

# The State Decoded

Legal codes, for humans.

## Version 1.0 Released

Posted on [April 24, 2017](#)

[Version 1.0 of The State Decoded](#) is now available on [GitHub](#). This marks the first official stable release of The State Decoded, which is now feature-complete.

Here are some of the highlights:

### Pinned Laws

Laws can now be "pinned" to save a list of favorite laws to review later. The pinned laws list is available from the main navigation, and to pin a new law either click the pin icon on the law page, or press the "p" key on your keyboard.

### Better Support for Inconsistent Data

A major change is the ability to have laws that share a section number. In response to law codes which have multiple sections with identical identifiers (such as [Virginia's § 16.1-69.6:1](#)), The State Decoded will now display all laws that match a given identifier on a single page. This removes the previous functionality where only one of these laws would be able to be viewed, effectively hiding the others.

There is also now better handling of sections where only some paragraphs have been repealed. Where previously the numbering would have led to inconsistencies in labelling, we can now effectively show this text as it was originally represented.

### Import & Export Improvements

The import engine has been overhauled to use a different XML parser, resulting in better handling of mixed content and structured data. Structural units that do not have their an identifier or number are also now supported, as well as legal codes that do not have a uniform, consistent structure.

The export system has also been reworked to use a plugin-based architecture, making it easy to add new file download formats to the system. PDF, Word, and EPUB downloads are now also included by default.

### Improved Command Line Tool

The command line tool can now be used to import data, manage editions, and test the environment without using the web interface. This makes it much easier to automate updating data.

\* \* \*

[Version 1.0](#) can be [downloaded now](#), and is also available as a [Vagrant image](#). Development is ongoing, and there are plenty of [open issues for future releases](#)—pull requests are always welcome!

Posted in [General](#)

---

## Miami-Dade Decoded Launches

Posted on [December 3, 2015](#)

The folks at Code for Miami and The OpenGov Foundation have launched [Miami-Dade Decoded](#), they announced in a [joint press release](#). It's a beta release, powered by v0.8 of The State Decoded (the current, stable release). The Miami Brigade of Code for America started work on this early last year, and made this a real community effort.

Posted in [General](#)

---

## Version 0.9 Released

Posted on [May 12, 2015](#)

### Version 0.9 Released

[Version 0.9 of The State Decoded](#) is now available on [GitHub](#). This release completes the work to store multiple editions of a code started in v0.8. Since we have made fundamental changes to how laws are stored, potentially breaking previous installations, we have added it as an additional pre-1.0 release.

### Edition Support

Site admins can now choose to import a new edition of the legal code, or replace the current edition of code. This allows multiple editions of each code to live in parallel with the others – so when your locality updates their code, you can create a historical record of previous versions. Right now you can browse and search previous editions of the code, and all related data is kept isolate to each version. In the future this will enable us to compare and show the differences between versions of the code, as Wikipedia and GitHub do.

### Improved Search

We now allow users to choose between the default Solr search engine and a new, optional MySQL search adapter. The MySQL adapter uses the built-in database to search, with no additional packages needed. This means that The State Decoded can now be run on most standard hosting without any extra setup. If your host can run WordPress, you can run State Decoded. Solr is still a far more powerful and robust solution, but users now have an easier setup option.

To support this, we've refactored the search system to better abstract the code. The result of this is that data is directly imported into the search system from the database instead of having to jump through XSLT hoops. This abstraction also means that new search engines can be used easily, by writing a new search interface adapter. For example, if anyone [wanted to write an adapter to use Elasticsearch](#), that's now much easier.

### Command Line Interface and Database Versioning

There is now a new command line tool that comes with the package called **statedecoded**. This command can be used to run various tasks, including updating the database when a new version of State Decoded comes out, and more. Data imports can be run via the tool as well, so it can now easily be scheduled with **cron** to import automatically on a regular basis. You can also create your own tasks to run using the tool.

The database structure itself can now be updated with this tool. This removes the need to delete the entire database and install from scratch every time the structure is changed.

### Municode Parser

We've added an example parser for working with Municipal Code Corporation's XML format, to go along with our parser for American Legal Publishing Corporation's data, and our default, generic parser. This improves the ability for localities to import data from their current codifier.

### And Much More

We've also made a large number of other bug fixes, design tweaks, and accessibility improvements.

