



11-2-2015

# UX Report: DataONE Toolbox for MatLab User- Experience Test Results

UAWG

*University of Tennessee, Knoxville*

Follow this and additional works at: [https://trace.tennessee.edu/utk\\_dataone](https://trace.tennessee.edu/utk_dataone)



Part of the [Library and Information Science Commons](#)

---

## Recommended Citation

UAWG, "UX Report: DataONE Toolbox for MatLab User-Experience Test Results" (2015). *DataONE Sociocultural and Usability & Assessment Working Groups*.

[https://trace.tennessee.edu/utk\\_dataone/163](https://trace.tennessee.edu/utk_dataone/163)

This Creative Written Work is brought to you for free and open access by the Communication and Information at Trace: Tennessee Research and Creative Exchange. It has been accepted for inclusion in DataONE Sociocultural and Usability & Assessment Working Groups by an authorized administrator of Trace: Tennessee Research and Creative Exchange. For more information, please contact [trace@utk.edu](mailto:trace@utk.edu).

DataONE Toolbox for MatLab User-Experience Test Results  
Usability and Assessment WorkGroup, DataONE  
November 2, 2015

The DataONE Usability and Assessment Working Group with members of the Cyberinfrastructure Working Group conducted a user-experience study of the DataONE toolbox for MatLab during the DataONE All-Hands Meeting in October 2015.

## **METHODOLOGY**

### **Protocol**

The user-experience testing of the DataONE toolbox for MatLab is not intended for research but solely as product development. Testing was conducted on October 27, 2015 during the afternoon session of the DataONE All-Hands Meeting. The testing was conducted by Rachel Volentine, CCI-ORNL User-Experience Lab Manager at the University of Tennessee, Knoxville, and members of the Cyberinfrastructure working group, including Matt Jones, Chris Jones, Lauren Walker, and Yang Cao.

Chris Jones lead two participants through a walkthrough of the MatLab tool while members of the U&A and CI Working Group observed and took notes. Each participant had a computer to complete each step as instructed. Participants were asked to verbalize their thoughts during the walkthrough. Participants completed a survey at the end of the walk-through to gather more feedback into the design. The walk-through document and the survey can be found at the end of the report.

### **Participants**

Two participants completed the testing. One participant uses MatLab less than once a month, while the other participant uses MatLab daily. They both considered themselves to be “advanced, daily in depth use” or “professional, expert use and mentoring”.

## **TASK RESULTS**

Below is a discussion of the U&A and CI Working Group notes during the walkthrough and the survey results.

### ***Installing the toolbox***

The participants thought installing the MatLab toolbox was easy, though they expressed surprise that they had to restart MatLab after installation.

### ***Creating objects***

There were no issues with creating RunManager object.

### ***Recording a script***

In the survey, participants thought the purpose of recording a script was to “record useful, potentially repeatable and traceable provenance for a single run,” and to “track provenance information and manage multiple runs of workflow.

### ***Listing completed runs***

There were some confusion about `list.run()` and use of semicolons to display a nicely formatted version. One participant wanted the option of using a wildcard search. There was some confusion on the names of tags. One participant did three runs but confused because two of the runs are the same. He thought that since they had the same tag that he had rewritten a pervious run. He thought new users would expect tags to be different. Run number was also not a clear term.

Participants were split on whether the filter parameters were useful. They thought the filters were a good start but more filters are needed, for example a wildcard option. One participant also had some confusion with the tags because he thought of a tag as a single discreet word used to convey a keyword type of concept rather than a descriptive string of any length and complexity as used by the DataONE toolbox.

### ***Viewing selected run***

There were some issues with how the parameters are passed but this is how MatLab does it, so DataONE toolbox should stay the same.

Currently, for every p/v pair it generates an output structure that you could capture, and one participant asked if there was a way to suppress the screen dump and just capture the output of the view data structure? `[x,y,z]=mgr.view(...)` should be `z=mgr.view(...)`.

