should be presented with a few options of imports.

   (Please note that you can always add your own shortcut to the admin tool bar to this URL to make the process quicker and easier to find).



IMPORT	DESCRIPTION
Node import	Import nodes from CSV file.
Statute Importer	
User import	Import users from CSV file.

2) Select your appropriate feed. In this scenario, we’ve created a feed named **Statute Importer**.



Home » Import

Statute Importer

STATUS

- No imported items.

IMPORT

Import CSV files with one or more of these columns: title, regarding, article, chapter, url.

- Column **title** is mandatory and considered unique: only one item per title value will be created.
- Download a template

Delimiter

,

The character that delimits fields in the CSV file.

□ No Headers

Check if the imported CSV file does not start with a header row. If checked, mapping sources must be named '0', '1', '2' etc.

File

Choose File No file chosen

Select a file from your local system.

3) Under *Import*, select **Choose File**, and upload the required file.

4) Select **Import**. The page will refresh when completed, and inform how many items were imported and or updated.

5) Verify your import has been successful.

6) Modify any CSS, Views, or other configurations as needed for the output of your feed.