

# FARGO: WRITE AND PUBLISH OUTLINES IN OPEN FORMATS

MARCO FIORETTI

Turn the web upside down with this text outliner – without installing a single piece of software.

**WHY DO THIS?**

- Prepare yourself for the open, distributed web of tomorrow, in an easy and fun way.
- Publish a nice-looking personal blog for free, in five minutes, without installing anything.
- Get familiar with OPML. You may need it again some day. Trust us.

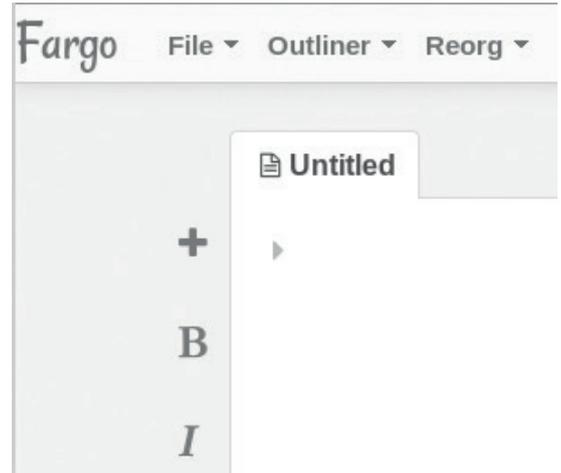
When you look at it closely, much written text has the same basic structure. Newspaper articles, philosophy essays, novel summaries, courseware, recipes... are all outlines – that is, hierarchical trees of topics and sub topics. If this is true, the more a software editor takes it into account, the more efficient it is, right?

*Fargo* is an outliner – that is a text editor designed to handle outlines in the most efficient way. Any outliner program provides tools to quickly navigate the elements of an outline and rearrange them at will, with the smallest possible effort. Above all, outliners can instantly hide certain levels or branches of an outline, so that at any moment you only see the exact amount of content and level of detail that you want to see.

Outliners are nothing new. In fact, the real value of *Fargo* is not in what it does, but in how and where it does it. This tutorial explains how *Fargo* works and how to use it, mainly from the point of view of a Linux user who would like to integrate it with their other online and desktop activities.

Now, the *Fargo* user interface is deceptively simple. It's easy to find the menus and buttons that perform an action, but to work with this tool (rather than against it) you must first understand the *Fargo* philosophy and what it does under the hood. We may even say that assimilating where the hood is is the hardest part here. Consequently, we will devote more space to explaining what *Fargo* provides, and how, than to explain how to actually use single menus or panels.

There are three points that were the origin, and still are at the core, of the *Fargo* proposal. The first is the observation that modern JavaScript-capable browsers are very powerful and run on hardware, even including mobile devices, much more powerful than



Eye candy and formatting functions in *Fargo* are limited to the bare minimum, and there are two powerful menus for viewing and managing outlines.

10 or even just five years ago. No question about that, but the other points deserve more reflection.

*Fargo* also works on the assumption that today “the cloud is ubiquitous and reliable” (not to mention, we may add, affordable). Residents in rural areas of Western countries, plus almost everybody else, may disagree on this. The final point is about lock in and... let's discuss it at the end of the tutorial.

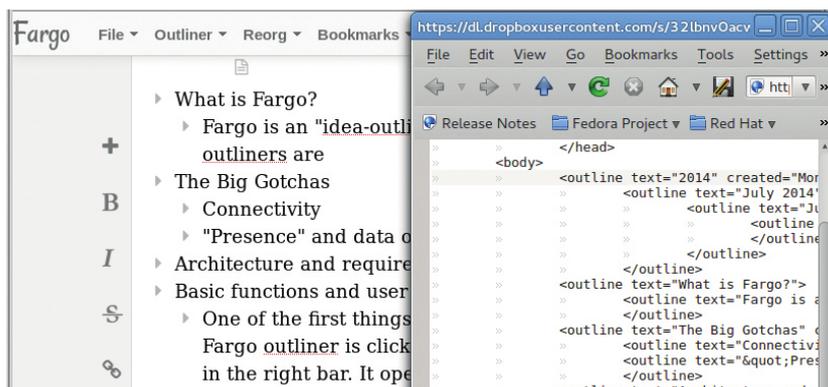
**Fargo architecture and requirements**

Installing *Fargo* is really simple: there is nothing to install! The only requirements are a browser that can handle JavaScript and HTML5 and a Dropbox account. Log in to Dropbox, point your browser to <http://fargo.io> and accept the request to let the *Fargo* app work in a dedicated subfolder of your account. If you don't see that request, it is because you've already been there. Tell your browser to erase all the cookies from the **fargo.io** domain and reload.

