MyData Documentation

Release 0.2.0-alpha2

James Wettenhall

Sep 06, 2019

Contents

1 Contents **1**

1.1 Overview . . . 1
1.2 Download . . . 5
1.3 MyTardis Prerequisites . . . 6
1.4 Settings . . . 8
1.5 Test Run . . . 17
1.6 Upload Methods . . . 18
1.7 Upload Speed . . . 23
1.8 User Groups . . . 30
1.9 Mac OS X Walkthrough . . . 36
1.10 MyData Tutorial . . . 41
1.11 License . . . 55

2 Indices and tables **69**

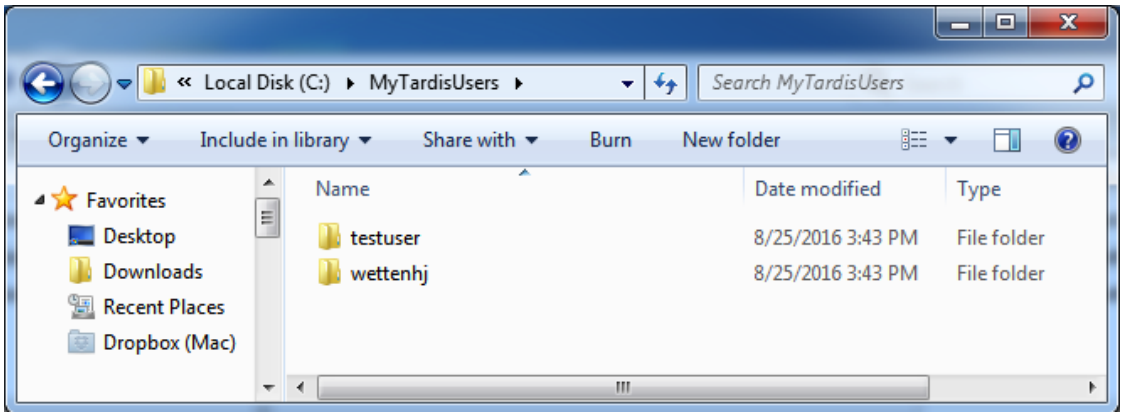
CHAPTER 1

Contents

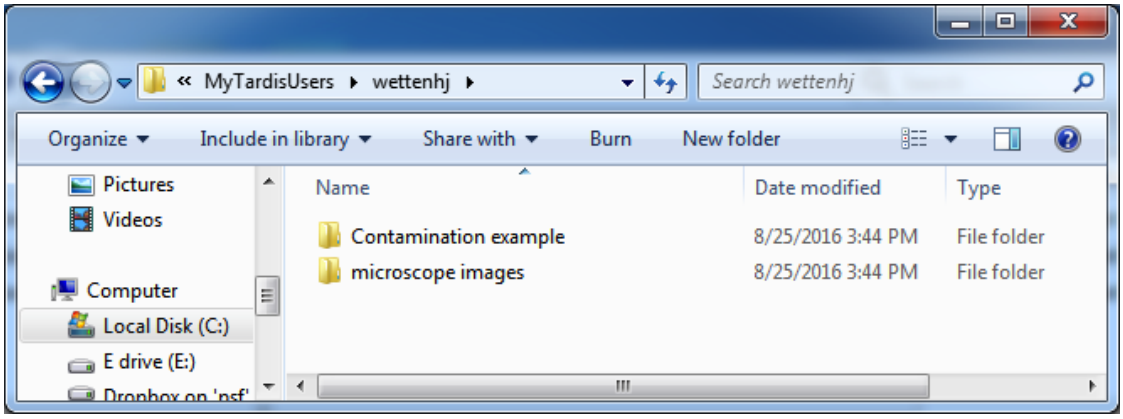
1.1 Overview

MyData is a desktop application for uploading data to MyTardis (https://github.com/mytardis/mytardis). It allows users of a data-collection instrument to have their data automatically uploaded to MyTardis simply by saving their data in a pre-defined folder structure. The simplest folder structures available is "Username / Dataset", which is described below.

We begin with a root data directory (e.g. "C:\MyTardisUsers") containing one folder for each MyTardis user. In the example below, we have two users with MyTardis usernames "testuser" and "wettenhj".

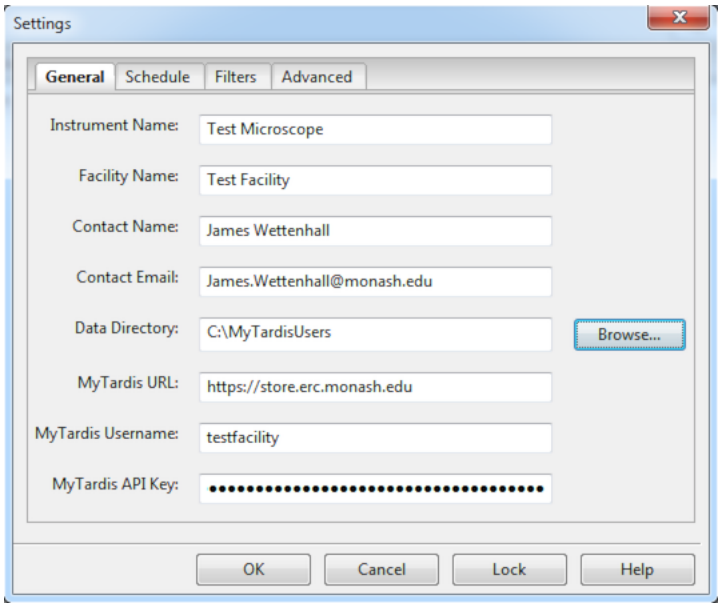


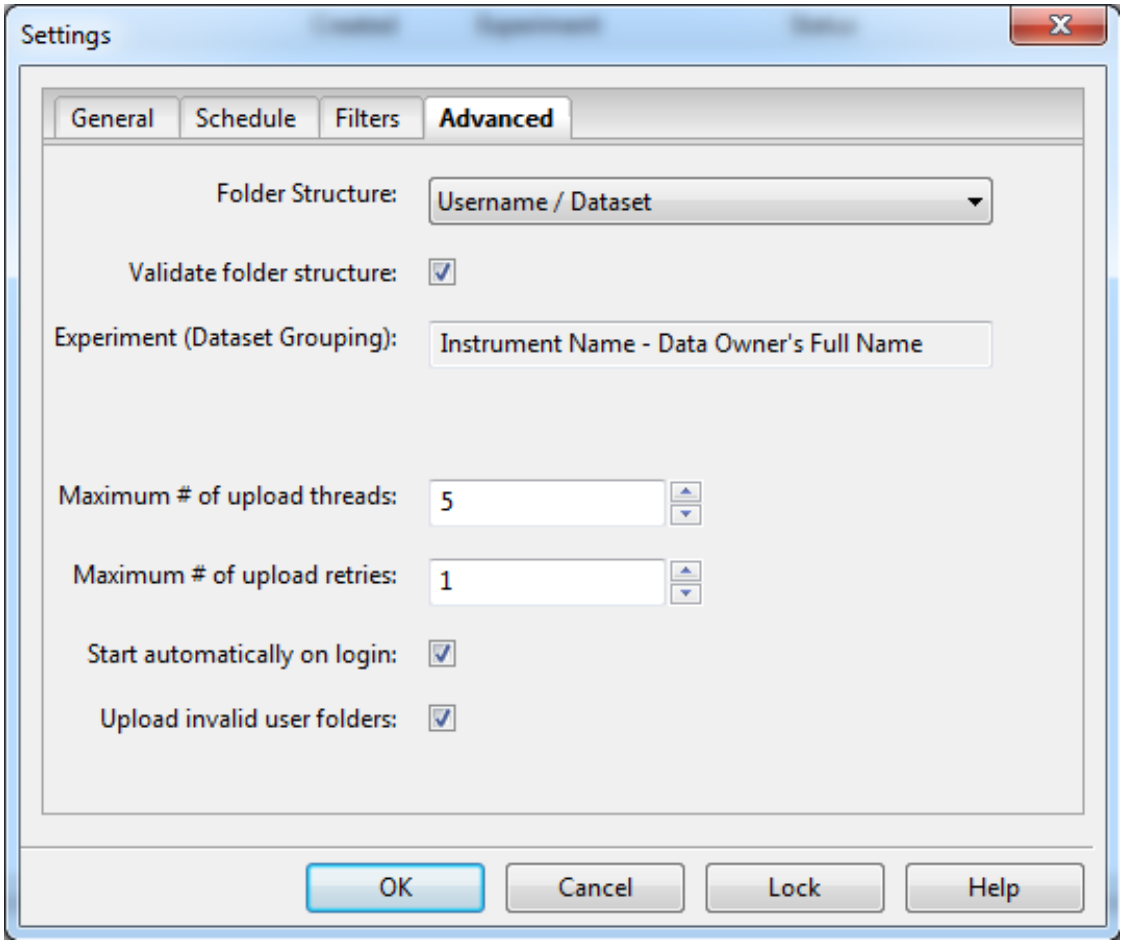
Within each user folder, we can add as many folders as we like, and each one will become a "dataset" within MyTardis:



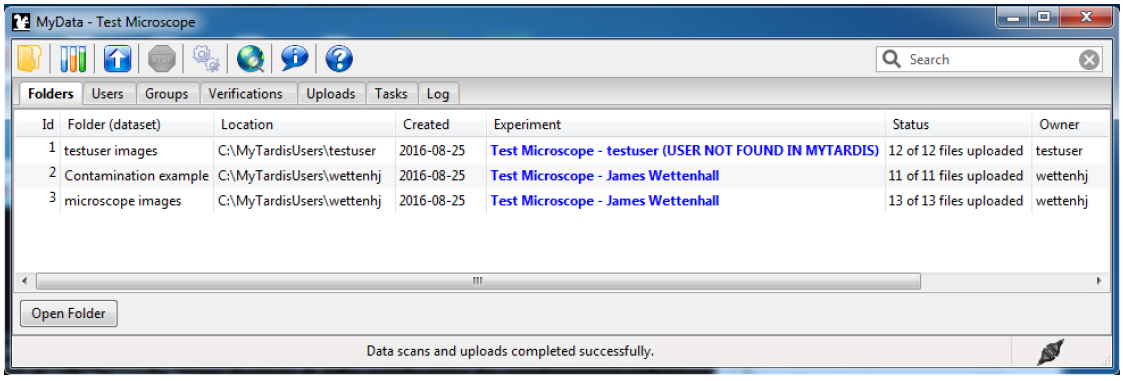
MyData is designed to be able to run interactively or in the background. Once its settings have been configured, it minimizes itself to the system tray (Windows) or menubar (Mac OS X) and runs the data scan and upload task according to a scheduled configured by the user. Many data-collection instrument PCs use a shared login account which remains logged in all day long. MyData at present cannot run as a service daemon - so it will not run while no users are logged in.