As always, this work wouldn't be possible without the generous support of [The John S. and James L. Knight Foundation](#), who fund [The OpenGov Foundation](#)'s work on The State Decoded. We also couldn't have done this without the help of Grant Ingersoll of [Lucidworks](#) fixing our Solr issues, and support from other amazing members of the Solr community, including Doug Turnbull, Erik Hatcher, and John Berryman. Design suggestions were, as always, made by John Athayde of [Meticulous](#).

Posted in [News](#)

## Version 0.8 Released

Posted on [November 11, 2013](#)

[Version 0.8 of The State Decoded is now available on GitHub](#). This is our biggest release to date, in part because it's a combination of what was meant to be both the v0.8 and the v0.9 releases. That means that it took twice as long to produce this release as was planned, but it was worth it. It's comprised of 709 Git commits, more changes that were committed for all prior eight versions combined. This is the final major release prior to v1.0. The two biggest improvements are a total overhaul of the user interface with a responsive design and the integration of the [Apache Solr](#) search engine. Here is a rundown of the major changes:

### New User Interface

John Athayde and Lynn Wallenstein of [Meticulous](#) were responsible for dropping the old design and creating a new one from scratch. They've rendered the site almost unrecognizably different, which a highly modular, flexible design that looks good on screens of all sizes, is easy to customize, and will continue to grow with the State Decoded project. The layout is coded in [SASS](#), which is handy for a lot of designers, and employs a pluggable template system that makes it easy to drop in custom designs. It even has a nice print stylesheet for folks who still like to have a hard copy in their hand. There's no overstating what a huge improvement that this is.



### Search Engine

It's really insulting to call this a mere search engine. The team at [Open Source Connections](#)—John Berryman and Doug Turnbull, in specific—designed an implementation of Apache's Solr search platform that's optimized for legal data. Of course, we used this to add a site search engine with spellcheck and live search suggestions, but having legal data indexed by Solr (and the underlying Lucene system) facilitates all sorts of exciting new features in the realms of natural language processing, information retrieval, content recommendations, and machine learning.

### Code Base Overhaul

The complexity of The State Decoded and the needs of its users outgrew the code base and data structure that had functioned from v0.1 through v0.7. Long-time State Decoded contributor [Bill Hunt](#) took on this project, completing it in just a few weeks, implementing a routing table, moving a lot of functionality into controllers, and routing all queries to a permalinks table. The old approach looks pretty clunky compared to what Bill built.

### Setup Simplified

We've tested out The State Decoded on the major Linux distributions and hosting platforms, identified all of the changes that were needed to allow it to work smoothly in those environments, and modified the software accordingly. There's now an automated environmental test suite, to make sure that The State Decoded will work properly, with multiple paths of accomplishing the same task, to require as little work as possible to configure the server. The installation process has fewer steps than ever, as everything that can possibly be automated is automated.

### Plus

There's an entire workflow to handle new editions of codes, bulk downloads are created automatically, a sample XSLT is included for State Decoded XML, there's a default home page that doesn't require any customization, it now supports non-unique section identifiers (!), there's an API method for search, there's a proper administrative section now, we have [an assets repository for the design's Photoshop files](#), a sitemap.xml is automatically generated, every law now has [Dublin Core](#) embedded, and dozen of other things.

The work done by Bill Hunt, Meticulous, and Open Source Connections wouldn't be possible without the generous support of The John S. and James L. Knight Foundation, whose funding made it possible to hire them. And Bill's ongoing work on The State Decoded is courtesy of his employer, [The OpenGov Foundation](#), who now employs him to contribute to the project and to implement The State Decoded in cities and states around the U.S.

Thanks, too, to Chris Birk, Karl Nicholas, Nick Skelsey, Josh Tauberer, and Rostislav Tsiomenko for their suggestions, testing, and contributed code.

Posted in [News](#)

---

## Documentation as a Continuous Process

Posted on [November 4, 2013](#)

The plan was simple: we'd create The State Decoded, and then we'd write the documentation for it. The documentation would be a big HTML file or something, and we'd include a copy with the download, and also host a copy on [statedecoded.com](#). That's a pretty standard model for open source software, and it seemed best to emulate what others did.

That was a terrible plan.

Documentation, as it turns out, is a lot like software. *Documentation is a process, not a product*. This realization wasn't born of a single *ah-ha* moment, but instead out of the needs made obvious during the process of creating The State Decoded.