Thirty seconds later, you will be able to start writing outlines and publishing them online using an interface, and with a final result, already close to what you could get at **Tumblr.com** or **WordPress.com**, but without the lock-in.

This happens because, while *Fargo* is a static JavaScript app that runs entirely in the browser of your own computer or smartphone, it behaves as if it were a traditional CMS engine and produces the same results: you can always write and archive outlines in the same way from any device and location. From the

We see two very important things here: first, text written in *Fargo* looks very clean and easy on one's eyes. Second, that all your works remain available in open source formats.



viewer's point of view it's the same too: everybody can access all the outlines that you made public as if they were a traditional website. *Fargo* can also generate static HTML versions of your outlines and upload them to a web server whenever you want.

This is why *Fargo* has the potential to turn the web upside down. The current model of web self publishing and working "in the cloud" is based on central CMS servers doing all the really heavy work, from database queries to page rendering, for many thousands of authors and visitors simultaneously. This architecture demands servers and data centres with very high costs and environmental impacts.

In the *Fargo* model, as much as possible is decentralised. Only sensible data such as passwords are stored in your device. Raw outlines are still stored on servers; that is, in a private folder of your Dropbox account, but all the processing happens in the browsers that run *Fargo*, or in those that display its static HTML pages.

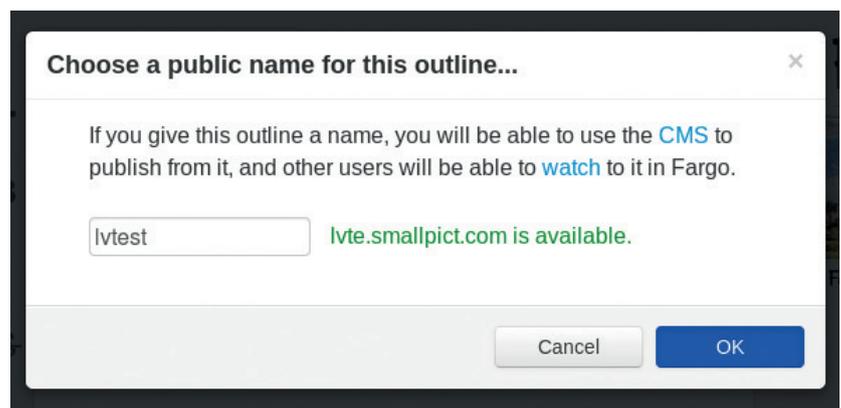
For authors, *Fargo* has another big advantage on server-based publishing systems like WordPress: since all the CMS logic runs in a browser, it can have a much more flexible and responsive interface, and provide a structure that naturally matches the structure of most writing.

### Structure of Fargo content

At the low level, each *Fargo* outline is a separate OPML file stored in your Dropbox account (see the OPML box below to understand what OPML is, and why it is great regardless of *Fargo*). Using Dropbox as filesystem also provides automated backups and versioning for free, even if you still have to backup everything outside Dropbox regularly.

The single elements of each outline, which can be nested at will, are called headlines (or even summits, if at the top level). We would have preferred terms like node or paragraph, because each *Fargo* headline can be as long as you want, and each time you press Enter, you create a new one, but headline it is.

Besides its unique position in the hierarchy, which of course you can change as you want, each headline can have attributes like identification code, creation



*Fargo* would be very useful even as a purely private editor, but it couldn't be easier to transform content in to a blog.

date or author-defined data, or be commented out. In the latter case, the headline will remain in the OPML file, but out of sight, and it will never be included in the HTML versions. We will explain how to comment or assign attributes in a moment.

All your Dropbox files are private, until you ask *Fargo* to create public, but read-only links to them. An outline can even embed content from external websites, if you pass them to *Fargo* with the browser Bookmarklet linked from the right bar.

When an outline grows unwieldy, you can archive all its headlines that you don't need to edit anymore, and still make them show up (and render) in the outline. To do that, you have to archive those headlines as "includes". Do do this, place the cursor on them, select File > Archive Cursor, and they will be moved to the **archive** sub-folder of your *Fargo* folder in Dropbox.