The first time you run MyData, you will be asked to enter some settings, telling the application how to connect to your MyTardis instance. You can use a MyTardis account which is shared amongst facility managers, but which general users don't have access to.

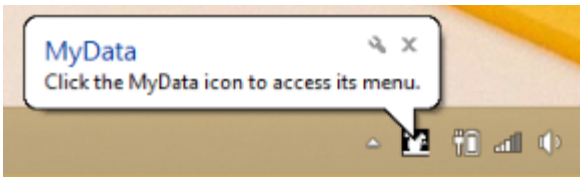




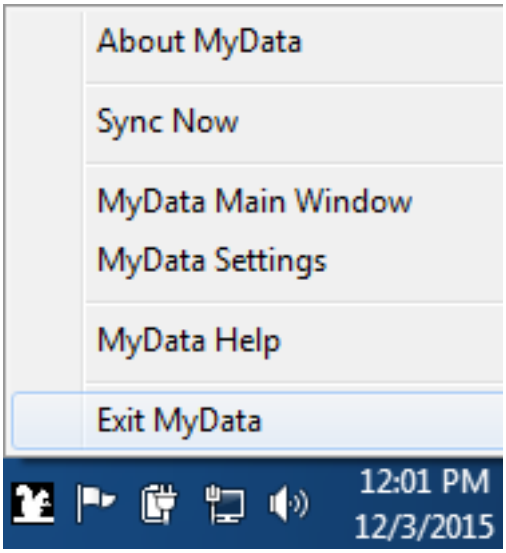
Each time the application starts up (and when you tell it to), it will scan all of the user and datset folders within your primary data directory (e.g. C:\MyTardisUsers) and present a list of all of the dataset folders in a tabular view (below). MyData will count the number of files within each dataset folder (including nested subdirectories), and then query MyTardis to determine how many of these files have already been uploaded. If MyData finds new files which haven't been uploaded, it will begin uploading them (with a default maximum of 5 simultaneous uploads). You can see progress of the uploads in the "Uploads" tab.



Closing MyData's main window will minimize the MyData application to an icon in the System Tray (below). It is possible to exit MyData using a menu item from the System Tray icon's pop-up menu (further below), but exiting will prevent MyData from being able to run scheduled tasks.



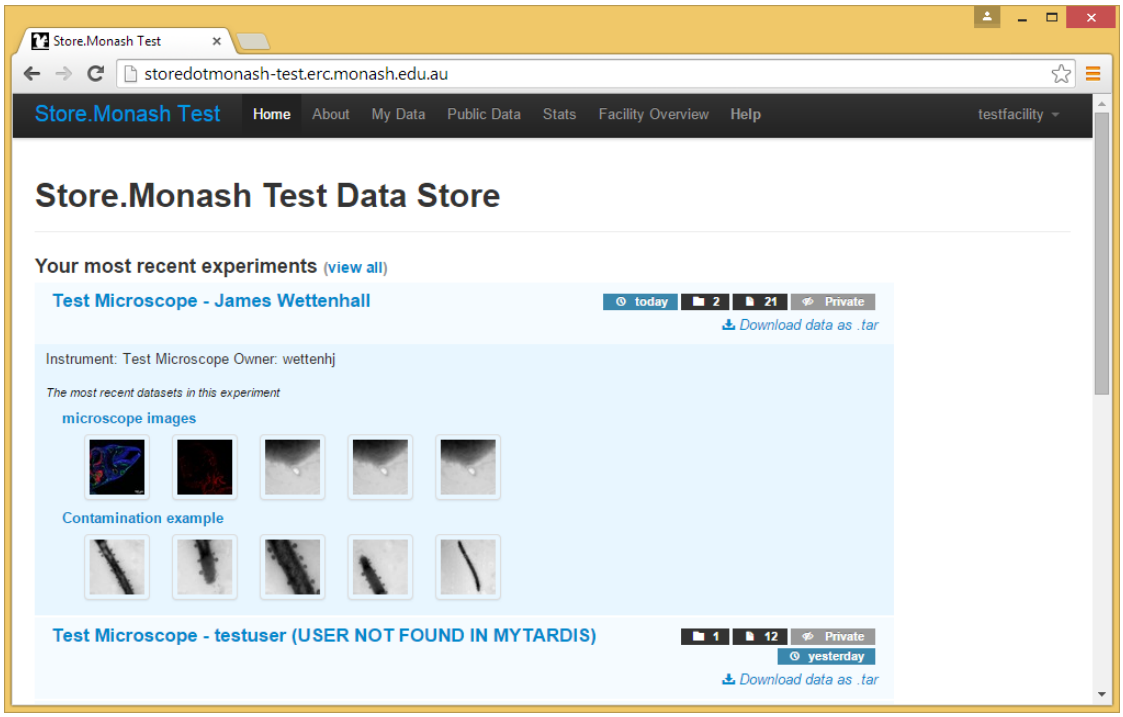
Clicking on MyData's System Tray icon will bring up a menu, allowing you to restore MyData's main window (the "Control Panel") or "Sync Now" to ensure that new data is uploaded promptly:



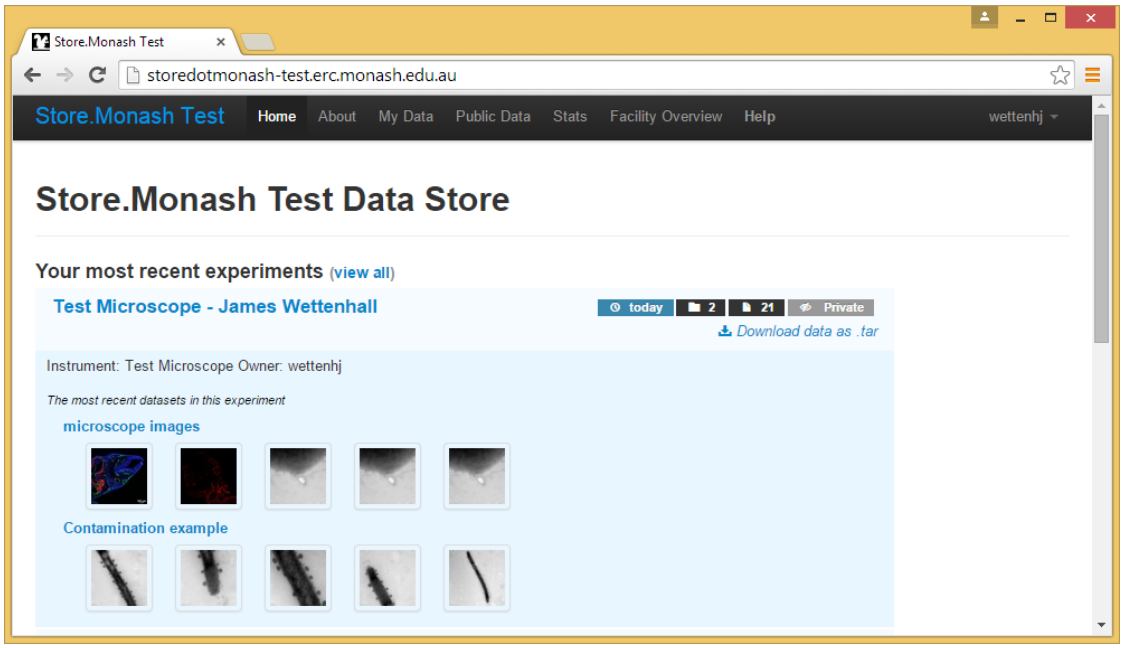
You can tell when MyData has finished uploading a dataset by looking at the number of files uploaded in the Status column of the Folders view. Then you can select that dataset's row in the Folders view and click on the "Web Browser" icon to view that dataset in MyTardis.

MyTardis uses "experiments" to organize collections of datasets. When using the default "Username / Dataset" folder structure, the default name for each experiment created by MyData will be the instrument name (e.g. "Test Microscope"), followed by the data owner's full name (if it can be retrieved from MyTardis using the username given as a folder name).

The experiment will initially be owned by the facility manager user specified in MyData's Settings dialog (e.g. "testfacility"). MyData will then use MyTardis's ObjectACL's (access control lists) to share ownership with the individual researcher (e.g. "wettenhj" or "skeith") who must have a MyTardis account. Below we can see the experiments created by MyData as owned by the facility manager user ("testfacility").



And below, we can see user wettenhj's data - note that "wettenhj" is now the logged-in MyTardis user in the upper-right corner, instead of "testfacility".



1.2 Download

1.2.1 Current Version

- Download MyData for Windows: MyData_v0.8.2.exe

- Download MyData for Mac OS X: MyData_v0.8.2.dmg
- Download MyData for RHEL 7: mydata-0.8.2-1.el7.centos.x86_64.rpm

1.2.2 Previous Version

- Download MyData for Windows: MyData_v0.8.1.exe
- Download MyData for Mac OS X: MyData_v0.8.1.dmg
- Download MyData for RHEL 6: mydata-0.8.1-1.el6.x86_64.rpm
- Download MyData for RHEL 7: mydata-0.8.1-1.el7.centos.x86_64.rpm

1.2.3 MyData Releases GitHub Page

- MyData Releases
- Old MyData Releases

1.3 MyTardis Prerequisites

These instructions are for MyTardis server administrators who wish to support uploads from MyData.

