

# Tale Workspaces

## 1. Background

With release v0.4, there is still significant confusion about the Tale creation process and the role of the "Home" folder. During both the compose and publish processes, selecting files from "Home" raises questions about where they should end up in the Tale or published package. The question was raised about whether we should consider removing the Home folder altogether, which led to the discussion of a "Project" folder that would be associated with each Tale during creation. Each Tale would have a dedicated folder where all work would be done. Kacper noted that this was in fact the design of the original Tale "Workspace" folder.

### 1.1. "Workspaces"

There is an existing implementation of Workspaces on the backend that were part of the original UI. This document revises and expands on the existing design.

### 1.2. "Home" folder

For some communities, the Home folder remains an important feature. At the Workshop, a researcher from High Energy Physics (HEP) indicated that she'd love to have her RC files available on the system. It has been noted that using RC files in containers automatically is problematic, but advanced users are still able to access files within the container manually.

### 1.3. Copy on Launch (was forking)

Workspaces are also related to the discussion about how to provide read/write access to a tale at launch. This led to the idea of "copy on launch" described in 3.5 below.

### 1.4. Custom environments

Workspaces are related to the [customization of environments](#).

### 1.5. Remixing

Workspaces are also related to remixing, which is defined in a [separate document](#).

### 1.6. Sharing

Workspaces are also related to sharing, which is defined TBD.

## 2. Proposed feature: Tale Workspaces

- During Tale creation (or launch), a Tale Workspace folder is created for each Tale. This is where all files associated with a given Tale must be stored.
- The Tale Workspace folder can be shared with other users in the system.
- The Tale Workspace folder will be available alongside "home" and "data" in container
- The Home directory would be used only for data transfer and to store private information -- utilities, keys, secrets, etc.
- Tale Workspaces may be seen as the equivalent of Binder repositories. If we adopt the repo2docker model, the Workspace will contain the buildpack files (e.g., apt.txt, requirements.txt)
- Tale Workspaces may also contain a copy of the latest version of the tale.yaml or equivalent to reflect changes in data, environment, etc.
- When launching a tale with read-only access, a read-write copy of the tale including the workspace will be made for the user.

### 3. Mockups

The following mockups address the Tale Workspaces along with proposed changes to improve usability related to the selection of data.

#### 3.1. Selecting Data

This section defines a new workflow and mockups for selecting data to be included in a Tale.

With v0.4, selecting data for a tale happens via the following steps:

- The user uploads data to their home directory, then selects it during the compose (or publish) process
- The user registers data via the "Register Data" dialog, then selects it during the compose process or upcoming feature to add data to a running tale.
- Note: In v0.4 this time it is not possible for a user to select from data already registered with the system, such as data associated with another tale.

We propose refactoring the "Register Data" dialog to instead allow the user to "Select Data" for a tale. The "Select Data" dialog would enable the user to select from either web (DOI, HTTP), local catalog sources (i.e., publicly available data in the WT system), or other Tale Workspaces. Mockups are presented in figures 3.1.1 - 3.1.3.

Web  
Catalog

Reference URI @

Search

The URL or DOI of the data object. Data packages can be imported into Whole Tale from DataONE and select Globus repositories. For a full list of DataONE member nodes and supported Globus repositories, visit the [data registration guide](#).

Folder Name

DOI:

Dataskit:

Repository:

Size:

Cancel

Select

Figure 3.1.1 shows is a mockup for selecting data from web (DOI or URL) sources. This combines the current register window with selection for inclusion in a tale. The user enters the URI or DOI for a dataset.