At first, project documentation was a few bare-bones notes on a GitHub wiki, so that folks would know how to install the software. But then we established a State Decoded XML format, for storing laws, which required a lot of documentation for internal use—that went up on the wiki, too. When we added an API to the software, that definitely needed documentation, and so that was another page on the wiki.

Things were getting unwieldy. As more people started contributing to the project, not only did more people need to understand how the software worked, but it also became hard to keep up with improvements to the code. It's easy to understand how software works when you wrote all of it, but it's rather harder when big portions of it were authored by others.

Perhaps most important, it needed to be easier for early adopters to try out The State Decoded before its v1.0 release, and that meant providing documentation that was sufficient for that task. Releasing software throughout the development process, in the open source style, means that documentation needs to be released throughout, too.

In short, open source software requires open source documentation.

There are some fine services available for hosting software documentation, but I didn't have to cast around long to find a clear path forward for The State Decoded: [GitHub Pages](#). Several people recommended it, so it seemed worth trying. GitHub Pages allows HTML and Markdown stored on GitHub to be published to a public website, using [Jekyll](#), an open source project. As with any GitHub repository, others can make pull requests to propose modifications to those web pages, or be authorized to make any modifications that they like. When those changes are accepted or saved, they're reflected on the public website. Thanks to third-party services like [Prose.io](#), somebody need not even know how to use Git or GitHub, or even HTML or Markdown, in order to edit the documentation. This is a marked improvement over wiki-hosted documentation. GitHub Pages is GitHub's

GitHub's setup process walked me through creating [a State Decoded documentation repository](#), to which I moved the files from the wiki. Minutes later, the project had [a documentation website](#).

In the intervening few months, the documentation has been modified every time that a notable change was made, [hundreds of times in all](#). That said, it's still basically a dressed-up version of the wiki-based mess of a year ago. It needs to be restructured, ordered, and edited. But now it's on a platform that facilitates those improvements, by making it part of my desktop development workflow.

As long as The State Decoded remains an actively maintained project—for years to come, I hope—it needs to have documentation that reflects that. That requires documentation that changes with the software, and that is part of the same ecosystem as the software itself. Hosting software documentation on GitHub and publishing it via Jekyll is the right path for The State Decoded, and I think it's a route that other open source projects would be wise to follow, too.

Posted in [General](#)

---

## New Site: San Francisco Decoded

Posted on [September 26, 2013](#)

Today [the OpenGov Foundation](#) launched [San Francisco Decoded](#), their State Decoded-powered website that puts the laws of San Francisco online. The site is possible because [the San Francisco Mayor's Office of Civic Innovation](#) provided [the raw text of their laws](#), which the growing team at OpenGov used as the raw material to power the website. Bravo to SF CIO Jay Nath, his team, and the folks at OpenGov for their great work in making this happen.

Posted in [Member Site](#)

---

## Don't Let the Perfect Be the Enemy of the Good

Posted on [September 23, 2013](#)

The basic concept behind The State Decoded is both simple and obvious: create a platform to display laws in a nice, understandable way, using the data already present in those laws. So why hadn't anybody done it before?

Because it's too hard to do *perfectly*.

The State Decoded is not perfect, by design. Its [definition scraper](#) may never identify 100% of defined words within legal codes, because legislators are inconsistent about how they write laws. Cross-references will not always be identified and linked, because they're legislators are inconsistent about how they write those, too. Some laws' hierarchical structures may not be indented properly, because they're labelled inconsistently. The State Decoded's interface with court ruling APIs may never return only the court rulings that affect a specific law, as opposed to rulings that merely *mention* a law, because courts don't provide metadata.

I'm not sure that it's technologically possible, at present, to solve these and a dozen other problems inherent in The State Decoded. Some very bright minds have looked at creating a State Decoded-like system over the past few years, and decided against it because of insurmountable obstacles. They were right about the scope of the problems, but I think they were wrong to conclude that they shouldn't proceed anyway, and create a system that's 99% of the way there.

Just look at a few state code websites. I picked a few out at random: [South Carolina](#), [Missouri](#), [New York](#), and [Maine](#). These are awful. Just terrible. The percentage of citizens who are capable of navigating and understanding these is a rounding error. Having embedded definitions for 99% of legal terms, cross-references linked 99% of the time, and linked court cases that are good-not-great—that's all far, far better than what the citizens of these states have access to right now.

Sir Robert Alexander Watson-Watt, the radar pioneer who created England's system to detect approaching Luftwaffe, [said of England's radar system](#): "give them the third best to go on with; the second best comes too late and the best never comes."