Participants expressed interest for a quiet parameter.

One participant thought the run details are understandable if you understand MatLab conventions and syntax, but he thought it might lead to confusion for users if they switched to Python or R. He thought this issue could be further investigated. The other participant thought the run details were kind of understandable but may not be the best way to do so.

Both participants thought the soil mapping script was more understandable with the comments, but one participant worried that too many comments could be problematic especially if they are basically duplications of information clearly stated in the script.

One participant had 25 items used by run and saw `modifiedTime` for the files, but was not sure whose time it was? He wasn't sure who modified what. He thought time stamp needed to be improved because it could seem confusing, like som ething was overwritten. There was some discussion about whether time was useful at all.

Participants were not sure if package ID is a term other people would know. Suggested that Unique passage ID may be a better name.

### ***YesWorkflow Diagram***

When they tried to create a YesWorkflow diagram there were some issues so they all looked at Chris' backup diagram. They liked the diagram, and wanted a way to bring it into something else, like a PDF. They would like to have multiple options. Overall, they thought the diagram was a good visual aid for how the code is working and what it is doing. One participant thought a legend explaining the color/shape categorization could help improve the diagram, and one participant thought it would be nice to have some display format where one could move the boxes around (e.g., Kepler).

The two participants were split on whether the YesWorkflow would be useful in the web display. The participant who thought it would be useful was worried that the execution would be poor and create more clutter. If the YesWorkflow is added to web display may want to hide the diagram unless the user chooses to view it.

### ***Deleting Run***

Overall, deleting a run was understandable. They wanted to be able to delete more than one run at a time. Delete by tag, package ID, or run number.

### ***Viewing and Modifying Metadata***

Overall, participants thought viewing and modifying metadata for a run was understandable, thought one participant called it clunky and would like to create a script that automates a lot of these low-level set calls. One participant wondered if there was any way to see the fields that can be populated. He was happy to learn that there was an option to do so. They also liked that you did not have to fill in all the fields right away and that a placeholder would be left in the other fields. Participants wanted a 'preflight' check before publishing, which is currently on the TODO list.

One participant ran into issues when he ran the set commands but then ran publish with an error, and now the `science_metadata_config` object is empty. Need to enable `runNumber` in `mgr.getMetadata()` rather than the UUID because it is too hard to type (and copy/paste is not working).

Both participants thought a template metadata file was the most convenient way to provide metadata for your runs.

### ***Publishing Run***

Both participants thought it was useful to publish a run, especially since publishing seems to increase data sharing which seems to be in line with DataONE's mission.

### ***Viewing the data package on the web***

Overall, participants liked the clean and simple design, and they stressed that they would prefer maintaining the simple/clean layout rather than adding information that clutters the UI. They thought

the provenance display was one of the best displays they have seen. They liked the plain English description in the provenance.

They liked the simple provenance layout but wanted some indication that they were additional ancestors (e.g., arrow or symbol).

When you click on the ancestor you see “view”, which the participants expected to mean you would see the R script (or other object). We recommend changing the term to “preview” or other term to mean that you see metadata. They liked the idea of getting a preview of the ancestor data, which we do not currently have for every ancestor object.

The amount of scrolling was an issue for the participants. They thought it would be easy to get lost on the page. They wished for a “back to top” or “jump” arrow. They also thought a tab system would help make it more obvious what information was available on the page. However, they did not want the system to become cluttered.

They thought the date the data was created was important and wanted it to be more prominent, but they did not think the upload date or date of DOI was important.

#### **FINAL COMMENTS**

Both participants would use this system frequently (Agree strongly).

Neither participant thought they would need the support of a technical person.

One participant was neutral about if the various functions in the system were well integrated, while one strongly agreed.

Neither participant strongly agreed that people would learn to use the system very quickly. There could be room for improvement to make the toolbox more understandable for “average” MatLab users rather than experts.