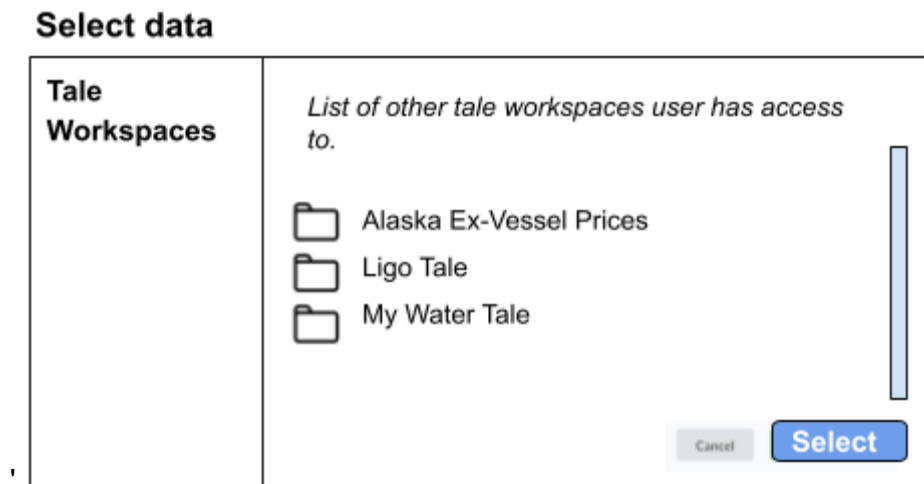
- From either the manage or compose pages (see below), the user selects the "Select Data" button and is presented with the "Select Data" dialog
- The user selects the "Web" tab and enters the DOI or URL (3.1.1)
  - Search displays basic information about the dataset
  - "Select" registers the dataset and adds it to the current tale.
- TBD:
  - Some datasets may take a very long time to register
  - Ideally, the user could select individual files or folders from within the dataset

### 3.1.2 Select data - Catalog

Figure 3.1.2 shows is a mockup for selecting data from data in the WT catalog.

Example workflow:

- From either the manage or compose pages (see below), the user selects the "Select Data" button and is presented with the "Select Data" dialog
- The user selects the "Catalog" tab and can select from any dataset already registered in the system. (3.1.1)
  - They can optionally pick individual files from the selected dataset or tale (ala filepicker)



### 3.1.3 Select data - workspaces

Figure 3.1.3 shows is a mockup for selecting data from tale workspaces

Example workflow:

- On the manage page, the user selects the "+" button in the Workspace panel, which displays a sub-icon to enable selecting data from other tale workspaces.
  - They can optionally pick individual files or folders (ala filepicker)

The Select Data panel would be integrated into the compose, run, and manage pages.

Compose:

- The user selects "Compose" (figure 3.2.1) to compose a new tale.
- The user selects the "Select Data" button and is presented with the "Select Data" dialog
- Selected files/folders are displayed under the input data section

Run:

- The user selects "Run" (figure 3.3.1) and has an active tale context
- The user selects the "+" button in the Data panel, which displays the "Select Data" dialog. The user can select from either Web or Catalog resources as above.

Manage:

- The user selects "Manage" (figure 3.4.1) and has an active tale context
- Data:
  - The user selects the "+" button in the Data panel, which displays the "Select Data" dialog. The user can select from either Web or Catalog resources as above.
- Workspace: The user can select data from other tale workspaces
  - The user selects the "+" button in the Workspace panel, which displays a sub-icon to enable selecting data from other tale workspaces.

### 3.2. Compose

We propose changing the compose page to use the "Select Data" dialog..

Example workflow:

- User selects the 'Compose' tab to create a new tale
  - Enters tale name
  - Selects compute environment
- Optionally, user can "Select Data..." which opens the "Select data" dialog (3.1)
  - Selected items are displayed under "Input data"
  - User can remove selected data before tale creation
- The user selects "Create Tale" which creates the tale object and tale workspace, starts the tale instance, and takes the user to the Run page for the new tale.

