# REDCap and CDW Data Pipeline

**Setup**

There are three pieces of setup required before the pipeline code will work:

1. Connect to the VPN if you are offsite. Then ensure you can load [\\nida-filestore\NRB-CPN-CDWData](file:///\\nida-filestore\NRB-CPN-CDWData). This is the network folder where all CDW data is dumped.
2. Request an API token in REDCap. This will allow you to import data directly to R.
   1. In REDCap, select the API link near the bottom of the Applications section in the left-hand navigation. If API is not present, please let me (Garrick) know.
   2. Click the “Request API Token” button.
   3. Within about a day, NIDDK should issue you an API token. It will be visible by navigating back to the API section of the project.
3. Add the API token to R. There are instructions [here](https://raymondbalise.github.io/tidyREDCap/articles/useAPI.html). The short version is:
   1. Run the following:
      1. install.packages("remotes")
      2. remotes::install\_cran("usethis")
      3. usethis::edit\_r\_environ()
   2. Add the following line to the file, replacing PROJ with a short version of the project name, and replacing API\_TOKEN with the actual token:
      1. PROJ\_key=“API\_TOKEN”

**Using the pipeline**

The pipeline handles importing and merging of all data. Follow these steps:

1. Add pipeline.R to your project folder. Then source the file by typing   
   source('pipeline.R')
2. Load the REDCap data from completed instruments:   
   all\_redcap\_data = load\_redcap\_data(Sys.getenv('PROJ\_key'), first\_record\_id = 100, record\_id\_name = 'record\_id', arc\_num\_name = 'arc')
   1. Replace PROJ\_key with the name you used in Step 3.b. in Setup.
   2. Check REDCap for the settings to the other parameters. For example, I believe NIC-MDD starts counting record IDs at 100, while Orexin starts at 1. I also know the projects differ on the record\_id\_name (the name of the first field in the first instrument) and the ARC # field name in REDCap.
   3. Other options not shown:
      1. keep\_individual\_instruments – Set to TRUE if you would like the store each instrument as an individual data frame in R. Default is FALSE. Either way, the function will give you a full data frame of all completed instruments merged together.
      2. redcap\_server – Set to ‘redcap’ if you are not using redcapsurvey
3. Load the CDW data:  
   protocol\_name = REPLACEME  
   all\_cdw = load\_cdw\_data(protocol\_name, '//nida-filestore/NRB-CPN-CDWData')
   1. protocol\_name refers to the beginning part of the files dumped in the CDW folder – everything up to the underscore. So, e.g., 922MDD(NIC-MDD) is a protocol\_name, and so is 1066Smokers(Orexin)
   2. Other options not shown:
      1. keep\_individual\_instruments – Set to FALSE if you do *not* want the individual CDW data types kept as separate instruments. Either way, the function will return a full data frame of all CDW data in long form.
4. Combine the data:  
   all\_data = combine\_redcap\_and\_cdw\_data()

**Details**

* The REDCap data returned by load\_redcap\_data() contains one row per participant. Only completed instruments are included.
* load\_redcap\_data() also creates two global variables: id\_mapping and id\_mapping\_sensitive. These are used to merge REDCap and CDW data. The final data frame given by combine\_redcap\_and\_cdw\_data()will use *only* REDCap IDs, including for CDW data.
* The data frame produced by load\_cdw\_data() is an attempt at reconciling slightly different formats for each of the four CDW data tables. If anything is missing or you have concerns here, reference one of the raw CDW tables. They are in variables prefixed with cdw\_. Please also send feedback if you hit issues!
* The final data frame is in long format. This is the most versatile way of dealing with the mixture of CDW and REDCap data. However, if it causes problems, please let me (Garrick) know.