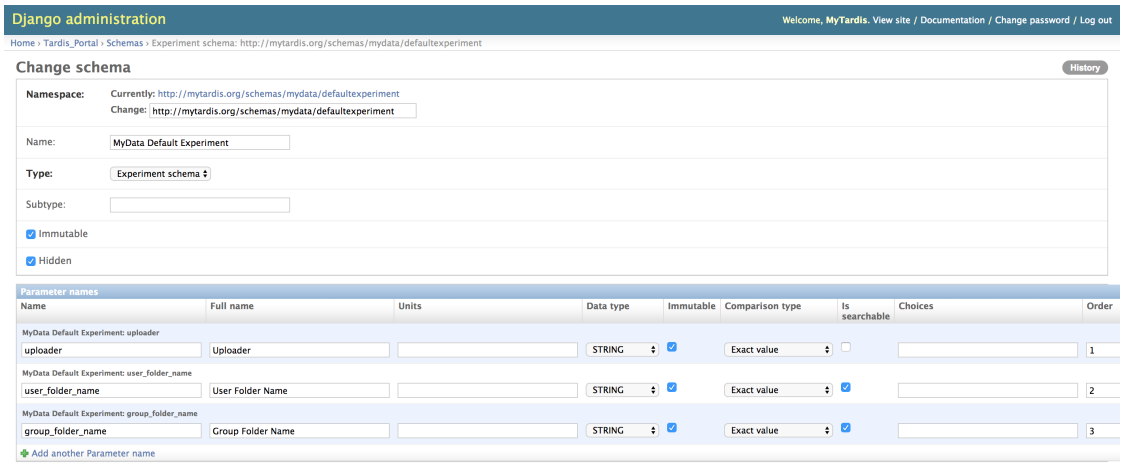
1.3.1 MyData App for MyTardis

MyData requires the "mydata" MyTardis app to be installed on the MyTardis server. This app, and its installation instructions can be found here: https://github.com/mytardis/mytardis-app-mydata/blob/master/README.md

You should use the HEAD of the master branch of "mytardis-app-mydata", i.e.

```
$ cd tardis/apps/
$ git clone https://github.com/mytardis/mytardis-app-mydata mydata
$ cd mydata
```

MyData stores metadata for each experiment it creates, including a reference to the MyData instance (uploader) which created the experiment, and the name of the user folder the experiment was created for. A schema must be added to MyTardis to support this:

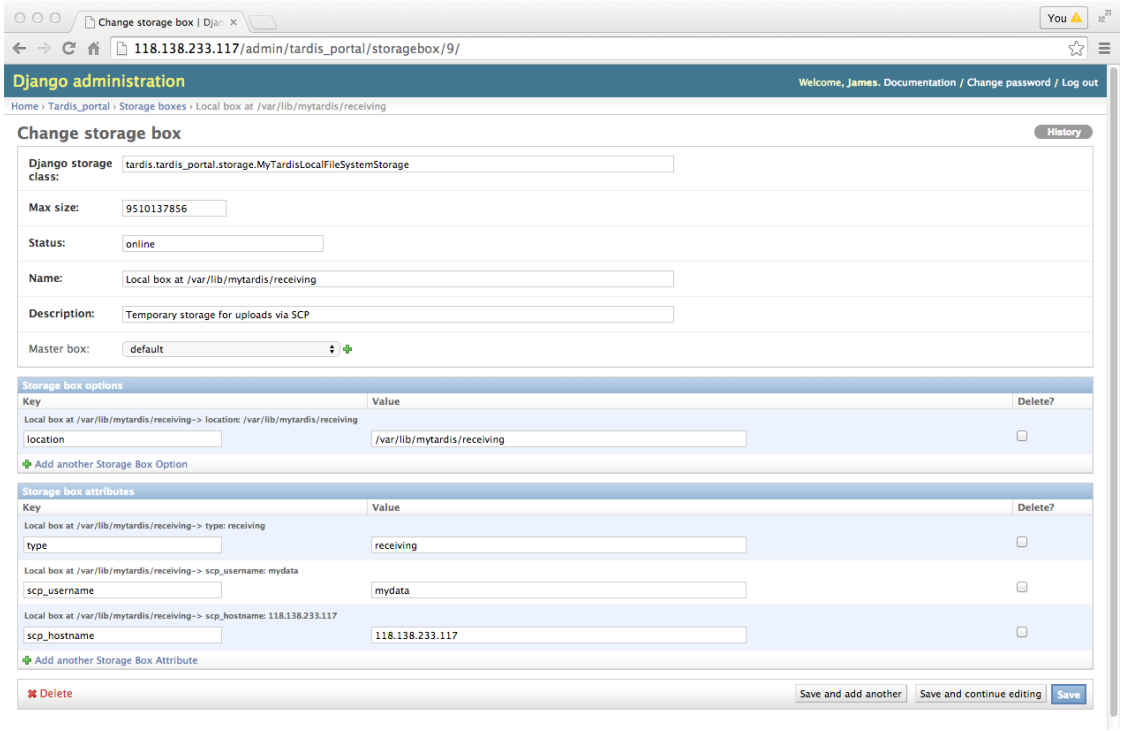


The final step of the instructions in https://github.com/mytardis/mytardis-app-mydata/blob/master/README.md describes how to create this schema, which is just a matter of running:

```
python mytardis.py loaddata tardis/apps/mydata/fixtures/default_experiment_schema.json
```

after installing the "mytardis-app-mydata" MyTardis app in "tardis/apps/mydata".

MyData requires the use of a "receiving" storage box (also know as a "staging" storage box) in MyTardis, which serves as a temporary location for uploaded files. MyTardis will automatically create a storage box if a client like MyData attempts to perform staging uploads via the API. To enable uploads via staging (using SCP) in MyData, which are recommended over HTTP POST uploads, it is necessary to add the "scp_username" and "scp_hostname" attributes to the storage box, as illustrated below.



DEFAULT_RECEIVING_DIR in tardis/settings.py should be set to match the location option of the "staging" (or "receiving") storage box, e.g. "/var/lib/mytardis/receiving". Similarly, DEFAULT_STORAGE_BASE_DIR in tardis/settings.py should be set to match the location option of the "master" storage box, which is called "default" above.

For more information on uploads via staging, see see *SCP to Staging*.

1.3.2 Creating a MyTardis User Account for MyData

The first time MyData is launched, a blank Settings dialog will appear, requiring a MyTardis username and an API key for that user account. The API key is saved to disk along with MyData's other settings to allow unattended uploads (see *Saving and Loading Settings*). While performing unattended uploads, MyData creates experiments, datasets and datafiles using MyTardis's RESTful API.

The MyTardis user account supplied to MyData should be a member of the facility managers group for the facility specified in MyData's Settings dialog, and it should have the following permissions, which can be set in MyTardis's Django Admin interface:

- tardis_portal.add_datafile

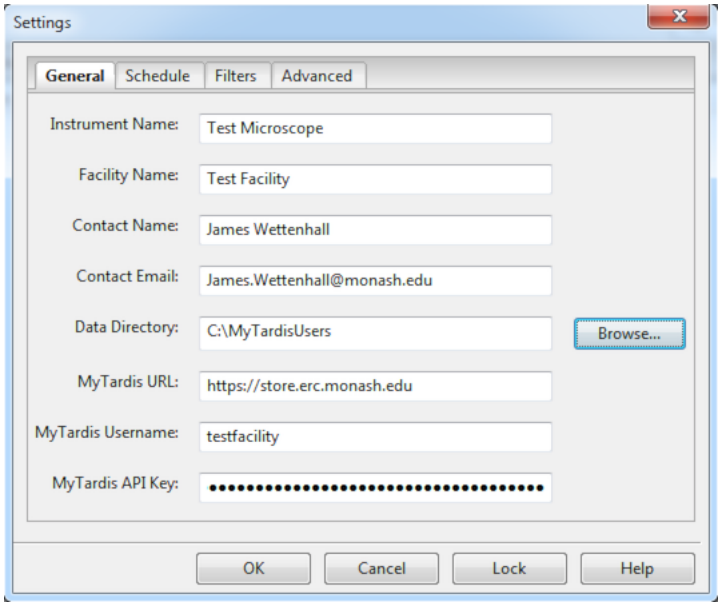
- tardis_portal.add_dataset
- tardis_portal.change_dataset
- tardis_portal.add_experiment
- tardis_portal.change_experiment
- tardis_portal.add_instrument
- tardis_portal.change_instrument
- tardis_portal.add_objectacl

Rather than manually adding permissions each time you create a MyTardis account to use with MyData, you can create a "mydata-default-permissions" group or a "facility-manager-default-permissions" group and add new MyData user accounts to the existing group to inherit some sensible default permissions. Below, we can see that the role account "testfacility" is a member of two groups, "facility-managers" and "test_facility_managers". The "facility-managers" group contanis sensible default permission to be inherited, and the "test_facility_managers" group membership grants the "testfacility" account access the "Test Facility" facility.

1.4 Settings

MyData's Settings dialog can be opened by clicking on the icon on MyData's toolbar, or by selecting the "MyData Settings" menu item in the MyData System Tray icon's pop-up menu. The Settings dialog will be automatically displayed the first time MyData is launched.

1.4.1 General



Instrument Name The name of the instrument (e.g. "Nikon Microscope #1") whose data is to be uploaded to MyTardis by this MyData instance. If an instrument record with this name doesn't already exist in MyTardis within the facility specified below, then MyData will offer to create one (assuming that you are a member of a facility managers group for that facility in MyTardis).

Facility Name The name of the facility containing the instrument to upload data from. A facility record must have been created by your MyTardis administrator before you can use MyData, and the default MyTardis username you enter below (the initial owner of all data uploaded by this instance) must be a member of the managers group for that facility. MyData will automatically check that a facility record matching this facility name exists on the MyTardis server specified by the MyTardis URL below. If it doesn't exist, MyData will offer suggestions for alternative facilities which your MyTardis account is a manager of (if any).

Contact Name MyData's preferred upload method (staging) requires approval from a MyTardis administrator. This Contact Name will be used when sending confirmation that access to MyTardis's staging area has been approved for this MyData instance.

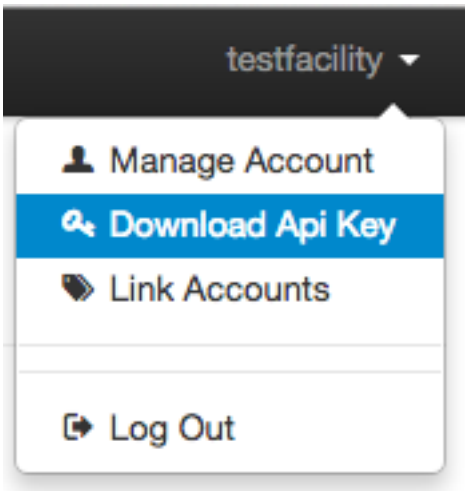
Contact Email MyData's preferred upload method (staging) requires approval from a MyTardis administrator. This Contact Email will be used when sending confirmation that access to MyTardis's staging area has been approved for this MyData instance.