,

#### 3.2.1 Compose window

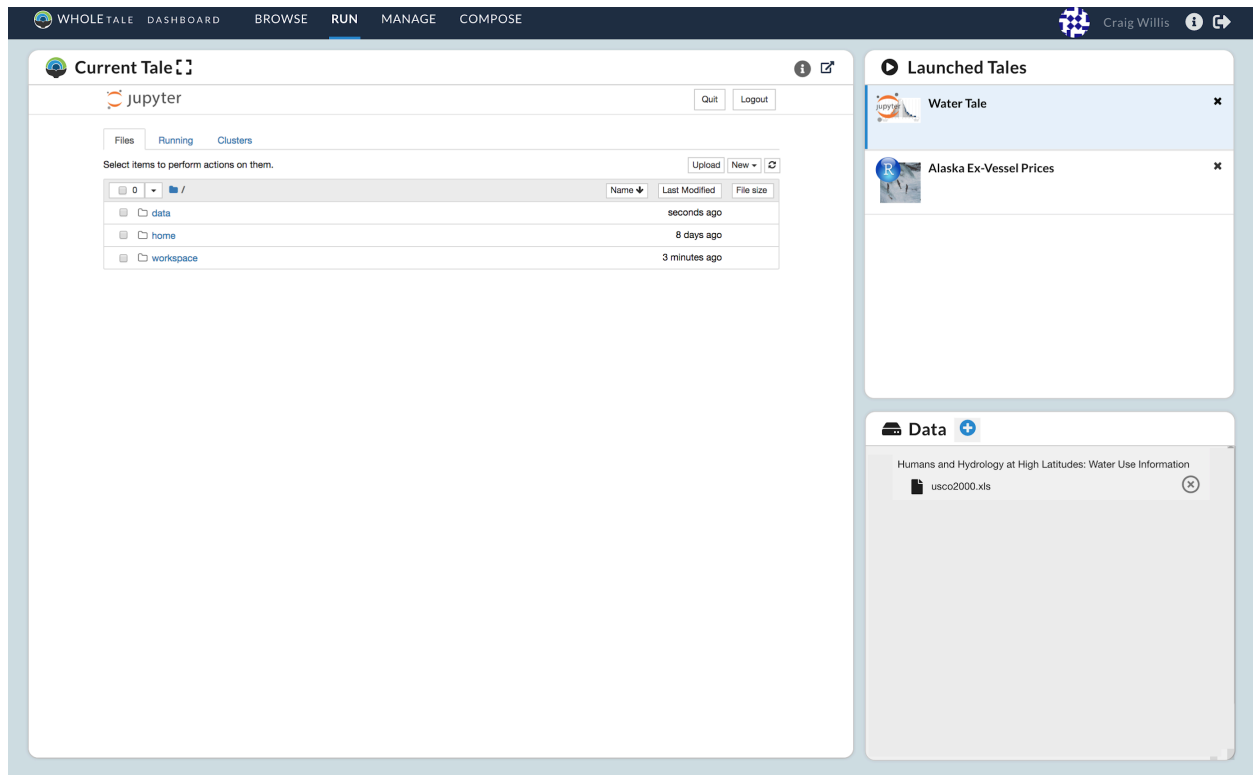
### 3.3. Run

We propose changing the Run page Data panel to use the Select Data dialog above. The Data panel displays the data currently associated with the tale.

Example workflow:

- The user selects a tale from the "Launched Tales" panel.
- The tale is displayed in the "Current Tale" panel

- The data associated with the tale is displayed in the Data panel.
  - The user can remove items from this panel, which will immediately change the data available in the /data directory in the running tale.
  - The user can select the "+" button, which will display the "Select Data" modal (3.1). Using the select data modal, they can add data to the running tale. The "Select Data" operation changes the active tale.
  - Note: We've selected the "X" icon to remove data from the tale instead of a checkbox. Reselecting data would happen through the "Select Data" dialog.



### 3.4. Manage

We propose a major change to the Manage page to have an active Tale context similar to the Run page. Instead of "Environments" the user will have the "Launched Tales" panel on the right. Selecting a Tale changes the Data panel context. Each running Tale has an associated workspace directory that contains all files associated with the tale.