### Images and interactive content? Of course!

In case you were worrying that a system optimised for outlines doesn't support anything but static text, relax! You can tell *Fargo* to keep an eye on a Dropbox subfolder for generic media (images, audio, PDFs, whatever). Then, any time you upload something there, *Fargo* will notice it and give you a URL for it in a pop-up window. You can add as much interactivity as you want to your *Fargo* outlines... as long as you write

#### LV PRO TIP

If you have a lot of texts on your hard drive that you would like to import in *Fargo*, don't worry. It's very likely, you can automate much of that work. One of the best tools for the is *Pandoc* (<http://johnmacfarlane.net/pandoc>): a very versatile converter that can transform any of dozens of formats into any of the others.

### What is OPML?

Really open file formats and communication standards are arguably even more important than free software. If somebody else sends you files in one of those formats, you can merrily ignore if they use proprietary software, and open those files with whatever application you prefer, directly on Linux.

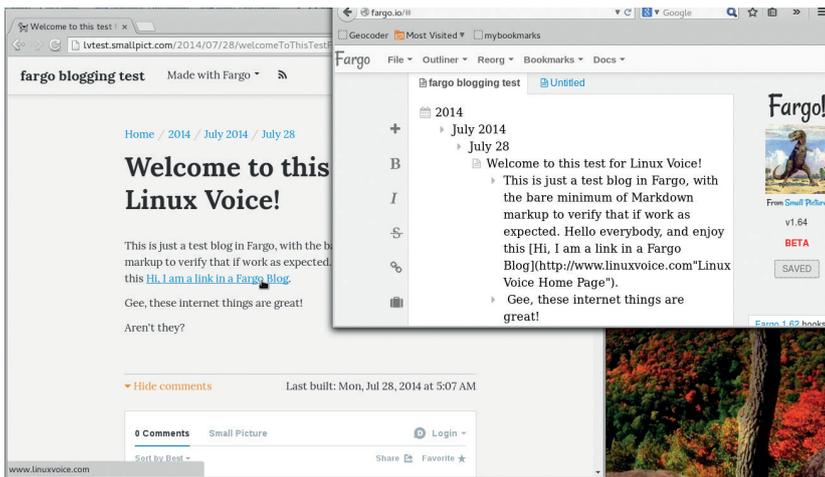
In the open formats family, the Outline Processor Markup Language (OPML – <http://dev.opml.org/spec2.html>), was developed specifically to process and exchange outlines. You have probably already seen it, or at least one of its applications: the list of links on the side of many websites known as "blogrolls" are just that: outlines that under the hood are most probably OPML files.

The most frequent application of OPML, at least on the web, is the automatic exchange of lists of RSS feeds between the

websites and software programs that generate, process and syndicate such feeds.

From a technical point of view, OPML is nothing other than another application of XML. In practice, this means that an OPML file, while terribly verbose, is just plain text that you can generate, parse and process automatically with many free software tools, from custom scripts to specialised editors.

It is equally important to realise that there's nothing to limit OPML to handling lists of headlines and relative links and abstracts. Formally speaking, OPML can handle anything whose structure is a hierarchic tree of nodes, each containing named attributes in text format. If you think about it for just a moment, you will realise that even your family tree, or your company's organisation chart, match this description.



What looks just like very well structured text, automatically becomes a simple blog, complete with Disqus comments, with just a few clicks.

it in JavaScript, as *Fargo* itself. In general, the developers have already started to think about JavaScript “verbs” for *Fargo* that would make such tasks easier. See <http://docs.fargo.io/fargoScripting/> for details.

You can already add snippets of JavaScript to a headline (including calls to internal *Fargo* functions) and run them by pressing `Ctrl+.` It is also possible to run some JavaScript code automatically, every time you reload *Fargo* or publish an outline.

### The Fargo user interface

The first thing you want to do in your *Fargo* outliner is click on ‘Cribsheet’ in the right sidebar, to get a cheatsheet with all the main commands. Next, you should take a look at the many resources in the Docs top menu. Just remember that whenever those pages say “Cmd”, (as in the Command key on OS X) what they mean on Linux is the Control key.

Now, let’s talk configuration. To access the *Fargo* configuration tabs, click on your name in the top-right corner of the browser window and select “Settings”. Besides a multimedia folder here, you can set a password to encrypt all your private outlines, the

### Integration with WordPress

Many bloggers simply cannot give up their WordPress accounts for *Fargo*, because they need some special plugin or, much more simply, they are just (co-)authors, not the owners of those blogs. What should they do, if they find the *Fargo* authoring interface much better than the WordPress one? Post to WordPress from *Fargo*, of course (only one blog per *Fargo* account, sorry!). The “Blog” tab of the *Fargo* settings interface is there just for that purpose: enter the URL of your blog, your username and password.