The State Decoded is the third best system. The second best doesn't exist yet, but I look forward to somebody creating it and obviating The State Decoded. The best may never come. And that's OK.

Posted in [General](#)

## How You Can Help

Posted on [September 16, 2013](#)

I received an e-mail the other day from somebody asking how he could contribute to the development of The State Decoded. As a rule, this is a sign that I'm doing something wrong. In the spirit of addressing that that, here are a list of relatively self-contained, interesting, diverse features that await addition to The State Decoded, that you or somebody you know might be interested in creating, for folks of all levels of technical knowledge and many fields of expertise.

### Create the Functionality to Add Laws to a Portfolio

Site users ought to be able to keep track of laws that are of interest of them. Using jQuery's `localStorage.setItem` / `localStorage.removeItem`, provide the functionality to let people add laws to a portfolio, and then create a page where people can see a list of the laws in their portfolio.

[Issue #30](#)

### Support Memcached and/or ElastiCache

Provide an option in `config.inc.php` to provide configuration information to connect to the object cache of choice, and modify `class.Law.inc.php` to cache laws within the cache upon reading them or, if already cached, read them from there, rather than the database. (Presumably laws should be cached in Memcached upon being requested, rather than pre-loading Memcached full of all laws, since most legal codes aren't liable to fit within a reasonable amount of server memory.)

[Issue #263](#)

### Establish an Interface for Showing Diffs of a Law

With each new release of a legal code, we add a new edition (tracked in the "editions" table) and add all of those new laws to the "laws" table. Provide the functionality to let somebody look through the various versions of a law over time, or compare two versions to see how they've changed.

[Issue #363](#)

### Add Word, PDF, and EPUB Export

Add new methods to `class.ParserController.inc.php` to create, at the time that a legal code is imported, Word, PDF, and EPUB versions of the legal code, and portions thereof. (Realistically, this should be three separate issues, since it's three separate projects.) Ideally there'd be Word and PDF versions of every law, every structural unit (chapter, title, etc.), and the entire legal code, and then EPUB versions of every structural unit and of the entire legal code.

[Issue #50](#)

### Provide an Option to Use the OpenDyslexic Font

Create a jQuery-based widget to let somebody enable or disable the use of [the OpenDyslexic font](#), by setting a cookie, and then a jQuery-based widget to toggle the use of that typeface for the body font (`article#law`) if that cookie is set.

[Issue #340](#)

### Sync Laws to GitHub

Some folks are pretty psyched about putting laws on GitHub, for various reasons. Create a method that will commit the plain text version of all laws in a given edition of the legal code to a specified GitHub repository, and add the necessary options to `config.inc.php` to enable that.

[Issue #161](#)

### Provide Vagrant Configurations

There's an effort underway to create a ready-to-go Vagrant configuration of the project, so that it's trivial for somebody to set up an implementation of The State Decoded on their own system. This sub-project has its own repository, and a couple of issues logged in its own issue tracker.

[Issue #284](#)

### Display Related Legal Self-Help Documents

I've made a first crack at interfacing with ProBonoNet's API to gather up a list of all of their free self-help legal documents. This needs to be extended, to store this data in a way that's available to The State Decoded, and then—here's the hard part—when somebody looks at a law for which there's a relevant self-help legal document available, we need to be able to identify that document and display text promoting it. We've got the UI elements in place for this, but we just lack the glue that allows us to say “this law about foreclosure is probably related to this guide about what to do if your home is being foreclosed on.” Solr may be a good way to make this match.

[Issue #162](#)

### Edit, Write, or Propose Changes to Documentation

The State Decoded has [some pretty decent documentation that's under active development](#), but it would strongly benefit from review by people who *aren't* contributors to the project. (People who already know a project in great detail aren't in a great mindset to write about it in a way that beginners can understand.) The documentation is hosted on GitHub, so pull requests can be made directly, or, for folks who aren't technical, [suggestions or proposed changes can be made in the form of an issue report](#).

[Documentation Repository](#)

This isn't everything that needs to be done, of course—these are just the interesting, relatively self-contained new features. You can see [the complete list of outstanding issues](#) on GitHub, or just [the list of new features awaiting creation](#).