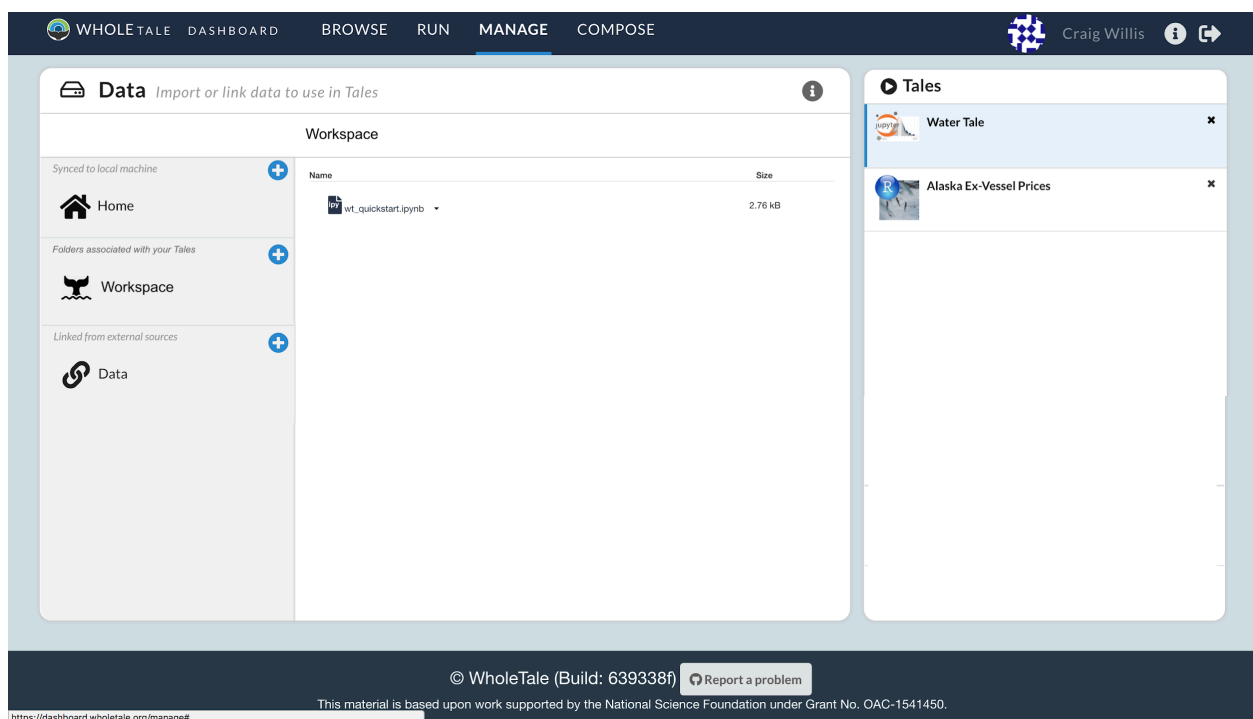
Example workflow:

- As with the Run page, the user selects a tale from the "Tales" panel, which changes the active context.
- Workspace is the workspace for the selected tale
- Data shows the data for the selected tale -- similar to the panel on the Run page.
- Selecting "+" on Home shows the current upload and create folder options (unchanged)

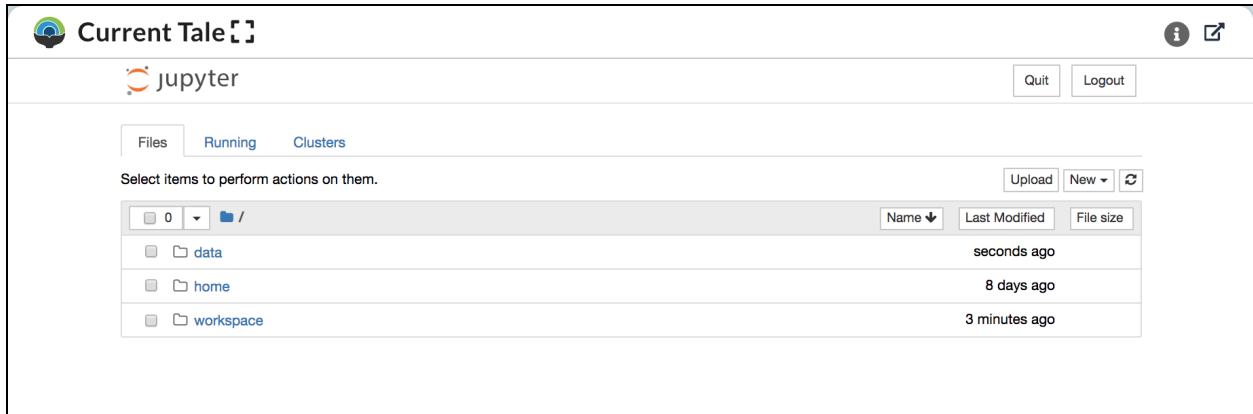
- Selecting "+" on Workspace displays the upload + create + select data
  - Upload allows the user to upload to the workspace
  - Create folder allows the user to create a folder in the workspace
  - "Select data" would display the "select data" modal but restricted to other workspaces the user has access to (3.1.3)
- Selecting "+" on Data displays the above "Select Data" modal (3.1.1 - 3.1.2), which combines the current register and select functions.
- The user can move or copy data from home > workspace or from workspace > home

