

What is Davis?

Davis is a “Data Management System” (DMS) primarily geared towards managing data that you want to present on the web. Think of a Content Management System (like Drupal, Wordpress, or Sitecore), except for your data. It comes with an administrative application that allows you to organize and publish your data (along with necessary metadata).

Davis is also a framework for building custom web applications on top of your data, such as a website to enable users to visualize data as maps, graphs, and tables across a wide array of measures. There is no “default” end-user facing application that comes with Davis. Instead, it is used as a starting point to build custom, full-featured data applications. The advantage of Davis is that not only will you be able to provide tailored data tools that meet your objectives and your audiences’ use cases, you’ll have a robust system for managing the underlying data, even if you have a library of hundreds or even thousands of separate measures.

Why Choose Davis?

Davis is built upon a structured, smart variable system that stitches together all your data behind the scenes. Variables are shared between data sets, meaning that the “Massachusetts” in one data set is the same as the “Massachusetts” in every other data set in the system.

Why is this important? Well, for example, let’s say you focus on US demographic data and you wish to publish online reports for all 50 states, each providing an overview of many different data measures. Due to the shared variable structure, reports of *all* data in the entire system for each of the 50 states can be generated. In addition, Davis’s intuitive data import wizard makes it easy to add new data sets—meaning you can continue to add data to the system and it will be automatically added to your state profile reports.

Smart, shared variables provide the foundation for other features into your data application too, such as:

- Sophisticated data visualizations. E.g. thematic maps, trend charts, and more.
- Multi-dimensional tables with consistent nested row/column formatting, like the example directly below. E.g. you may wish to build your tables such that columns map to years and row headers are nested. With smart variables, this is easy to enforce across all data sets.

| | | 2012 | 2013 |
|----|--------|------|------|
| MA | Female | 0.56 | 0.58 |
| | Male | 0.56 | 0.58 |

- Guided data exploration. E.g. structured data set browsing, faceted search, and “similar data sets” suggestions.

Davis vs. Datacenter?

You may be familiar with Velir’s Legacy platform – Datacenter. Davis is the next generation of Datacenter, and is suited for the same type of projects. However, Davis has been rebuilt from the ground up to make it more flexible and easy to use.

Technology

First, Davis has been rebuilt on an open source tech stack, using Node.js and PostgreSQL. This means that you are no longer limited to Microsoft hosting options only. Instead, you are free to shop around for the best hosting provider that suits your needs. From Amazon AWS to Heroku to Google—there are numerous cost conscious and scalable hosting options for Node.js applications!

Data model

The internal data model for Davis is more generalized and industry standard compared to Datacenter (the details below explain how). Davis has been generalized to work well with a wider variety of datasets. Gone are the days where you

will need to find a workaround to make your data “fit” the system.

Datasets vs. Indicators

Data in Davis are organized by *Datasets* rather than by *Indicators*. A *Dataset* is more general than an *Indicator*. It may contain data for a single indicator or multiple indicators, depending on your needs. For example, say you have 2 indicators: “Child Population” and “Adult Population.” You could either separate these into 2 datasets:

Child population

| Location | Year | Population |
|----------|------|------------|
| MA | 2012 | |
| NY | 2012 | |

Adult population

| Location | Year | Population |
|----------|------|------------|
| MA | 2012 | |
| NY | 2012 | |

OR, you could combine them both into one dataset.

Population

| Location | Year | Child Population | Adult Population |
|----------|------|------------------|------------------|
| MA | 2012 | | ... |
| NY | 2012 | | ... |

You could even combine all demographic indicators into a single dataset if you so desired. Davis tries to stay out of the way and lets you organize your data *however you want*.

Enhanced Variable Structure

The way in which variables are structured also has been upgraded with Davis.

- First, there are no longer mandatory variables. Datacenter required that all indicators have a *TimeFrame*, *Location*, and *DataType* variables. This is not the case with Davis, so the system is far more flexible in this way.
- Additional variables are no longer erroneously called *Distributions*. They are just *Variables*. A single dataset can have between one and infinite variables.
- Variables can be one of three types:
 - *Categorical*: These each have a defined set of “Attributes” that can be independently managed in the system. Examples: State, Race/Ethnicity, Age group
 - *Number value*: A quantitative value. Used to represent both discrete and continuous numerical variables. Examples: % of Population, GDP, Mortality Rate, Margin of Error, Ratio
 - *Text value*: A qualitative value that does not have a defined set of attributes. Examples: Last Name, Address
- Variables may or may not be *Shared* with other datasets. Datacenter required that all variables be shared between indicators. Davis on the other hand allows variables to be unique to a Dataset. This has the benefits of not artificially combining two variables, just because they have the same name. For example, say you have age breakdowns from two separate surveys, but the breaks are different. It makes sense to keep these two “Age” variables separate and distinct from one another, rather than sharing them across multiple data sets. By keeping them separate, you will always be able to see exactly which attributes belong to a certain variable.

Import Format

Because Davis uses a much simpler data model, the import format has been simplified and standardized. Each column simply represents a variable, and each row is an individual observation.

There are no more “DataType” and “Data” special columns.

Datacenter

| Location | TimeFrame | Gender | DataType | Data |
|----------|-----------|--------|----------|---------|
| MA | 2012 | Male | Percent | 0.5 |
| MA | 2012 | Female | Percent | 0.5 |
| MA | 2012 | Male | Count | 1000000 |
| MA | 2012 | Female | Count | 1100000 |

Davis

| State | Year | Gender | Percent | Count |
|-------|------|--------|---------|--------------|
| MA | 2012 | Male | 0.5 | 1000000 0 |
| MA | 2012 | Female | 0.5 | 1100000 0 |

In this example, there are 3 categorical variables: **State, Year, Gender** and two numerical variables: **Percent, Count**

Another Example - Davis

| Year | Make | Model | Engine Size | MPG | License Number |
|------|-------|--------|-------------|-----|----------------|
| 2016 | Honda | Civic | 1.5 | 35 | rkre91 |
| 2016 | Honda | Accord | 2.4 | 31 | 5ht9j5 |

In this example, there are 3 categorical variables: **Year, Make, Model**, two numerical variables: **Engine Size, MPG** and one text variable: **License Number**

Robust Variable Matching

One of the lessons we learned with Datacenter is that it can be problematic when different variables (distributions) are combined into one, simply because they have the same name. For example, one distribution might include many different Race/Ethnicity categories given that the categories

were derived from different data sources. Davis makes it easier to keep these variables separate with a robust variable matching process during the data import.

Programmable API

All of Davis’s core logic (the kernel) sits behind a modern [GraphQL API](#). In fact, the Davis Admin UI communicates with this kernel solely using this API. This means that the Admin UI can be cut out of your workflow for specific tasks if desired. The API opens us up to add features such as:

- Exporting data sets from SAS (or other systems) directly into Davis
- Creating special-purpose bulk data imports/exports
- Initiating a publish from an icon on your desktop, so that you don’t have to go into the admin system to do this.