While the [new configuration system in Drupal 8](https://api.drupal.org/api/drupal/core%21core.api.php/group/config_api/8) strives to make the process of exporting and importing site configuration feel almost effortless, immensely complex logic facilitates this process. Over the past five years, the entire configuration system code was written and rewritten multiple times, and we think we got much of it right in its present form. As a result of this work, it is now possible to store configuration data in a consistent manner and to manage changes to configuration. Although we made every attempt to document how and why decisions were made – and to always update [issue queues](https://www.drupal.org/project/issues/search/drupal?text=&assigned=&submitted=&project_issue_followers=&status%5B0%5D=Open&status%5B1%5D=1&status%5B2%5D=13&status%5B3%5D=8&status%5B4%5D=14&status%5B5%5D=15&status%5B6%5D=2&status%5B7%5D=4&status%5B8%5D=16&status%5B9%5D=3&status%5B10%5D=5&status%5B11%5D=6&status%5B12%5D=18&status%5B13%5D=7&&&version%5B0%5D=8.x&component%5B0%5D=configuration%20entity%20system&component%5B1%5D=configuration%20system&issue_tags_op=%3D&issue_tags=), [documentation](https://www.drupal.org/node/1667894), and [change notices](https://www.drupal.org/list-changes/published?keywords_description=configuration&to_branch=&version=&created_op=%3E%3D&created%5Bvalue%5D=&created%5Bmin%5D=&created%5Bmax%5D=) – it is not reasonable to expect everyone to read all of this material. But I did, and in this post I try to distill years of thinking, discussions, issue summaries, code sprints, and code to ease your transition to Drupal 8.

In this article I highlight nine concepts that are key to understanding the configuration system. This article is light on details and heavy on links to additional resources.

1. **It is called the “configuration system.”** The Configuration Management Initiative (CMI) is, by most reasonable measures, feature complete. The number of [CMI critical issues](https://www.drupal.org/project/issues/search/drupal?project_issue_followers=&status%5B%5D=Open&priorities%5B%5D=400&version%5B%5D=8.x&component%5B%5D=configuration+entity+system&component%5B%5D=configuration+system&issue_tags_op=%3D) was reduced to zero back in the Spring and the #drupal-cmi IRC channel has been very quiet over the past few months. Drupal now has a functional configuration management system, but we only should call the former a CMS. While it is tempting to think of “CMI” as an [orphaned initialism](https://en.wikipedia.org/wiki/Acronym#Pseudo-acronyms), like AARP or SAT, we aspire to avoid confusion. Our preferred phrase to describe the result of CMI is “configuration system.” This is the phrase we [use in the issue queue](https://www.drupal.org/project/issues/search/drupal?project_issue_followers=&status%5B%5D=Open&version%5B%5D=8.x&component%5B%5D=configuration+system&issue_tags_op=%3D) and [the configuration system documentation](https://www.drupal.org/documentation/administer/config).
2. **DEV ➞ PROD.** Perhaps the most important concept to understand is that the configuration system is designed to optimize the process of moving configurationbetween instances of the same site. It is not intended to allow exporting the configuration from one site to another. In order to move configuration data, the site and import files must have matching values for UUID in the system.site configuration item. In other words, additional environments should initially be set up as clones of the site. We did not, for instance, hope to facilitate exporting configuration from [whitehouse.gov](http://whitehouse.gov/) and importing it into [harvard.edu](http://harvard.edu/).
3. **The configuration system is highly configurable.** Out of the box the configuration system stores configuration data in the database. However, it allows websites to easily switch to [file-based storage](https://www.drupal.org/node/2416555), MongoDB, Redis, or another favorite key/value store. In fact, there is a growing ecosystem of modules related to the configuration system, such as[Configuration Update](https://www.drupal.org/project/config_update), [Configuration Tools](https://www.drupal.org/project/config_tools), [Configuration Synchronizer](https://www.drupal.org/project/config_sync), and [Configuration Development](https://www.drupal.org/project/config_devel).
4. **There is no “recommended” workflow.** The configuration system is quite flexible and we can imagine multiple workflows. On one end of the spectrum, we expect some small sites will not ever use the configuration manager module to import and export configuration. For the sites that utilize the full capabilities of the configuration system, one key question they will need to answer regards the role that site administrators will play in managing configuration. I suspect many sites will disable configuration forms on their production sites – perhaps using modules like [Configuration Read-Only Mode](https://www.drupal.org/project/config_readonly) – and make all configuration changes in their version control system.
5. **Sites, not modules, own configuration.** When a module is installed, and the configuration system imports the configuration data from the module’s [config/install](https://www.drupal.org/node/2234799)directory (and perhaps the [config/optional](https://www.drupal.org/node/2453919) directory), the configuration system assumes the site owner is now in control of the configuration data. This is a contentious point to some developers because module maintainers will need to use [update hooks](https://api.drupal.org/api/drupal/core!lib!Drupal!Core!Extension!module.api.php/function/hook_update_N/8) rather than making simple changes to their configuration. Changing the files in a module’s config/install directory after the module has been installed will have no effect on the site.
6. **Developers will still use Features.** The [Features module in Drupal 8](https://www.drupal.org/project/features) changes how the configuration system works to allow modules to control their configuration. Mike Potter, Nedjo Rogers, and others have been making Features in Drupal 8 do the kinds of things Features was originally intended to do, which is to bundle functionality, such as a “photo gallery feature.” The configuration system makes the work of the Features module maintainers exponentially easier and as a result, we all expect using Features to be more enjoyable in Drupal 8 than it was in Drupal 7.
7. **There are two kinds of configuration in Drupal 8: simple configuration and configuration entities.** [Simple configuration](https://www.drupal.org/node/1809490) stores basic configuration, such as boolean values, integers, or texts. Simple configuration has exactly one copy or version, and is somewhat similar to using [variable\_get()](https://api.drupal.org/api/drupal/includes%21bootstrap.inc/function/variable_get/7) and [variable\_set()](https://api.drupal.org/api/drupal/includes%21bootstrap.inc/function/variable_get/7) in Drupal 7. [Configuration entities](https://www.drupal.org/node/1809494), like [content types](https://www.drupal.org/node/2192175), enable the creation of zero or more items and are far more complex. Examples of configuration entities include [views](https://www.drupal.org/node/1912118), [content types](https://www.drupal.org/node/2192175), and [image styles](https://api.drupal.org/api/drupal/core%21modules%21image%21src%21Entity%21ImageStyle.php/class/ImageStyle/8).
8. **Multilingual needs drove many decisions.** Many of the features of the configuration system exist to support multilingual sites. For example, the primary reason [schema files](https://www.drupal.org/node/1905070)were introduced was for multilingual support. And many of the benefits to enabling multilingual functionality resulted in enhancements that had much wider benefits. The multilingual initiative was perhaps the best organized and documented Drupal 8 initiative and their [initiative website](http://www.drupal8multilingual.org/) contains extensive information and documentation.
9. **Configuration can still be overridden in settings.php.** The $config variable in the[settings.php](https://api.drupal.org/api/drupal/sites!default!default.settings.php/8) file provides a mechanism for overriding configuration data. This is called the [configuration override system](https://www.drupal.org/node/1928898). Overrides in settings.php take precedence over values provided by modules. This is a good method for storing sensitive data that should not be stored in the database. Note, however, that the values in the active configuration – not the values from settings.php – are displayed on configuration forms. Of course, this behavior can be modified to match expected workflows. For example, some site administrators will want the configuration forms to indicate when form values are overridden in settings.php.