Simple VIVO

Mike Conlon

October 6, 2015

# The Pump

## # Intro to the Pump, , The VIVO Pump (under development), provides a means for managing VIVO data using spreadsheets. Rows and columns in , spreadsheets represent data in VIVO. The Pump update method takes spreadsheet data, and a definition file, and , returns VIVO RDF (add and subtract) for updating VIVO with the values in the spreadsheet. The Pump get method, uses the definition file and the data in VIVO to create a spreadsheet. Schematically:, , update method: Spreadsheet Data => The Pump => VIVO RDF => VIVO, + definition file, , get method: Spreadsheet Data <= The Pump <= VIVO RDF <= VIVO, + definition file, , The same definition file can be used for get and update, and the spreadsheet resulting from a get can be used in a, subsequent update, providing a "round trip" capability. That is, you can "get" data, improve it manually or, programmatically, and use it to "update" VIVO., , ## Work in Progress, , The Pump is under active development, and beta testing at the University of Florida. , , See [Issues](https://github.com/mconlon17/vivo-pump/issues) for work in progress, features to be added, and bugs to be , addressed., , , ## Testing the Pump, , \* [Test Case Overview](https://github.com/mconlon17/vivo-pump/wiki/test-case-overview), \* [Simple VIVO Tests](https://github.com/mconlon17/vivo-pump/wiki/simple-vivo-test-cases), , ## The Pump API, , [The Pump API](https://github.com/mconlon17/vivo-pump/wiki/the-pump-api)

# Introduction to the Pump

## # Intro to the Pump, , The VIVO Pump (under development), provides a means for managing VIVO data using spreadsheets. Rows and columns in , spreadsheets represent data in VIVO. The Pump update method takes spreadsheet data, and a definition file, and , returns VIVO RDF (add and subtract) for updating VIVO with the values in the spreadsheet. The Pump get methods, uses the definition file and the data in VIVO to create a spreadsheet. Schematically:, , Spreadsheet Data => The Pump => VIVO RDF (the update method), + definition file, , Spreadsheet Data <= The Pump <= VIVO RDF (the get method), + definition file, , The same definition file can be used for get and update, and the spreadsheet resulting from a get can be used in a, subsequent update, providing a "round trip" capability. That is, you can get" data, improve it manually or, programmatically, and use it to "update" VIVO., , ## Work in Progress, , The Pump is under active development, and beta testing at the University of Florida. , , See [Issues](https://github.com/mconlon17/vivo-pump/issues) for work in progress, features to be added, and bugs to be , addressed., , , ## Testing the Pump, , \* [Test Case Overview](https://github.com/mconlon17/vivo-pump/wiki/test-case-overview), \* [Simple VIVO Tests](https://github.com/mconlon17/vivo-pump/wiki/simple-vivo-tests), , ## The Pump API, , [The Pump API](https://github.com/mconlon17/vivo-pump/wiki/the-pump-api),