#### Notes:

- Home is the same across all tales and present without a selected tale.



When accessing the running Tale, the contents are the same as the manage page with "home", "workspace" and "data" directories. The contents of these directories matches the contents in the Manage page at all times.



Open questions:

- Should the Tale workspace contain a copy of the tale.yaml, or does this only exist at publish?
- How do we handle deletion of the Tale workspace? Is it deleted with the Tale (for owners) or Tale instance (for non-owners)?
- Should we recommend or prescribe a workspace organization (inputs, outputs, scripts)?
- Do we retain the concept of the "narrative" or "entrypoint"?

### 3.5. Browse and "Copy on Launch"

(Updated 1/28/19)

With v0.5, a user running a tale has read-only access to the code and data. We propose changing the launch behavior to copy the tale and tale workspace depending on user access levels.

Per 1/28/19 discussions, we have identified three use cases with subtly different problems:

- User A wants to run user B's tale: The main problem here is if results are written to the workspace, the Tale will fail if filesystem is read-only. This suggests the need for a "results" or similar output directory
- User A wants to modify user B's tale: User A needs a copy of the workspace to modify it. This is the primary case for "Copy on Launch."
- User A wants many students to run example/tutorial tale: Ideally, the tale would be runnable with read-write output, but still referencing the same workspace. This way, the instructor can update the code without requiring students to restart.

Example workflow:

- User selects "Launch"
- Edit mode: If the user has edit access to the tale, the tale is launched using the workspace associated with the tale (read/write permissions).



- Copy mode: If the user does not have edit access, a copy of the tale and workspace is created at launch:
  - Tale parent is set to origin tale ID
  - User has full read/write access to the tale copy, including metadata

TBD:

- Today the browse panel shows a mixture of my tales and public tales and the "launched tales" are instances.
- If we introduce the "copy on launch", we have a new problem -- how to distinguish between public tales and my copy of the public tale.
  - CodeOcean does this by separating the "Browse" from the "Dashboard" (effectively "my" stuff)
  - We could convert "Launched" from instances to tales -- so the "Launched" tale list effectively contains a list of my things and copies of things I've run. This would require adding "Edit/View" and possibly "Stop/Start" options on the Tale.
- Introduces the question of "versioning" -- what to do when I've launched/copied a tale that changes.
- We would also need to address to how edit the tale metadata for my copy.
- One proposal is to split the browse view into public things and my things. Another is to introduce a new tab. Another is to repurpose the "Launched tales" for this.

### 3.6. Publish

The use of tale workspaces simplifies the publish process.

Example workflow:

- From either the Run or Manage page, the user selects "Publish"
  - TBD: Do we require tag/versioning tale or validation prior to publish?
- The Publish modal is displayed, but the file picker only operates on the tale workspace
  - Optionally we could drop the filepicker and tell the user to remove data prior to publishing -- using either manage or interactive environment.
- TBD: How to handle intermediate results when you only want to publish final results.
  - Keep the file picker or enforce a prescribed structure (e.g., scratch/temp)
- Q. Should licence be set in WT or at Publish?

