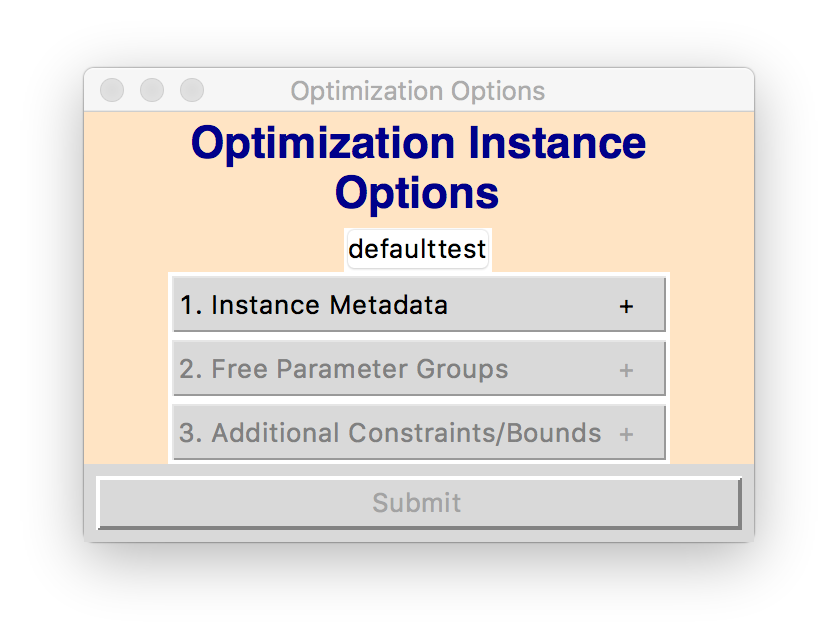
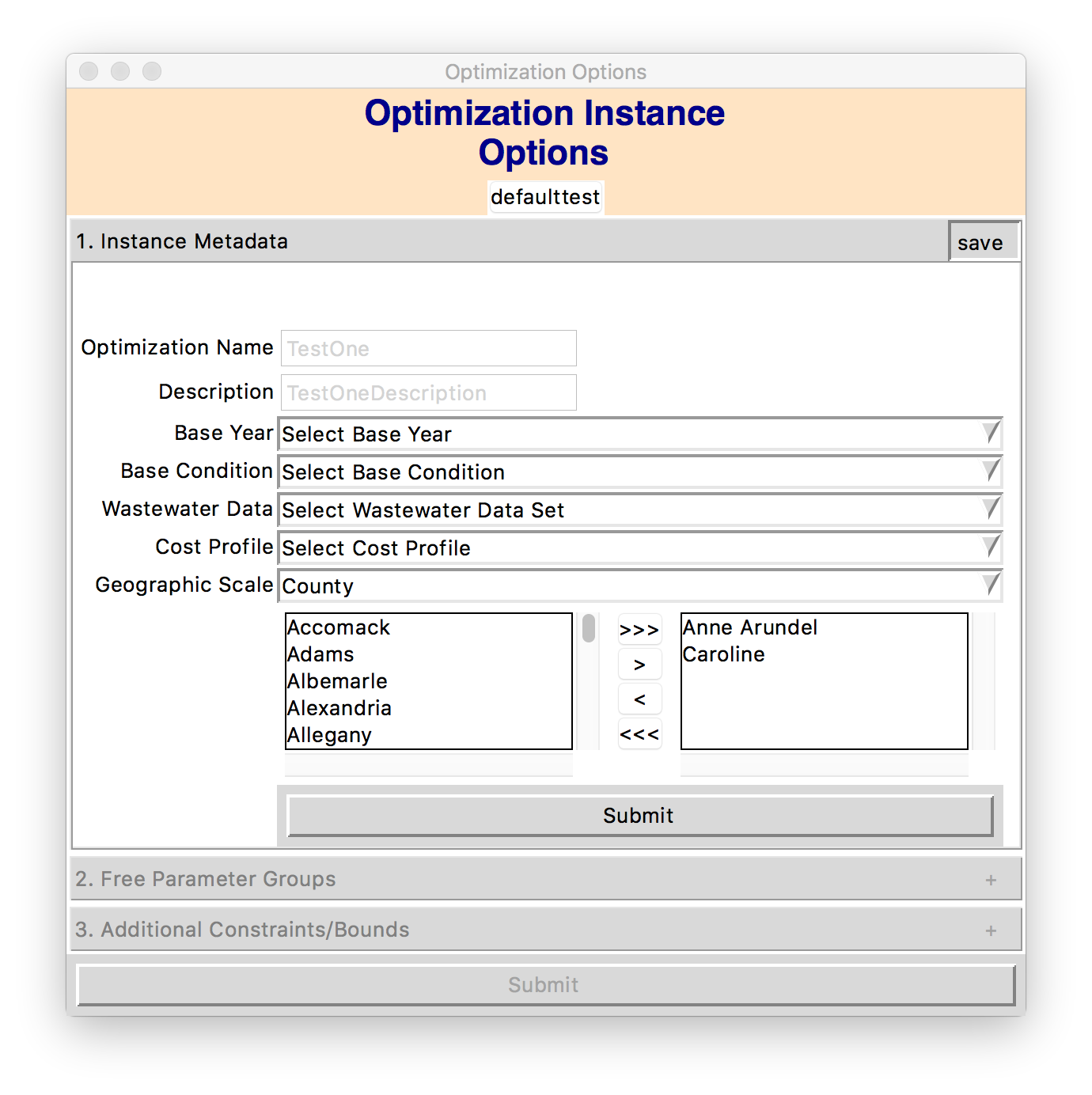
***OptSandbox description***