If you need to format your blog posts in ways that *Fargo* doesn’t support, just check the Markdown box, and all the markup you add in your headlines will automatically be converted to HTML before sending it to the blog. After this initial configuration, every time you want to post create a new headline for the title, another right below it with the content, and click on the WordPress icon in the left sidebar.

autosave behaviour and your contact information (profile page, email, Twitter and Facebook accounts). In the same place, you can define separate CSS styles for each level of your outlines, or a background image.

*Fargo* can handle multiple outlines simultaneously, each in a different tab. The editor marks each headline with a wedge on the left, which will be black if there is unexpanded test underneath it, or grey otherwise. The actual content of a headline can be formatted with HTML or Markdown syntax.

Setting the standard attributes of a headline, or giving it custom ones, is easy: select the headline, click on the suitcase icon (or select Outliner > Edit Attributes) and enter the attribute name on the left and its value on the right. Click on the + button if you also need to add custom attributes, and repeat. Headlines can also be individually commented by pressing `Ctrl+\`. When you do that, their wedges will become chevrons. To uncomment them, press `Cmd+\` again.

### Working with Fargo

Looking at *Fargo* as just an editor, its two main features are the ‘Outliner’ and ‘Reorg’ top menus. The first is used to control how much you see of the current outline and toggle between Non-Render and Render mode: use the first mode to write or edit raw markup inside an outline, and the other to see what the results looks like.

As the name suggests, the ‘Reorg’ menu helps to reorganise your writings. The entries to move one or more headlines up or down the outline they are in, or to change their indentation levels, are all there.

The main functions found in both those top menus are also available in ‘Pad’ format, to work faster on touchscreen devices. The *Fargo* ‘Arrow Pad’ (Outliner > Show Arrow Pad) has two buttons, one to collapse or expand parts of the hierarchy, and the other to toggle Navigate and Reorg mode. Depending on the mode the four arrow icons in the pad will let you move headlines around, or navigate from one to another.

On devices with real keyboards, you can use shortcuts for almost all menu entries. Tab and Shift+Tab, for example, increase and decrease the indentation level of a headline. Remember that in *Fargo* pressing the Enter key does not enter a newline, or split the current text in two. It just add one more empty headline below the current one, regardless of where the cursor was when you typed.

Once you have acquired familiarity with the outline-oriented interface of *Fargo*, you will also be able to use it to build a public, simple blog. The post shown in the image above-left was created in four main steps (the extra, really simple details are all at <http://fargo.io/docs/blogging/firstPost.html>):

- 1 Create a new outline (File > New).
- 2 Give it a name (File > Name Outline), let’s say ‘goinuxvoice’.
- 3 Write some content in the usual way.
- 4 When you are done, put the cursor on the top

#### LV PRO TIP

Take advantage of *Fargo* to reorder all those disorganised folders that you likely have in your Dropbox account! This will make it much easier to keep stuff you want to publish through *Fargo* from everything else, and you should already be doing it anyway.

headline, and click the Eye icon in the left bar.

The last action will create a new subdomain, **glinuxvoice.smallpict.com** (Small Pict is the company that develops Fargo). All visitors of that domain will be transparently redirected to static HTML copies, organised like a blog, of all the posts that you add to that named outline. The documentation also explain how to add WordPress-like categories or generate RSS feeds.

If you plan to use *Fargo* just for private outlines, but occasionally want to share one of them with others, in read-only mode, select File > View In Reader: this will produce a public URL of your outline that you can distribute to your friends, students or colleagues.

## Desktop integration and automation

What you have learned so far is enough to make most aspiring authors of outliners and personal blogs happy, but we Linux users are more demanding than the average bear.

Writing outlines and optionally publishing them online with *Fargo* is easy and efficient, but could we do more? For example, would it be possible to reuse *Fargo* content in other publishing systems, with as little manual work as possible?

Or what about writing outlines locally (even when there is no connectivity), and uploading them automatically when you connect to the internet?

The first activity – re-use – is pretty easy. Set up the Linux client for Dropbox to automatically copy all the raw outlines onto your computer in OPML format, then play with tools like *Pandoc* to convert them to other formats, as in these two examples:

```
#> pandoc -f opml -t html outline.opml > outline.html
```

```
#> pandoc -f opml -t markdown outline.opml > outline.md
```

In other words, it's easy to avoid being locked into *Fargo* as an outline-based editor.