### 3.7. Validation

- Originally discussed as a "clean" environment (i.e., no home directory)
- Options:
  - Provide interactive environment, allow user to determine when things are done
  - Provide a "background" non-interactive environment where we run on their behalf

- In either case, this is when we instrument for provenance (ala ReproZip)
- User clicks "Validate tale"
  - Starts a separate tale instance with no home directory, with copy of workspace
  - User can interactively validate the tale
  - When done, they select "Publish"
  - Publish workflow captures current state of validated tale
    - Published thing is versioned
  - Validation may be interrupted/abandoned -- deletes the instance and tale copy

### 3.8. Remix

Tale Workspaces and "copy on launch" introduce a new approach to remixing. The user can launch an existing tale and "remix" by dynamically changing elements of the tale.

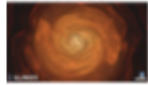


Example workflow:

- User launches a tale that they don't own
  - They get a copy of the tale
- To remix, they can edit along 3 dimensions -- code, environment, data
  - Metadata
    - As described above, user can edit the metadata for their copy (i.e., user can change Author), but need to track what it was based on via parent/child relationship
  - Code:
    - You have a copy of the workspace, delete and revise as you see fit
  - Environment:
    - Need the ability to modify the tale environment. This can be done via view/edit (see 3.9)
    - After creating a copy through launch, there is an edit icon that takes the user to some variation of the compose window. Here they can change environment.
    - Or edit tale and pick environment

### 3.9. View/Edit Tale

- User should be able to view any tale without running
- Should be able to see DOIs/information of associated data, publications, or parent tales
- If edit access, user can edit tale metadata
  - Change title, authors, categories, description, image
  - Change environment (remix support)
  - Change data (select data modal -- remix support)

- Change associated publications
- Change license/s
- User cannot edit
  - Datasets used is derived from current state of data
  - Derived from
  - Created
  - Version?

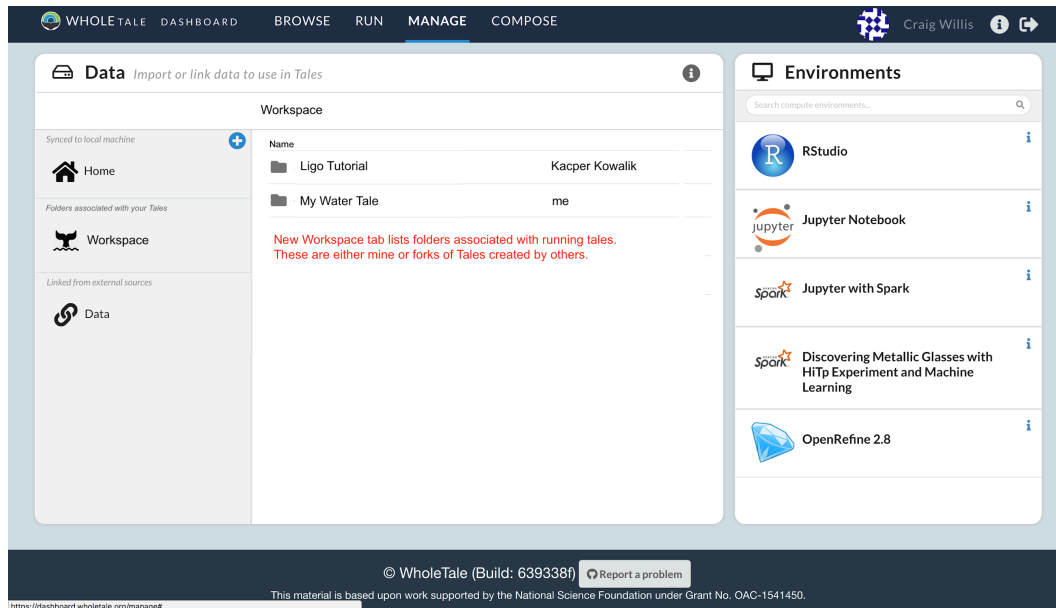
<b>Title</b>	Accelerated discovery of metallic glasses through iteration of machine learning and high-throughput experiments	
<b>Authors</b>	Logan Ward	
<b>Categories</b>	Materials Science	
<b>Description</b>	This tale includes scripts for creating and validating machine learning models to predict the glass-forming ability of metal alloys, and then using those models to discover new metallic glasses.	
<b>Environment</b>	 Jupyter Notebook	
<b>Datasets used</b>	Fang, Ren; Ward, Logan; Williams, Travis; Laws, Kevin J.; Wolverton, Christopher; Hatrick-Simpers, Jason; Mehta, Apurva, "Accelerated Discovery of Metallic Glasses through Iteration of Machine Learning and High-Throughput Experiments," 2018, <a href="http://dx.doi.org/doi:10.18126/M2B06M">http://dx.doi.org/doi:10.18126/M2B06M</a>	
<b>Publications</b>	Fang, Ren; Ward, Logan; Williams, Travis; Laws, Kevin J.; Wolverton, Christopher; Hatrick-Simpers, Jason; Mehta, Apurva, "Accelerated Discovery of Metallic Glasses through Iteration of Machine Learning and High-Throughput Experiments," Science Advances 13 Apr 2018: Vol. 4, no. 4, eaag1566 DOI: 10.1126/sciadv.aag1566	
<b>Data</b>	 Browse data	
<b>License</b>	CC0	
<b>Derived from</b>	(information about parent tale, if present)	
<b>Versions</b>	v1	
<b>Created</b>	Jan 1, 1970	
		<a href="#">Launch</a> <a href="#">Delete</a>

