

Proposed GUI Changes to support DataMgr 3.0

Draft: August 19, 2014

Some of the changes distinguishing DataMgr 3.0 from the previous DataMgr's are the following:

- There can be different grids and/or extents for different variables.
- Variables can be constant in time
- Coordinate variables are in the VDC.
- Multiple DataMgr's can be used at the same time, with potential name-collisions.

To enable the user to make use of these features, the following changes to the GUI are proposed:

1. In the Data Load menu, support a menu item "Merge Data into current session" which would allow the user to specify an additional DataMgr to be loaded without deleting the existing DataMgr's. [Note: we could also support "import additional data into a session" when we are able to create 3.0 DataMgr's from imported data]. The paths and filenames associated with the various DataMgr's will be remembered in the session. When the session in the VAPOR GUI already indicates multiple DataMgr's (i.e. a session file was read which was created from a session with multiple DataMgr's), the prompt in the data load menu will allow the user to select the matching DataMgr, for example by "Merge Data 1 into current session" will provide a file selector with the name and path to the first DataMgr file.
2. When a name collision is detected in the loading of a DataMgr, this will be reported in a warning message to the user, and the names of the colliding variables will be modified and presented to the user with an appropriate index appended; e.g. U_1 and U_3 would refer to variables U in the first and third DataMgr that is loaded.
3. Variables that are constant in time will be available at all timesteps.
4. Coordinate variables will not be exposed in the GUI.
5. In every renderer panel that currently has a variable selector, the compression and refinement level selectors will only show the settings that are available for all the selected variables in the renderer panel. For example the Flow variables must all use the same refinement and compression levels, even if they come from different DataMgr's. [This restriction is not necessary if it is deemed problematic].
6. In some panels that allow selection of multiple variables there may be a requirement that the multiple variables be on the same grid (e.g. we may require isosurface color-mapping variables to be on the same grid as the isosurface variable, or barbs that are aligned to the grid must have their 3 field variables on the same grid). To support this (and also for Python) it will be useful to have a method on the DataMgr3.0 to test whether two variables are on the same grid.

7. Each renderer that currently handles 3D regions (DVR, Isosurface, Flow) will have its own 3D Box control to specify the 3D region that will be used in rendering. This will enable, for example, two DVR's to simultaneously render data from non-intersecting boxes. The Region panel will initially indicate the maximum extent of all existing variables from all DataMgr's, but can be reduced to limit rendering to a smaller region of data. The Boxes in the DVR, Iso, and Flow panels will default to the extents of the variables used in the renderer, but can be modified to restrict rendering to a smaller box.
8. The Python editor will require that all input variables be on the same grid (although not necessarily from the same DataMgr). When one input variable is selected, only other variables on the same grid will be selectable. The derived output variables will be on the same grid as the input variables.
9. Session files will retain DataMgr index with each variable name so that loading sessions can properly match variable names that result from name collisions.