# Simple VIVO

## *a command line tool for managing data in VIVO using spreadsheets*, , ## Could VIVO ever be "simple"?, , VIVO provides an integrated view of the scholarly work of your organization. The scholarly work of your organization is complex -- faculty, research staff, their activities and accomplishments. And these are connected to each other and to institutions, journals, dates, and concepts. , , Let's think about VIVO as being about people. In the figure below, we see people in the center of the diagram. Things "below" the people are things that exist in the academic environment and can exist in your VIVO without reference to people. As we think about managing data in VIVO, and in building a VIVO, these things represent the place to start. We can put these things in VIVO and expect them to be there when we begin to put in people and journals., , The things at the top of the figure are the details of the scholarly record of people. They represent the things that will appear on a person's curriculum vitae., , , , VIVO is great at representing all these things and more. But to keep it "simple," we will focus on the domains in the diagram., , ## The Basic Idea, , What if we could manage the data in VIVO using spreadsheets?, , That's the basic idea behind "Simple VIVO" a tool for using the VIVO Pump to take spreadsheet data and put into VIVO (called an update) and to get data from VIVO and put it in a spreadsheet (called a get)., , Spreadsheets have rows and columns. The rows in the spreadsheet will correspond to "things" in VIVO. Depending on the scenario, your spreadsheet might contain people, publications, grants, courses or other kinds of things., , The columns in your spreadsheet will correspond to attributes of things in VIVO. So, for example, if you are working with people, your columns might contain attributes such as "name" and "phone number"., , For each row in each column, your spreadsheet has a cell. Cells correspond to the values that will be in VIVO. In Simple VIVO you can have one of three different things in each cell:, , 1. Your cell contains a value. That value will be used to update VIVO., 1. Your cell contains the word "None." None is a special word in Simple VIVO. It means that VIVO will be updated to remove whatever value is currently in VIVO. If you have None in a cell for the phone number of a person, then, whatever phone number is currently in VIVO will be removed., 1. Your cell is empty or blank. Blank is also a special value for Simple VIVO. It means "do nothing." Whatever value VIVO might have for the attribute is left unchanged., , Using these three things, you can manage data in your VIVO using a spreadsheet. , , ## An Example, , Suppose we have data in our VIVO on our faculty. Three of the faculty members are shown below:, , URI | Name | Phone Number, ---|---|---, <http://my.school.edu/individual/n123321> | Jones, Catherine | 345-8999, <http://my.school.edu/individual/n467823> | Pinckey, William | (404) 345-8991, <http://my.school.edu/individual/n858832> | Hernandez, Juan |, , In reviewing this data, we find we would like to make the following changes:, , 1. Catherine's phone number should have an area code, 1. William's last name is misspelled and should not have a "c", 1. William has left the university and his phone number should be removed, 1. Juan's phone number should be provided., , We prepare an update set of data containing the following entries:, , URI | Name | Phone Number, ---|---|---, <http://my.school.edu/individual/n123321> || (404) 345-8999, <http://my.school.edu/individual/n467823> | Pinkey, William | None, <http://my.school.edu/individual/n858832> | Hernandez, Juan | (404) 345-8993, , This update spreadsheet can be considered a set of instructions to Simple VIVO. We can read it say the following:, , 1. For the individual with URI <http://my.school.edu/individual/n123321>, leave the name unchanged (it is blank and blank means "do nothing") and change the phone number to (404) 345-8999., 1. For the individual with URI <http://my.school.edu/individual/n467823>, change the name as shown and remove the phone number (None means remove whatever value you find), 1. For the individual with URI <http://my.school.edu/individual/n858832>, change the name as shown (Simple VIVO will notice that the name you gave is the name that is already in VIVO and so no change will actually be made. This is very handy. It means you can get data from VIVO and change only the items that need improving. All the other items can be left as they came from VIVO. The update will leave them untouched) and change the phone number to (404) 345-8993., , When the update is performed, the data in VIVO will look like:, , URI | Name | Phone Number, ---|---|---, <http://my.school.edu/individual/n123321> | Jones, Catherine | (404) 345-8999, <http://my.school.edu/individual/n467823> | Pinkey, William |, <http://my.school.edu/individual/n858832> | Hernandez, Juan | (404) 345-8993, , Everything looks good. The names and phone numbers are all correct., , ## Data In, , Getting data into VIVO is simple using Simple VIVO. You put the data you would like to add in a spreadsheet and save it as a "CSV File" -- a comma separated value file. You can also use a tab separated file (TSV) or use any, delimiter of your choosing (specified in the configuration parameters of Simple VIVO. For now, let's assume your, configuration is using a comma separated file., , You tell Simple VIVO the name of your spreadsheet and the name of your definition file. That's it. Simple VIVO, will