Data Directory Choose a folder where you would like to store your data. e.g. D:\Data

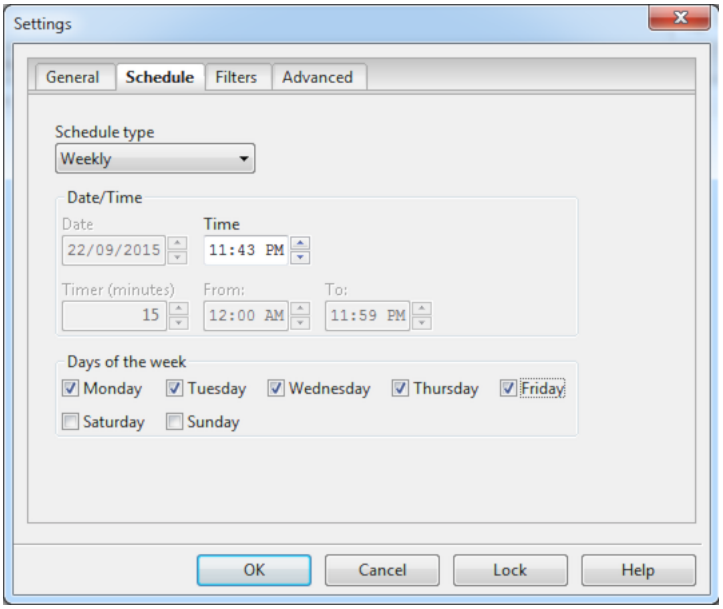
MyTardis URL The URL of a MyTardis server running a MyTardis version compatible with MyData, e.g. http://118.138.241.91/

MyTardis Username Do not put your individual MyTardis username (e.g. "jsmith") in here. Because MyData is designed to be able to upload multiple users' data from an instrument PC, the default username used by MyData should generally be a facility role account, e.g. "testfacility".

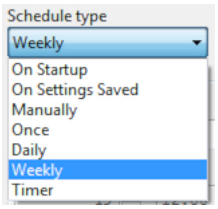
MyTardis API Key API keys are similar to passwords, but they are easier to revoke and renew when necessary. Ask your MyTardis administrator for the API key corresponding to your facility role account.



1.4.2 Schedule



Schedule types



Schedule type - On Startup Run the folder scans and uploads automatically when MyData is launched.

Schedule type - On Settings Saved Run the folder scans and uploads automatically afte the user clicks OK on the Settings dialog.

Schedule type - Manually Only run the folder scans and uploads in response to user interaction - either by clicking the Refresh icon on the toolbar, or by clicking the "Sync Now" menu item in the system tray menu.

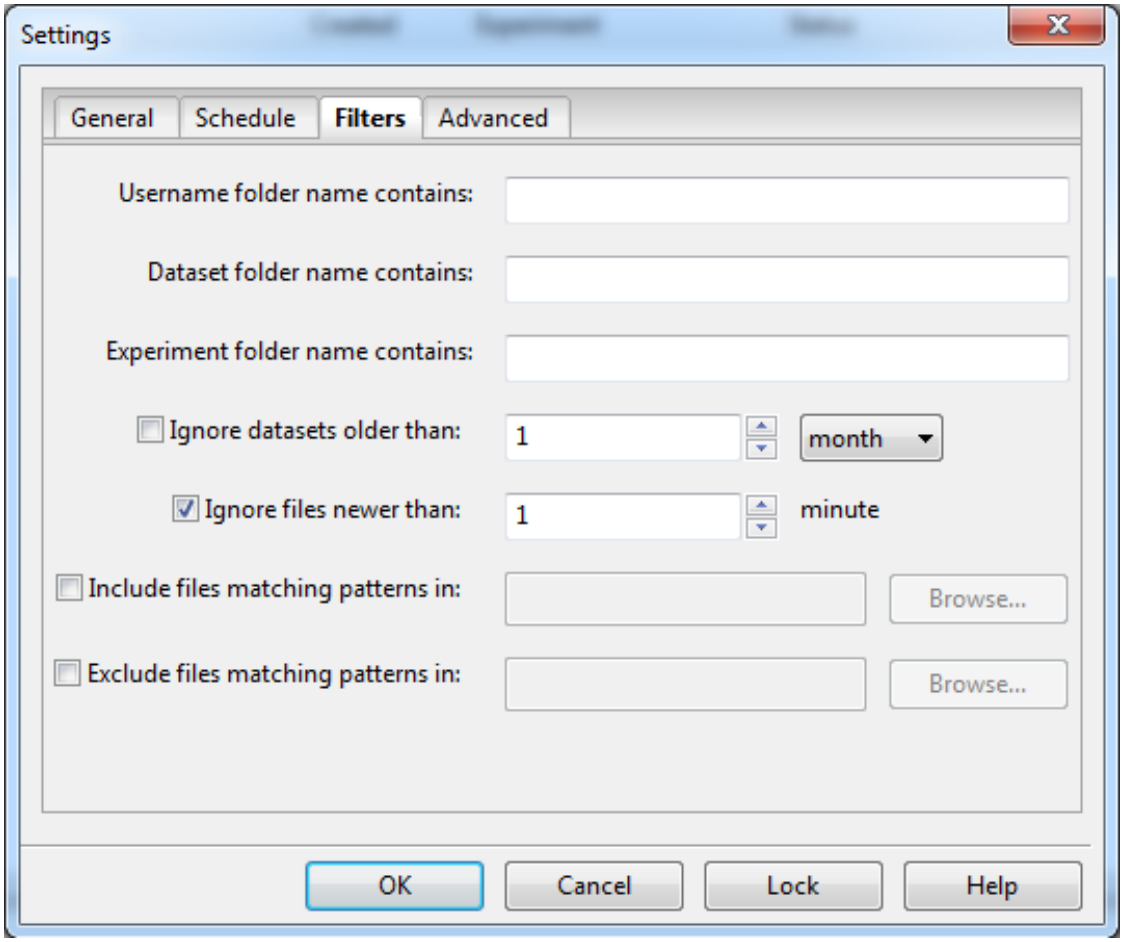
Schedule type - Once Run the folder scans and uploads once, on the date specified by the Date field and at the time specified by the Time field.

Schedule type - Daily Run the folder scans and uploads every day, at the time specified by the Time field.

Schedule type - Weekly Run the folder scans and uploads every week on the day(s) specified by the weekday checkboxes, at the time specified by the Time field.

Schedule type - Timer Run the folder scans and uploads repeatedly with an interval specified by the "Timer (minutes)" field between the hours of "From" and "To", every day.

1.4.3 Filters



Username/Email/User Group folder name contains Only scan user folders (or user group folders) whose username (or email or user group) contains the string provided. The actual text of this setting will change, depending on the Folder Structure specified in the Advanced tab.

Dataset folder name contains Only scan dataset folders whose folder name contains the string provided.

Experiment folder name contains Only scan experiment folders whose folder name contains the string provided. This field will be hidden unless the Folder Structure specified in the Advanced tab includes an Experiment folder.

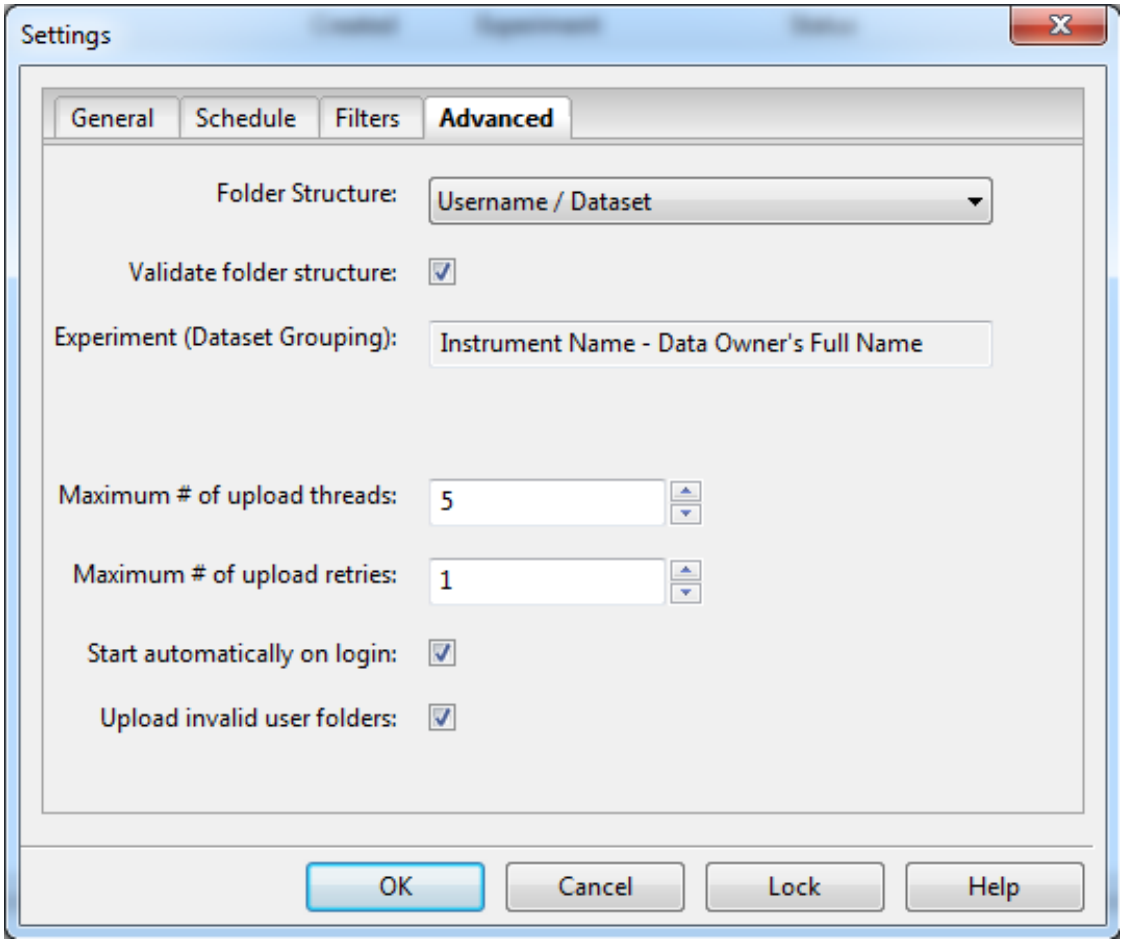
Ignore datasets older than MyData is designed to be used for uploading recent data. If it is configured to use an existing data directory containing a large backlog of old data, it is advisable to instruct MyData to ignore old datasets so that it focus on uploading the recent datasets.

Ignore files newer than MyData can ignore recently modified files. MyTardis does not yet support file versioning, so once a file has been uploaded and verified, it will not be replaced by a newer version. Therefore, it is important to ensure that a file doesn't get uploaded while it is still being modified.