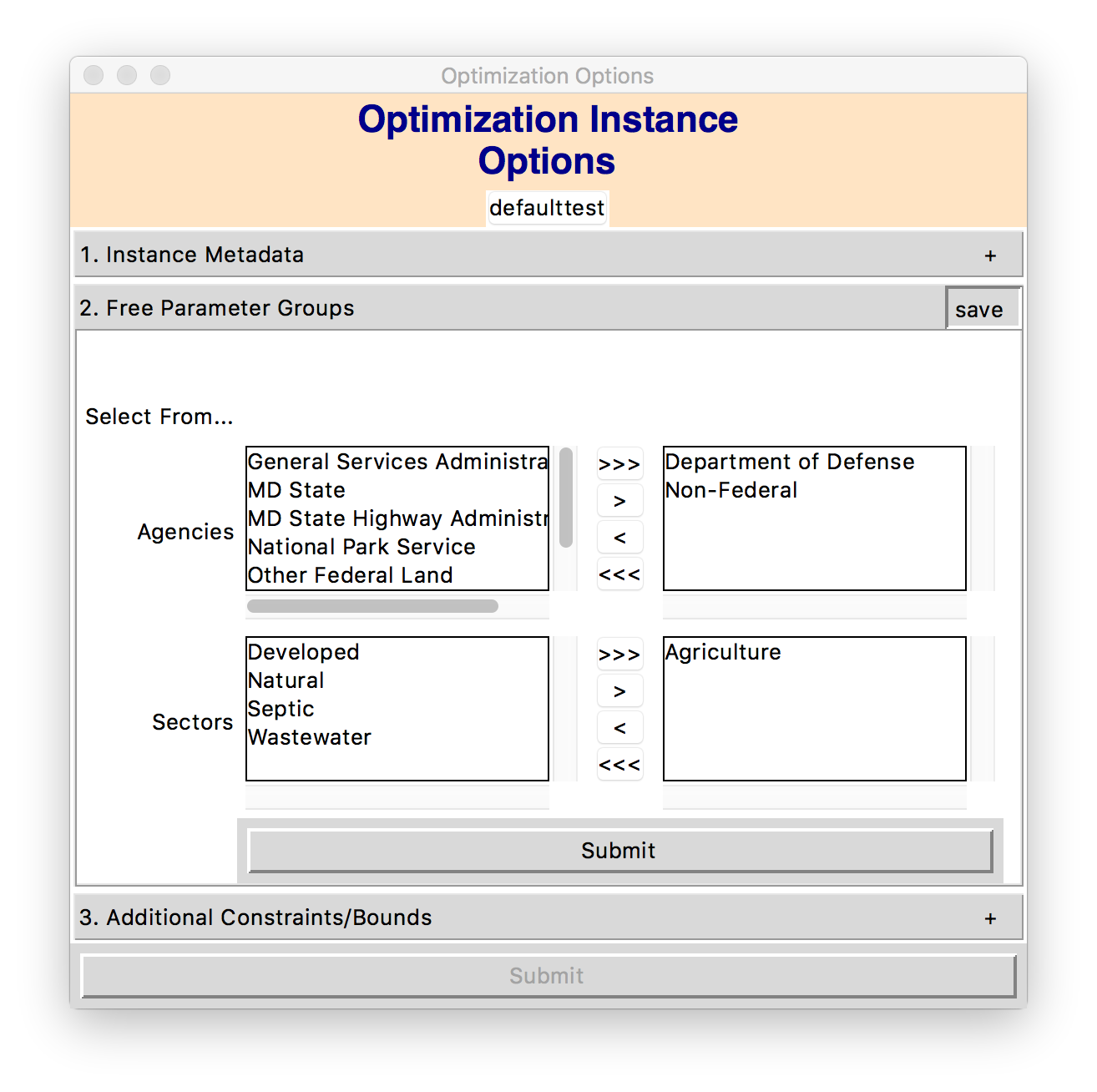
1. An OptInstance object is created with blank attributes.
   1. Source Data and Base Condition tables are read in. [TblLoader]
   2. TableQuery objects are created to provide hooks to the source and base tables
2. A GUI window opens with three collapsible/toggle frames populated with configuration options that will determine the parameter space of the OptInstance.