add the data in your spreadsheet to the data in VIVO., , Let's say you wanted to add some new publications to your VIVO and your publications are stored in a spreadsheet called pubs.csv. Let's further assume that you have a definition file that defines your spreadsheet -- many such definition files are provided with Simple VIVO -- see [[Provided Data Scenarios|simple-vivo#provided-data-scenarios]] below. You would use:, , python sv.py -a update -defn pubs\_def.json -src pubs.csv, , The data in pubs.csv will now be in VIVO. Simple., , Let's look at the items on the command line to see what each signifies:, , python sv.py is the way we tell our system to run the Simple VIVO program. Your system administrator can confirm that this will work on your system., , -a update tells Simple VIVO that you want to update VIVO, using data from a source spreadsheet according to a definition., , -defn pubs\_def.json tells Simple VIVO that you have a definition file called pubs\_def.json that you would like to use to define the columns in your spreadsheet for the purpose of updating VIVO., , -src pubs.csv tells Simple VIVO that you have a data file called pubs.csv that will provide values to be added to VIVO., , ## Data Out, , Getting data out of VIVO is simple using Simple VIVO. You specify the name of a definition file and the name of the file you want to store the data in. Simple VIVO uses the definition file to access your VIVO, retrieve the data,, and return a spreadsheet., , To retrieve a list of the faculty at your institution, you would use:, , python sv.py -a get -defn faculty\_def.json -src faculty.csv, , The result is a new file called faculty.csv that contains the data for the faculty as defined by faculty\_def.json, , Each of the items on the command line are described below:, , python sv.py is the way in which a python program such as sv.py is executed on your system. Your system administrator can determine if this is how you will start Simple VIVO on your system., , -a get indicates to Simple VIVO that you want to get data from VIVO. -a stands for "action". So you are requesting the get action., , -defn faculty\_def.json indicates to Simple VIVO that you will be using the faculty\_def.json definition file. This file must be available on your system. If Simple VIVO can not find the definition file you specify, it will, provide an error message to that effect., , -src faculty.csv indicates to Simple VIVO that the output spreadsheet will be called faculty.csv. Once Simple VIVO runs, you will have a new file called faculty.csv. You will be able to open the spreadsheet in Excel, Numbers, a text editor or other program., , ## Round Tripping, , Getting data into VIVO and getting data out look very similar. The only difference is the action. The same definition file is used to get the data and to update the data. This insures that a spreadsheet that was produced by a "get" can be used by an "update". , , It is common to get data from VIVO, improve it in some way, and then provide it back to VIVO. Suppose we were managing the organizations of our institution. We may have noticed that several of the institutions in our VIVO were missing URLs to their web pages, or missing phone numbers. We could use the steps below to improve the organizational data in our VIVO:, , 1. Get the organizational data from VIVO:, , python sv.py -a get -defn org\_def.json -src org.csv, , 1. Open orgs.csv in Excel or your favorite spreadsheet editor and make changes as needed -- provide missing phone numbers, URLs or making other changes such as correcting spelling in names or organizations., 1. Put the improved data back in VIVO:, , python sv.py -a update -defn org\_def.json -src org.csv, , These steps are very common -- we get data from VIVO, improve it, and put the improved data back in VIVO., , ## Provided Data Scenarios, , Follow one of the provided scenarios to manage data in your VIVO. The most common scenarios for managing data regarding the scholarship of your institution are provided., , \* [Dates](https://github.com/mconlon17/vivo-pump/wiki/dates), \* [Organizations](https://github.com/mconlon17/vivo-pump/wiki/organizations), \* [Concepts](https://github.com/mconlon17/vivo-pump/wiki/concepts), \* [People](https://github.com/mconlon17/vivo-pump/wiki/people), \* [Positions](https://github.com/mconlon17/vivo-pump/wiki/positions), \* [Educational Background](https://github.com/mconlon17/vivo-pump/wiki/educational-background), \* [Awards and Honors](https://github.com/mconlon17/vivo-pump/wiki/awards-and-honors), \* [Publications](https://github.com/mconlon17/vivo-pump/wiki/publications), \* [Journals](https://github.com/mconlon17/vivo-pump/wiki/journals), \* [Publishers](https://github.com/mconlon17/vivo-pump/wiki/publishers), \* [Authors](https://github.com/mconlon17/vivo-pump/wiki/authors), \* [Grants](https://github.com/mconlon17/vivo-pump/wiki/grants), \* [Sponsors](https://github.com/mconlon17/vivo-pump/wiki/sponsors), \* [Teaching](https://github.com/mconlon17/vivo-pump/wiki/teaching), \* [Mentoring](https://github.com/mconlon17/vivo-pump/wiki/mentoring), \* [Service](https://github.com/mconlon17/vivo-pump/wiki/service), , ## Adding Scenarios, , Adding new scenarios requires a knowledge of the VIVO ontologies, as well as knowledge of the [[Pump Definition File|the-pump-definition-file]]. Studying the provided scenarios and the descriptions provided in this wiki will get you started. You will find support on-line on the VIVO email lists and in the [VIVO Wiki](http://wiki.duraspace.org/display/VIVO).