Includes files matching patterns in Specifying an includes file will tell MyData to only upload files matching specified patterns, e.g. '*.txt'. The includes file should contain one pattern on each line. Any lines beginning with '#' or ';' will be ignored. The patterns will be matched using Python's fnmatch: https://docs.python.org/2/library/fnmatch.html If both an includes and an excludes file are specified, then filenames matching one or more includes patterns will be uploaded, even if they also match one or more excludes patterns.

Excludes files matching patterns in Specifying an excludes file will tell MyData not to upload files matching specified patterns, e.g. *.bak The excludes file should contain one pattern on each line. Any lines beginning with '#' or ';' will be ignored. The patterns will be matched using Python's fnmatch: https://docs.python.org/2/library/fnmatch.html If both an includes and an excludes file are specified, then filenames matching one or more includes patterns will be uploaded, even if they also match one or more excludes patterns.

1.4.4 Advanced



Folder Structure - Username / Dataset Folders immediately inside the main data directory (e.g. "D:\Data\jsmith") are assumed to be MyTardis usernames. Folders inside each user folder (e.g. "D:\Data\jsmith\Dataset1" will be mapped to MyTardis datasets. Datasets will be automatically grouped into MyTardis experiments according to the "Experiment (Dataset Grouping)" field below.

Folder Structure - Email / Dataset This folder structure works best when email addresses are unique per user in

MyTardis. There is no constraint requiring email addresses to be unique in MyTardis, but if MyTardis is using an external authentication provider (e.g. LDAP), there may be a requirement in the authentication provider making email addresses unique. Folders immediately inside the main data directory (e.g. "D:\Data\John.Smith@example.com") are assumed to be email addresses which can be used to match MyTardis user accounts. If you wish to use email addresses as folder names, an alternative is to use the "Username / Dataset" folder structure and use email addresses for usernames in MyTardis. Folders inside each email folder (e.g. "D:\Data\John.Smith@example.com\Dataset1" will be mapped to MyTardis datasets. Datasets will be automatically grouped into MyTardis experiments according to the "Experiment (Dataset Grouping)" field below.

Folder Structure - Username / Experiment / Dataset Folders immediately inside the main data directory (e.g. "D:\Data\jsmith") are assumed to be MyTardis usernames. Folders inside each user folder (e.g. "D:\Data\jsmith\Experiment1" will be mapped to MyTardis experiments. Folders inside each experiment folder (e.g. "D:\Data\jsmith\Experiment1\Dataset1") will be mapped to MyTardis datasets.

Folder Structure - Email / Experiment / Dataset This folder structure works best when email addresses are unique per user in MyTardis. There is no constraint requiring email addresses to be unique in MyTardis, but if MyTardis is using an external authentication provider (e.g. LDAP), there may be a requirement in the authentication provider making email addresses unique. Folders immediately inside the main data directory (e.g. "D:\Data\John.Smith@example.com") are assumed to be email addresses which can be used to match MyTardis user accounts. If you wish to use email addresses as folder names, an alternative is to use the "Username / Experiment / Dataset" folder structure and use email addresses for usernames in MyTardis. Folders inside each email folder (e.g. "D:\Data\John.Smith@example.com\Experiment1" will be mapped to MyTardis experiments. Folders inside each experiment folder (e.g. "D:\Data\John.Smith@examples.com\Experiment1\Dataset1") will be mapped to MyTardis datasets.

Folder Structure - Username / "MyTardis" / Experiment / Dataset Folders immediately inside the main data directory (e.g. "D:\Data\jsmith") are assumed to be MyTardis usernames. Folders inside each "MyTardis" folder (e.g. "D:\Data\jsmith\MyTardis\Experiment1" will be mapped to MyTardis experiments. Folders inside each experiment folder (e.g. "D:\Data\jsmith\MyTardis\Experiment1\Dataset1") will be mapped to MyTardis datasets.

Folder Structure - User Group / Instrument / Full Name / Dataset Folders immediately inside the main data directory (e.g. "D:\Data\SmithLab") are assumed to be MyTardis user groups. The actual group name in MyTardis (e.g. "TestFacility-SmithLab") may have a prefix (e.g. "TestFacility-") prepended to it, specified by the "User Group Prefix" field below. Each user group folder should contain exactly one folder (e.g. "D:\Data\SmithLab\Nikon Microscope #1") specifying the name of the instrument. Using this scheme allows copying data from multiple instruments to a file share with the instrument name folder allowing users to distinguish between datasets from different instruments on the file share. Folders inside each instrument folder (e.g. "D:\Data\SmithLab\Nikon Microscope #1\John Smith") indicate the name of the researcher who collected the data or the researcher who owns the data. Access control in MyTardis will be determined by the user group ("Smith Lab"), whereas the researcher's full name will be used to determine the default experiment (dataset grouping) in MyTardis. Folders inside each full name folder (e.g. "D:\Data\SmithLab\Nikon Microscope #1\John Smith\Dataset1") will be mapped to MyTardis datasets.

Folder Structure - Experiment / Dataset Folders immediately inside the main data directory (e.g. "D:\Data\Experiment1") will be mapped to MyTardis experiments. Folders inside each experiment folder (e.g. "D:\Data\Experiment1\Dataset1") will be mapped to MyTardis datasets. This folder structure is designed for a data collection facility which is primarily staff-operated (not user-operated), so instead of creating a folder for each user, an 'experiment' folder is used to group datasets which should be accessible by the same group of researchers.

Folder Structure - Dataset Folders immediately inside the main data directory (e.g. "D:\Data\Dataset1") will be mapped to MyTardis datasets. Datasets will be automatically grouped into MyTardis experiments according to the "Experiment (Dataset Grouping)" field below. This folder structure is designed for a facility using inflexible data collection software making it difficult to structure folders according to who should have access to them.

Folder Structure - User Group / Dataset Folders immediately inside the main data directory (e.g. "D:\Data\SmithLab") are assumed to be MyTardis user groups. Folders inside each user group folder

(e.g. "D:\Data\SmithLab\Dataset1" will be mapped to MyTardis datasets. Datasets will be automatically grouped into a MyTardis experiment whose title is the name of the User Group.

Folder Structure - User Group / Experiment / Dataset Folders immediately inside the main data directory (e.g. "D:\Data\SmithLab") are assumed to be MyTardis user groups. Folders inside each user group folder (e.g. "D:\Data\SmithLab\Experiment1" will be mapped to MyTardis experiments. Folders inside each experiment folder (e.g. "D:\Data\SmithLab\Experiment1\Dataset1") will be mapped to MyTardis datasets.

Validate Folder Structure When this is checked, MyData will ensure that the folders provided appear to be in the correct structure, and it will count the total number of datasets. This can be disabled if you have a large number of dataset folders and slow disk access.

Experiment (Dataset Grouping) Defines how datasets will be grouped together into experiments in MyTardis. Currently, this field is automatically populated when you select a folder structure (above), and cannot be modified further.

User Group Prefix Used with the "User Group / Instrument / Full Name / Dataset" folder structure. Folders immediately inside the main data directory (e.g. "D:\Data\SmithLab") are assumed to be MyTardis user groups. The actual group name in MyTardis (e.g. "TestFacility-SmithLab") may have a prefix (e.g. "TestFacility-") prepended to it.

Max # of upload threads The maximum number of uploads to perform concurrently. If greater than one, MyData will spawn multiple scp (secure copy) processes which (for large datafiles) may impact significantly on CPU usage of your system, which could affect other applications running alongside MyData. The default value is 5.

Max # of upload retries The maximum number of times to retry uploading a file whose upload initially fails, e.g. due to a connection timeout error.

Start automatically on login As of v0.7.0-beta6, this checkbox is now disabled (read only) on Windows, because MyData is configured to start automatically for all users (if using MyData's setup wizard's default options) when it is first installed. Then it is up to the system administrator to decide whether to leave the MyData shortcut in C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup\

On Mac and Linux, MyData configures itself to start automatically on login only for the current user, based on the value of this checkbox.

On Mac OS X, a login item will be created in the user's ~/Library/Preferences/com.apple.loginitems.plist which can be accessed from System Preferences, Users & Groups, Login Items.

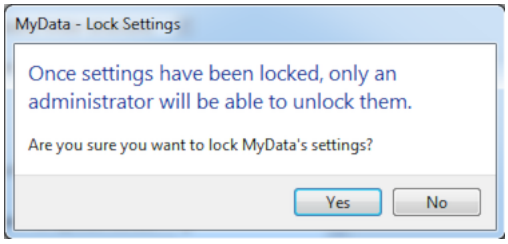
On Linux, MyData.desktop will be copied to ~/.config/autostart/ if this checkbox is ticked.

Upload invalid user folders If MyData finds a user (or group) folder which doesn't match a user (or group) on the MyTardis server, it can be configured to upload the data anyway (and assign it to the facility role account) by leaving this checkbox ticked. Or the checkbox can be unticked if you want MyData to ignore user folders which can't be mapped to users or groups on the MyTardis server.