Posted in [How-To](#)

---

## First Documentation Release

Posted on [August 9, 2013](#)

[The first draft of The State Decoded's documentation is now available](#). Documentation was being built up piecemeal on a GitHub wiki, but it's [been moved to its own GitHub repository](#) and [a dedicated website](#). The pages are created in a mix of HTML and Markdown (Markdown has difficulty with some of the sample XML), and the site is built in Jekyll, which means that any changes made in the documentation GitHub repository are reflected promptly on the documentation website. This makes it simple for anybody to update the documentation—to fix a mistake, add an example, or even add a whole new section.

There's a great deal more to be done with the documentation. It needs to be organized, structured narratively, enhanced with illustrations, and simply cover more material. But it's not bad, and now it's easy for others to help make it better.

Posted in [News](#)

---

## Version 0.7 Released

Posted on [June 20, 2013](#)

[Version 0.7 of The State Decoded is now available on GitHub](#). This is a really meaty release, dedicated entirely to optimizations: it's faster, more efficient, easier to extend, easier to contribute to, easier to deploy, and easier to navigate. This release is comprised of a whopping 353 Git commits—that's more than every commit that went into versions 0.1 through 0.6, *combined*. Here are some of the major changes:

### Tuning

Every line of code was reviewed to see how it could be made faster—in tiny ways (instances of `stristr()` replaced with `strstr()`—or, better, `strpos()`) and in large ways (tossing out whole methods and starting again). The indices in MySQL were evaluated and revised, and any PHP error of level `E_NOTICE` and above was quieted. And the parser's memory usage has been reduced substantially, making it faster and more efficient to process large legal codes that previously might have strained (or broken) Apache's per-process memory limit.

### Caching

There are now hooks for both [APC](#) and [Varnish](#), so that folks running either of those popular caching applications (or, better, both of them) can reap the speed benefits. API keys, all constants, and templates are cached in APC now, with more caching on tap in upcoming releases.

### Development Environment

This version was developed substantially within a [Vagrant](#) staging environment, resulting in the inevitable optimization of The State Decoded to run within Vagrant. That involved a lot of tiny changes (e.g., respecting port numbers in URLs) that collectively create a smooth experience when developing locally. We have [the under-development Vagrant configuration for The State Decoded in its own repository](#). This version and all future versions will have a Vagrant machine image available for download, to make it trivial to get started. Vagrant is working on a path to deploy Vagrant machine configurations as [AWS](#) instances, which is why this is a bandwagon worth hopping on now.

### Standardization

Out with [HTML Purifier](#), in with [HTML Tidy](#). Out with [MDB2](#), in with [PDO](#). Out with late-nineties-style commenting and code formatting, in with PEAR-style commenting and code formatting. There was nothing *wrong* with any of those prior approaches, but it's best to establish an environment that contributors expect—that makes it easier for folks to contribute code to the project, or customize it for their own website. Also, HTML Tidy and PDO are already installed by default on a great many systems, which simplifies the setup process.

### Extensibility

Several steps have been made to facilitate customization. All non-obvious database columns are now commented, there's infrastructure for inline help text (stored and distributed as JSON), and there's support for importing, storing, and display arbitrary metadata fields alongside the standard data about each law.

### New Features

And, of course, we couldn't resist a few new features. There's now keyboard navigation within laws and structures, for those power-users who want to flip through laws quickly. There's baked-in support for [Disqus](#)-based commenting on each law page—just enter your site's Disqus shortname in `config.inc.php` and you're up and running. And, finally, there's bulk generation of both plain text and JSON versions of laws. To what end? Dunno—that's for you to figure out.

This release involved a lot of work over the course of four months. [Bill Hunt](#) has scrubbed in to help as a core contributor, and he's responsible for a lot of these improvements. Some very helpful bug reports, wiki edits, and pull requests came from [Chris Birk](#) and [Daniel Trebbien](#).

Next up: versions 0.8 and 0.9, which will be released soon, and close together. Version 0.8 will be comprised of [the UI/UX branch](#), which only has [a handful of small tasks remaining](#), thanks to months of work by [Meticulous](#) ([John Athayde](#) and [Lynn Wallenstein](#)). And Version 0.9 will be comprised of [the Solr branch](#), and is dedicated to baking [Apache Solr](#) into The State Decoded. That's based on several months of work by the team at [OpenSource Connections](#), who finished up earlier this week—there are only about [a dozen outstanding issues](#), all of which build on OSC's work to add some valuable new features to The State Decoded.

Posted in [General](#)

---