* 1. Frame #1: Metadata
     1. Has entries for [*Instance Name, Instance Description, Base Year, Base Condition, Wastewater Data, Cost Profile, Geographic Scale, Geographic Area*]



* + 1. When frame is closed:
       1. metadata options are saved to OptInstance
       2. from the metadata options, a table that contains all of the lrsegs included in this OptInstance is generated and saved to OptInstance
       3. the next frame, ‘free parameter groups’, is ungreyed and has its options loaded
          1. the base condition data is queried to determine the agencies present in the included lrsegs
          2. all sector names are retrieved.
  1. Frame #2: Free parameter groups
     1. Has entries for [*Agencies, Sectors*]



* + 1. When frame is closed:
       1. free parameter groups are saved to OptInstance
       2. the next frame, ‘additional constraints’, is ungreyed and has its options loaded
  1. Frame #3: Additional constraints
     1. Has entries for [TBD]
     2. When frame is closed:
        1. The submit button is ungreyed

1. Empty parameter matrices (3) are created with Application and BMP coordinates
   1. Load source tables are queried to get “ya’ad” tables (i.e. the targets of BMP application): geographies, agencies, load sources, animal names, FIPS, FIPSFrom, FIPSTo and their units and amounts.
   2. The ya’ad tables are used as the row\_indices for new empty parameter matrices. Columns are specified by the list of all BMPs
2. Coordinates (Ya’ad, BMP) in each parameter matrix are marked with NaNs or ones to indicate eligibility for implementation (in other words, identifying which dimensions of the parameter space to keep and which to discard)
3. **Eligible dimensions are iterated through to identify Hard Upper Bounds for each dimension.**
4. Reformat Possibilities Matrix to Multi-index vector
5. Write Multi-index vector to tab-delimited text file