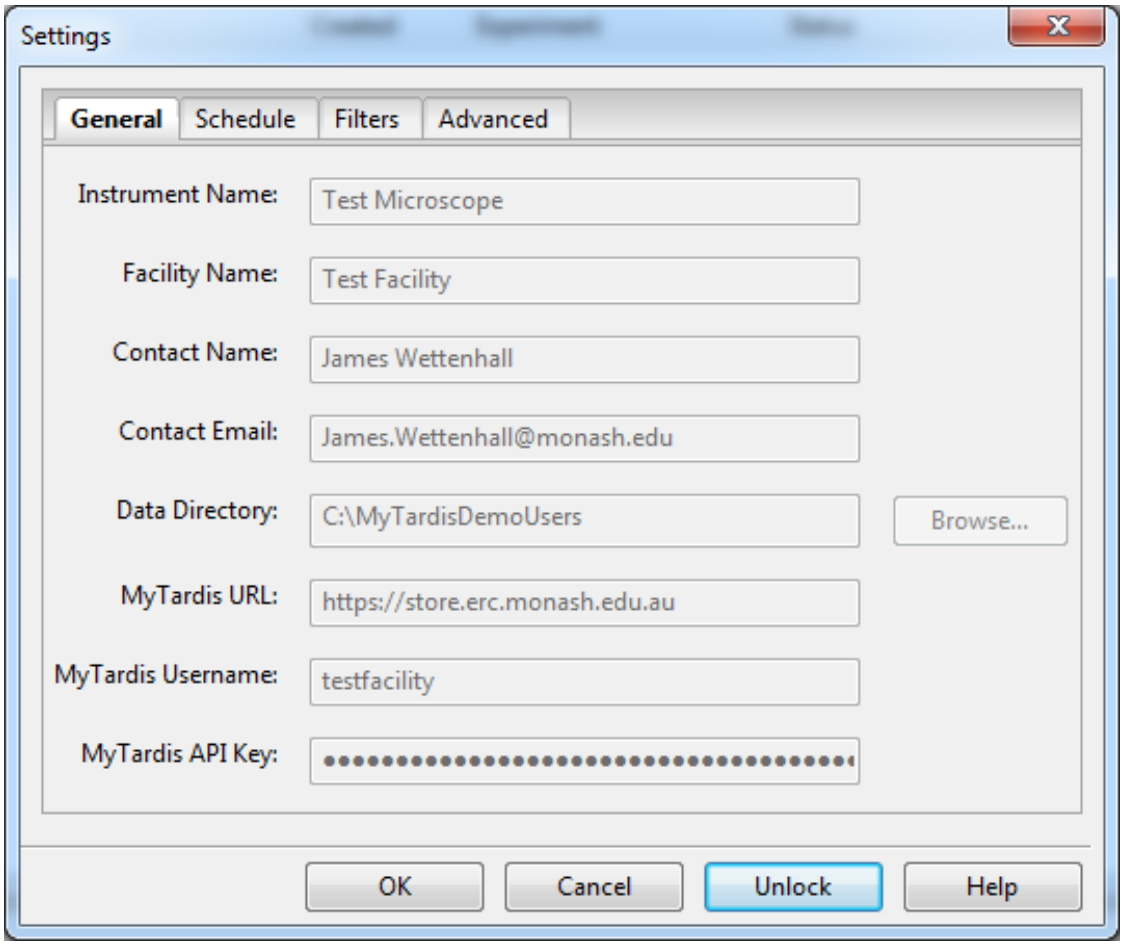
1.4.5 Locking and Unlocking MyData's Settings

At the bottom of MyData's Setting dialog is a Lock/Unlock button, whose label toggles between "Lock" and "Unlock" depending on whether the Settings dialog's fields are editable or read-only. When the Settings dialog's fields are editable, clicking the "Lock" button will make them read-only, preventing any further changes to MyData's settings until an administrator has unlocked the settings. The locked status will persist after closing and relaunching MyData.

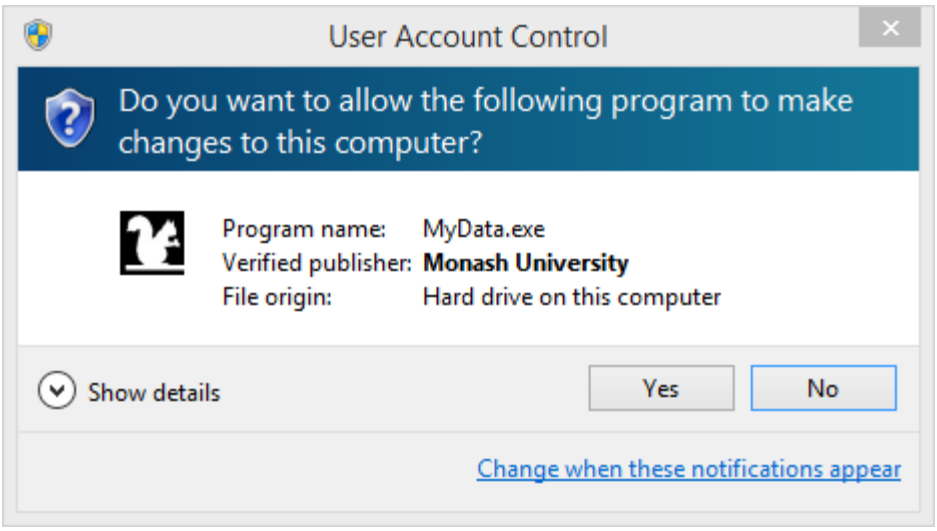
Clicking the "Lock" button displays the confirmation dialog below.



Once MyData's settings are locked, all of the fields in the Settings dialog will become read-only.



Clicking on the "Unlock" button will result in a request for administrator privileges.



Once administrator privileges have been verified, it will be possible to modify MyData's settings again.

N.B. This is NOT a security mechanism - it is a mechanism for preventing the accidental modification of settings in a production workflow. It does not prevent advanced users from determining where MyData saves its last used configuration to disk (e.g. C:\ProgramData\Monash University\MyData\MyData.cfg) and updating the settings outside of MyData.

1.4.6 Saving and Loading Settings

Each time you click OK on the Settings Dialog, your settings are validated, and then saved automatically to a location within your user home folder, which is OS-dependent, e.g. "C:\ProgramData\Monash University\MyData\MyData.cfg" or "/Users/jsmith/Library/Application Support/MyData/MyData.cfg".

The settings file is in plain-text file whose format is described here: https://docs.python.org/2/library/configparser.html. An example can be found here: MyDataDemo.cfg.

Any facilities with potentially malicious users may wish to consider what happens if a user gets hold of an API key for a facility role account, saved in a MyData configuration file. The API key cannot be used in place of a password to log into MyTardis's web interface, but it can be used with MyTardis's RESTful API to gain facility manager privileges. These privileges would not include deleting data, but for a technically minded user familiar with RESTful APIs, the API key could potentially be used to modify another user's data. Facilities need to decide whether this is an acceptable risk. Many facilities already use shared accounts on data-collection PCs, so the risk of one user modifying another user's data subdirectory is already there.

Settings can be saved to an arbitrary location chosen by the user by clicking Control-s (Windows) or Command-s (Mac OS X) from MyData's Settings dialog, keeping in mind the risks stated above. A saved settings file can then be dragged and dropped onto MyData's settings dialog to import the settings. This feature is currently used primarily by MyData developers for testing different configurations. It is expected that the MyData settings for each individual instrument PC will remain constant once the initial configuration is done.

1.4.7 Settings only configurable in MyData.cfg

The following settings do not appear in the Settings Dialog, but can be configured directly in MyData.cfg, which is typically found at a location like: "C:\ProgramData\Monash University\MyData\MyData.cfg" or "/Users/jsmith/Library/Application Support/MyData/MyData.cfg".

You should exit MyData before modifying MyData.cfg, and then restart it after saving your changes to MyData.cfg.

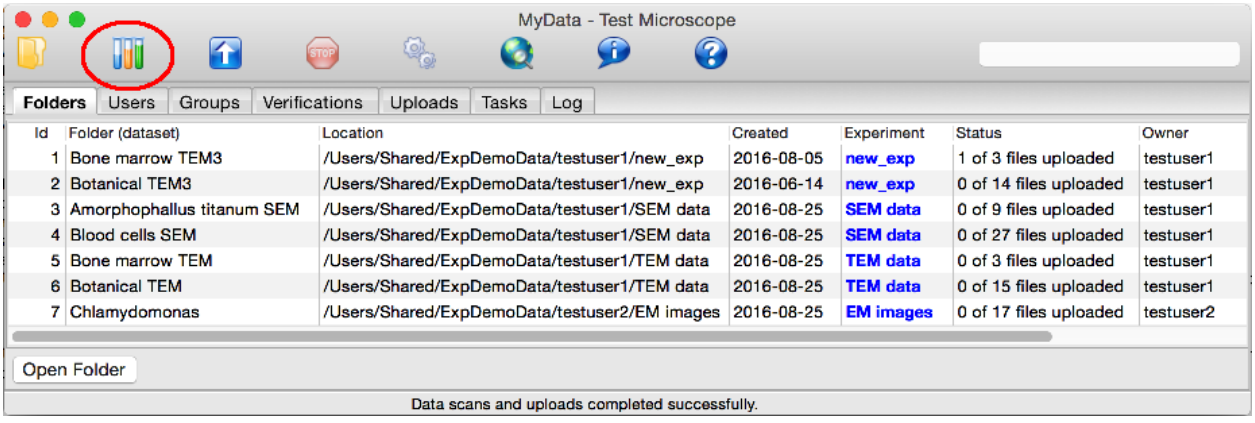
The "verification" in "max_verification_threads" below refers to the DataFile lookups performed by MyData to determine whether each local file has been previously uploaded to and verified on the MyTardis server, whereas the "verification" in "verification_delay" refers to MyData's request for MyTardis to verify that a newly uploaded file has the correct size and checksum.

Setting	Default value	Description
cache_datafile_lookups	True	Whether to cache results of successful datafile lookups
connection_timeout	10	Timeout (in seconds) used for HTTP responses and SSH connections
max_verification_threads	5	Maximum number of concurrent DataFile lookups
verification_delay	3	Upon a successful upload, MyData will request verification after a short delay (e.g. 3 seconds)
immutable_datasets	False	Whether datasets created by MyData should be read-only
progress_poll_interval	1	Interval in seconds between RESTful progress queries
cipher	aes128-gcm@openssh.com,aes128-ctr	Encryption cipher for SCP uploads
use_none_cipher	False	Use None cipher (only applicable for HPN-SSH)
fake_md5_sum	False	Skip MD5 calculation, and just send a string of zeroes
ignore_new_datasets	False	Ignore new datasets
ignore_new_interval_number	0	Number of intervals (e.g. months) to ignore for
ignore_new_interval_unit	months	Interval used for ignoring new datasets

1.5 Test Run

MyData's Test Run can give you a preview of what data would be uploaded before you begin uploading the data. This is particularly useful because MyTardis doesn't allow deleting of uploaded data, except by site administrators, so instead of trying a real upload, getting it wrong, and having to ask for help with cleaning up the unwanted data, you can do a practice run first.