- The user can edit the environment associated with the tale by selecting from any image available on the system. See [Custom environments](#).

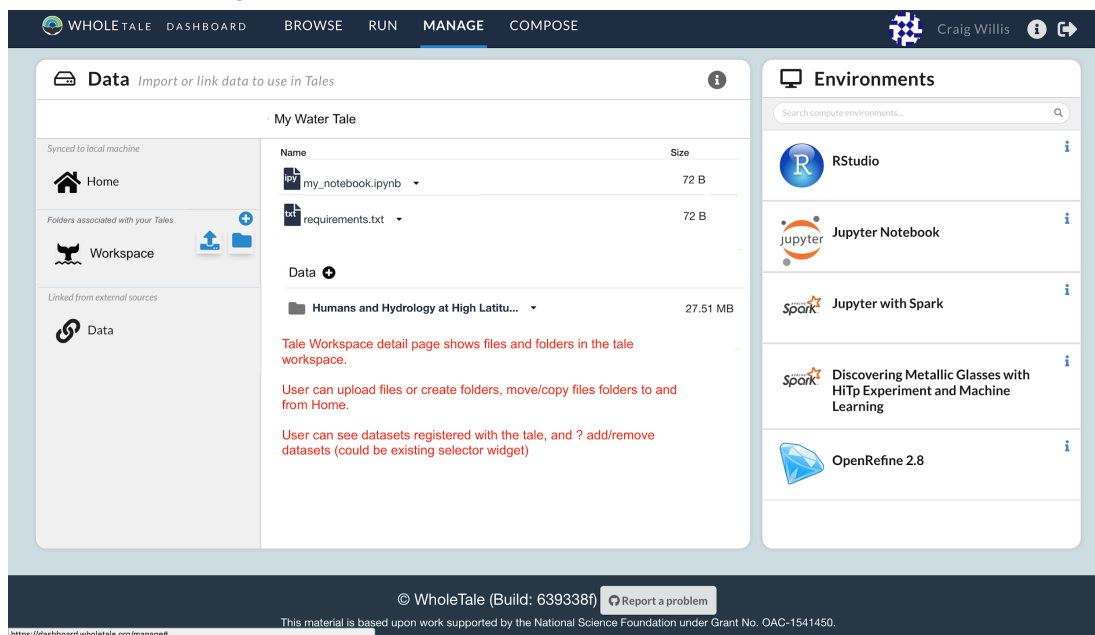
### 3.10. Alternative Workspace Manage

Alternative to Manage model defined in Section 3.4

The Manage page contains a new Workspaces section. Each running Tale has an associated workspace directory that contains all files associated with the tale.



The user can view and modify the contents of the Tale workspace via the Manage page. This includes uploading files, creating folders, moving/copying files from home. The Tale Workspace also shows the registered datasets associated with the Tale, which can be added/removed.



When accessing the running Tale, the contents are the same as the workspace directory with a "data" directory for registered data. (Note: this raises a question about how to handle when the user has a "data" directory in the Tale workspace)