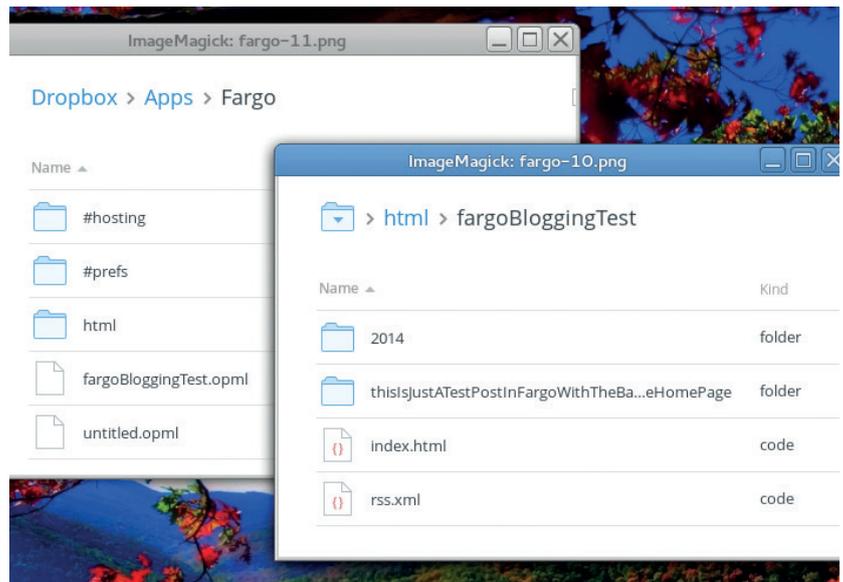
The reverse path – that is, generating OPML outlines on your computer and using them in *Fargo* – is not possible yet. Not directly, at least. If you put OPML files in your *Fargo* folder at **Dropbox.com** nothing will happen. The only available workaround so far seems to be uploading those files somewhere else, and then to tell *Fargo* to include them. This location can even be another subfolder of your Dropbox account, as long as you share it to get a publicly accessible URL usable by *Fargo*.

## Control, and alternatives

All this finally leads us back to the final basic point of *Fargo*, the one that we only hinted at in the beginning, and to the future developments of this technology.

One of the official announcements of *Fargo* proudly points out that using it “you have a lot more freedom about where you host your website”. In reality, as you should have already noticed, things are quite different from that, at least now and for average users.

On one hand, you have to have a Dropbox account and let them “see” your private documents, which is not all that comfortable in this post-Snowden era. On



This is where all your raw content, obviously in open formats, ends up in *Fargo*: inside dedicated folders and subfolders of the *Fargo* app space of your Dropbox account.

the other, if you want to use *Fargo* for blogging, your online presence will only be as stable as the **smallpict.com** domain name, and the willingness of its owners to let you use it for free.

Wouldn't be great if all the servers *Fargo* needed were a Raspberry Pi under your desk, and it could use any domain name of your choice?

Truth be told, Dave Winer and the other developers of *Fargo* do see all the limitations, and are more than willing to overcome them. In fact, we already have some alternatives today, and a road ahead to solve the problem for good.

The already existing, but radical solution to the problem just mentioned is to not use *Fargo*. If you think about it, a desktop-based outline editor coupled with a static blog engine like *Mynt* or *Jekyll* already provides most of what you may get from *Fargo*. Especially on Linux, which gives you the ability to couple it with the right set of shell scripts.

At the same time, it is hard to beat the ease of use and device independence of *Fargo*. And the companion free software of *Fargo* called *Fargo Publisher* (<https://github.com/scripting/fargoPublisher>) can already transfer HTML versions of *Fargo* outlines to any server of your choice, solving the domain name problem for good. The process is quite complex, but Chris Dadswell, who is already using it, made a great job of documenting all the steps at <http://scriven.chrisdadswell.co.uk/articles/howtofargoselfpublishingstorageoptions.html> and <http://scriven.chrisdadswell.co.uk/articles/howToSelfPublishingFargoBlog.html>.

The Dropbox dependency remains, but with any luck we'll also get over it. Stay tuned for another tutorial when that day comes! 📖

**Marco Fioretti is a Free Software and open data campaigner who has advocated FOSS all over the world.**

### LV PRO TIP

Markdown (<http://daringfireball.net/projects/markdown/>) may be the most popular, if not the most versatile, markup system for plain text available today. Learning to write and convert text with Markdown, regardless of *Fargo*, would be a very smart move if you want to publish lots of text regularly.

### LV PRO TIP

You can transform your outline in online presentations as explained at <http://fargo.io/docs/presentations.html>.