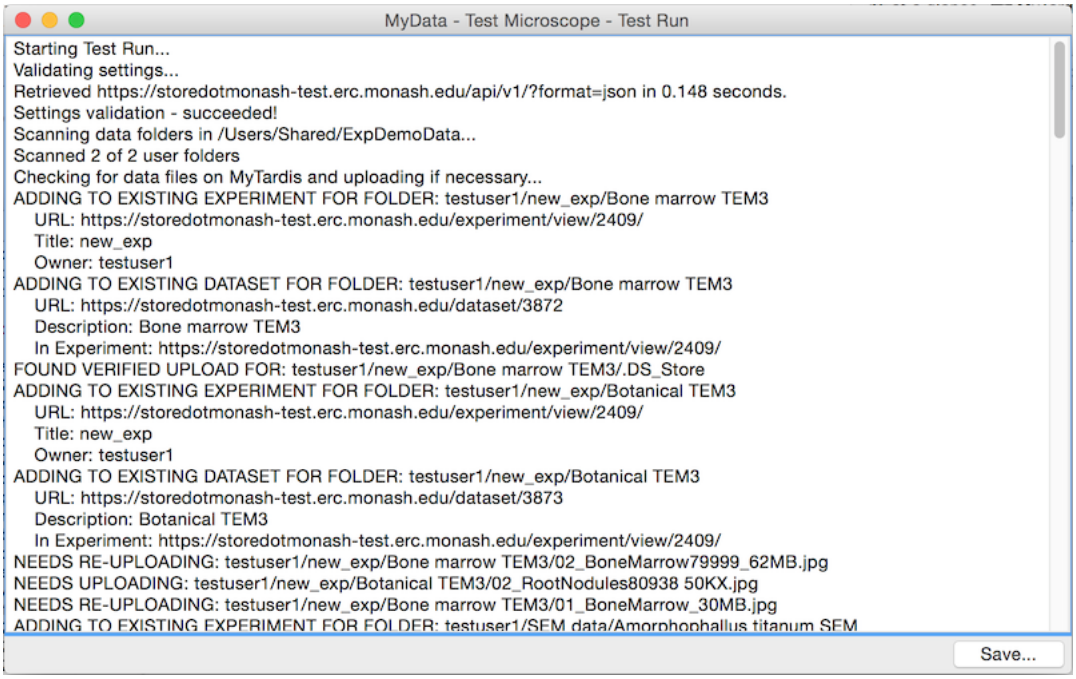
The Test Run can be launched from the "Test Tubes" icon on MyData's toolbar:



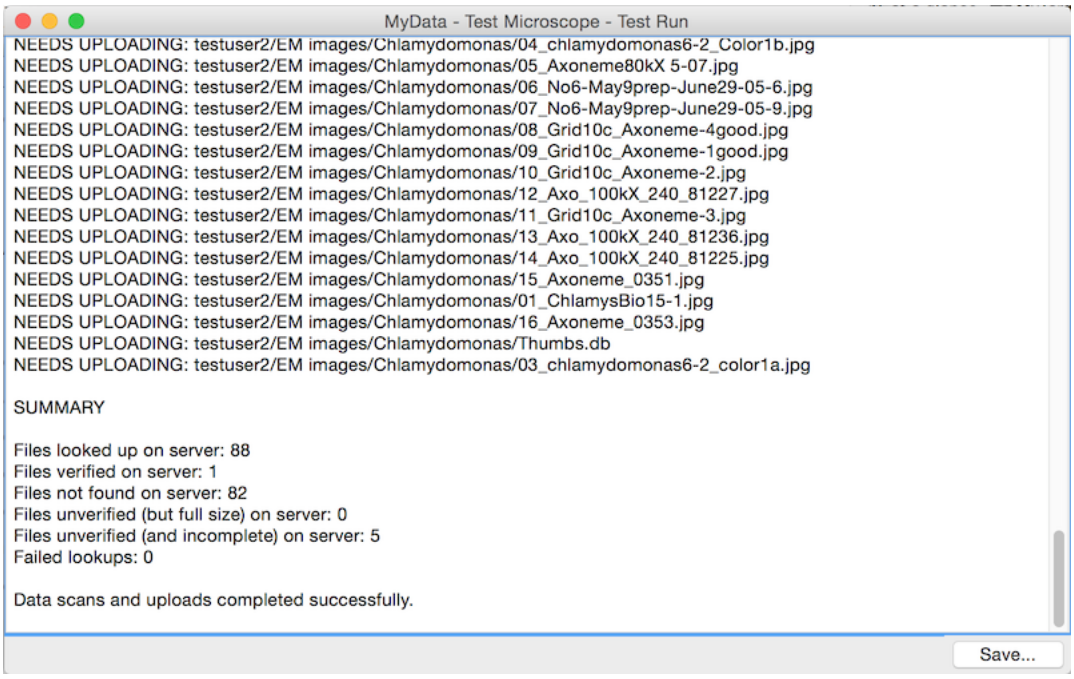
Id	Folder (dataset)	Location	Created	Experiment	Status	Owner
1	Bone marrow TEM3	/Users/Shared/ExpDemoData/testuser1/new_exp	2016-08-05	new_exp	1 of 3 files uploaded	testuser1
2	Botanical TEM3	/Users/Shared/ExpDemoData/testuser1/new_exp	2016-06-14	new_exp	0 of 14 files uploaded	testuser1
3	Amorphophallus titanum SEM	/Users/Shared/ExpDemoData/testuser1/SEM data	2016-08-25	SEM data	0 of 9 files uploaded	testuser1
4	Blood cells SEM	/Users/Shared/ExpDemoData/testuser1/SEM data	2016-08-25	SEM data	0 of 27 files uploaded	testuser1
5	Bone marrow TEM	/Users/Shared/ExpDemoData/testuser1/TEM data	2016-08-25	TEM data	0 of 3 files uploaded	testuser1
6	Botanical TEM	/Users/Shared/ExpDemoData/testuser1/TEM data	2016-08-25	TEM data	0 of 15 files uploaded	testuser1
7	Chlamydomonas	/Users/Shared/ExpDemoData/testuser2/EM images	2016-08-25	EM images	0 of 17 files uploaded	testuser2

The first part of the Test Run output shows the results of MyData's initial settings validation and data folder scanning.

This is followed by a list of the experiments and datasets which would be created (or added to), and a list of the files which would be uploaded:



The Test Run output finishes with a summary of the number of files which need to be uploaded.



The Test Run output can be saved to a text file for viewing in an external application.

1.6 Upload Methods

MyData supports two methods for uploading data to MyTardis:

1. HTTP POST
2. SCP via Staging

"HTTP POST" (MyData's default upload method) is automatically enabled as soon as you have entered some basic settings into MyData (see *Settings* and *Downloading and installing the demo configuration for MyData*). The "HTTP POST" method is easy to get up and running quickly for trying out MyData with small datasets.

But for large datasets and large datafiles, the "SCP via Staging" method is preferred for the following reasons:

1. For large datafile uploads (multiple gigabytes), the "HTTP POST" method can put significant strain on the MyTardis server's memory, affecting all users connecting to that MyTardis server.
2. Partially complete uploads can be resumed when using "SCP via Staging", but not when using "HTTP POST".
3. The "HTTP POST" method only allows one concurrent upload, because it uses the "poster" Python library, which uses "urllib2" which is not thread-safe.

1.6.1 Concurrent Upload Threads and Subprocesses launched by MyData

The maximum number of upload threads can be specified in the advanced tab of MyData's Settings Dialog (see *Advanced*). This setting will have no effect when using the "HTTP POST" upload method, which has a maximum of one concurrent upload.

When using multiple upload threads, you won't see multiple "MyData" processes running in your process monitor / task manager, but you will see multiple "scp" (secure copy) processes running which are launched from "MyData" as subprocesses. You may also see multiple "ssh" processes which are used to run remote commands on MyTardis's staging server to determine the size of an incomplete upload on MyTardis's staging server and to append an uploaded chunk to a partially uploaded datafile in MyTardis's staging area.

While using MyData's "SCP to Staging" upload method, you may also notice a "dd" subprocess running for each upload. "dd" is used to extract a chunk to upload from a datafile.

On Mac OS X, in addition to the brief "ssh" processes described above, you may also notice some lingering "ssh" processes (one per upload thread), which are used to set up a "ControlMaster" ssh process (see http://www.openbsd.org/cgi-bin/man.cgi?query=ssh_config), which allows MyData to reuse an existing SSH connection for appending additional datafile chunks to a partial upload.

OpenSSH's ControlMaster/ControlPath functionality is not available in Windows builds of OpenSSH: http://stackoverflow.com/questions/20959792/is-ssh-controlmaster-with-cygwin-on-windows-actually-possible
So we can't use this method to reuse SSH connections for SCP-uploading subsequent datafile chunks on Windows. Out of necessity, MyData creates a new SSH ("SCP") connection for every chunk uploaded, at least it does for large datafiles.

For small datafile uploads on Windows, if the chunk size is too small, then calling "scp.exe" repeatedly will waste time reconnecting to the same MyTardis staging server repeatedly after only spending a fraction of a second actually uploading each chunk. If the chunk size is too large, then MyData won't be able to display smooth progress updates. For small datafiles on Windows (less than 10 MB), MyData upload the entire datafile with one call to "scp.exe", so you won't see incremental progress updates in MyData's Uploads view.

1.6.2 HTTP POST

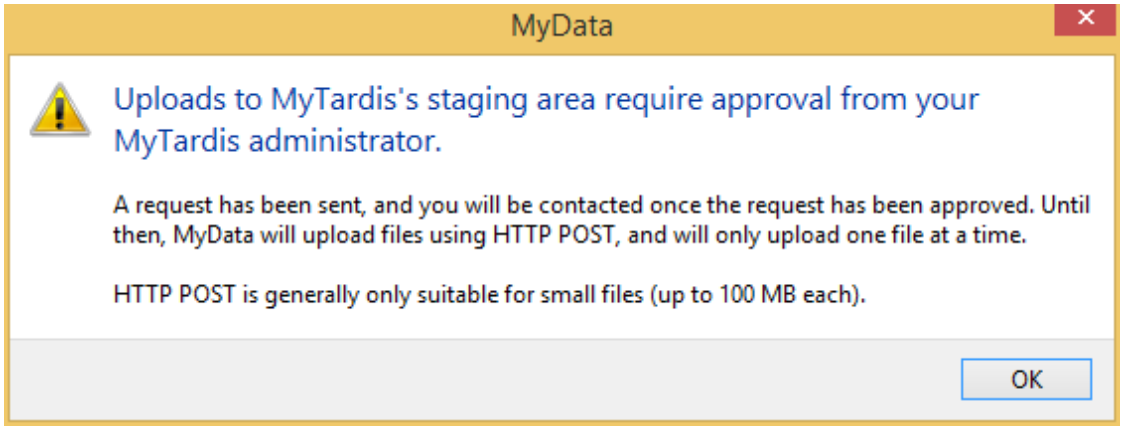
MyData's "HTTP POST" upload method uses the "Via multipart form POST" method of MyTardis's RESTful API. For more details, see: https://mytardis.readthedocs.org/en/latest/api.html#via-multipart-form-post

1.6.3 SCP to Staging

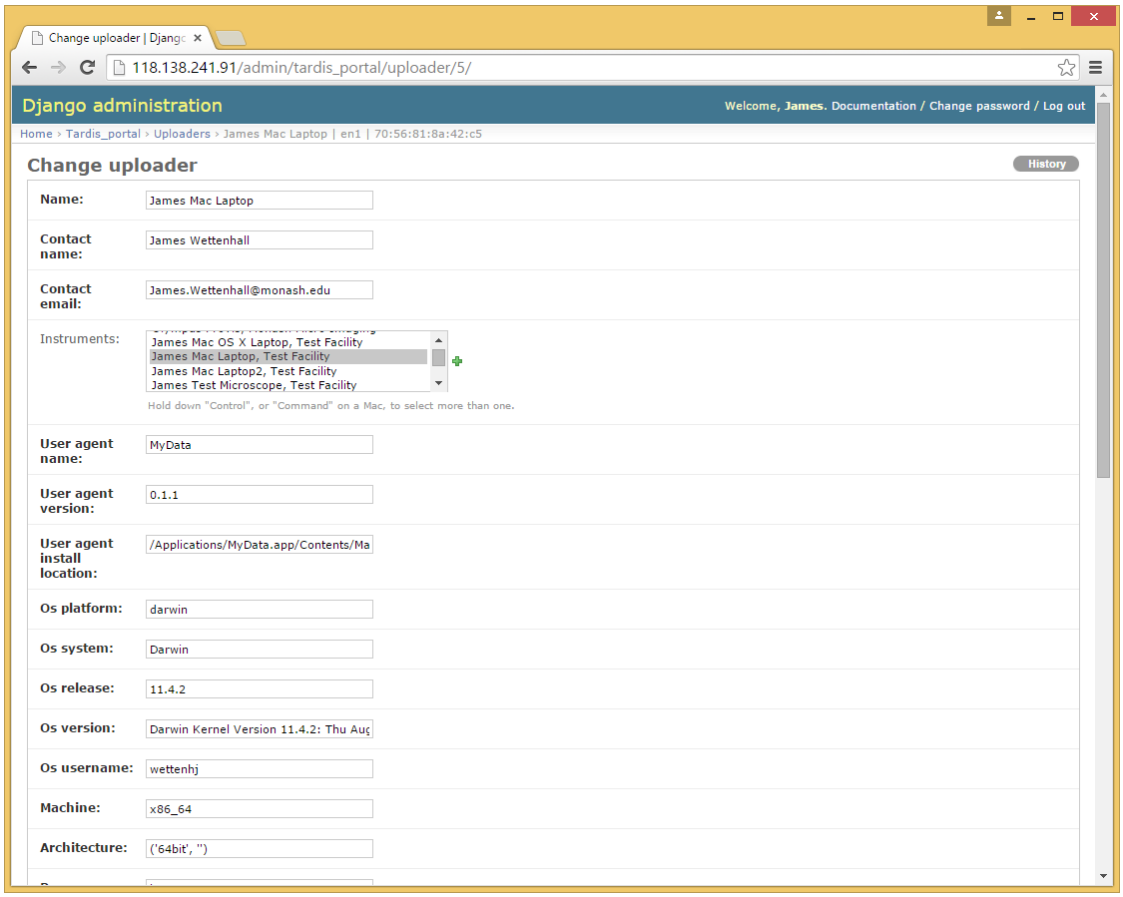
MyData's "SCP to Staging" upload method uses the "Via staging location" method of MyTardis's RESTful API. For more details, see: https://mytardis.readthedocs.org/en/latest/api.html#via-staging-location

When using the "SCP to Staging" method, MyData informs MyTardis of its intention to upload a datafile (and registers the filename, size and checksum in a Datafile record in MyTardis), and MyTardis then supplies MyData with a temporary location to upload the datafile to. MyData will be granted access to upload the datafile to that temporary location using scp (secure copy). The server which MyData connects when uploading via scp (known as the "staging server") may be the same as the MyTardis server, or it may be a different server which mounts the same storage as MyTardis. MyTardis runs scheduled background tasks to check for datafiles which have been registered but not yet verified, and for unverified datafiles which were to be uploaded via staging, MyTardis will copy the uploaded datafile from the staging area to its final destination (MyTardis's file store) while checking its size and calculating its MD5 checksum to verify its integrity.

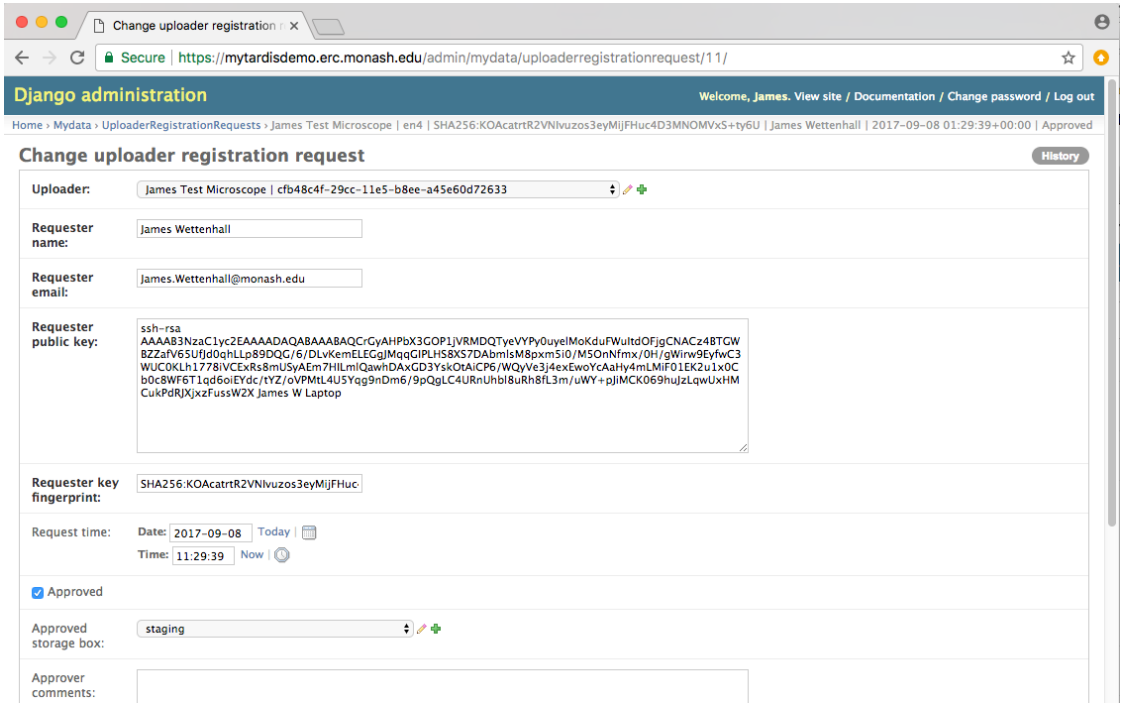
The first time a user runs MyData, they wil see a warning indicating that MyData's preferred upload method (SCP via staging) hasn't yet been approved by the MyTardis administartor:



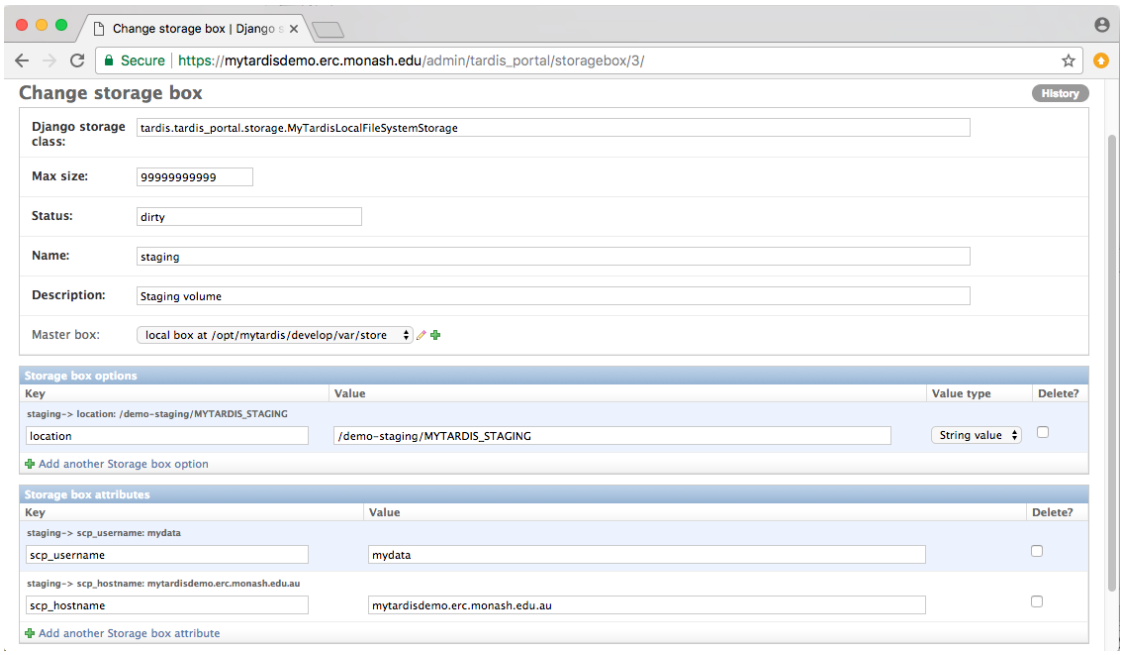
MyData uploads some basic information about the instrument PC and about the MyData installation to its MyTardis server. This basic information is called an "uploader" record. Once an uploader record has been created in MyTardis, no other users (of MyTardis's RESTful API) will be able to access the uploader record unless they know its MAC address, a unique string associated with the MyData user's active network interface (Ethernet or WiFi). A single MyData user could create multiple uploader records from each PC they run MyData on, one for each network interface on each PC.



The MyTardis administrator can approve the request in the Django Admin interface (after adding the public key to the appropriate /home/mydata/.ssh/authorized_keys file):



While approving the request, the MyTardis administrator can assign an appropriate storage box to the MyData uploader. In this case, the "staging" storage box has been selected. This storage box has attributes for "scp_username" and "scp_hostname" (and optionally "scp_port" which defaults to 22). It is the MyTardis administrator's responsibility to paste the public key into the appropriate authorized_keys file to allow MyData to upload without a password.



Below is a sample of a MyTardis administrator's notes made (in the approval_comments field in MyTardis's UploadRegistrationRequest model) when approving one of these upload requests:

Ran the following as root on the staging host (118.138.241.33) :

```
$ adduser mydata
$ mkdir /home/mydata/.ssh
$ echo "ssh-rsa AAAAB3NzaC... MyData Key" > /home/mydata/.ssh/authorized_keys
$ chown -R mydata:mydata /home/mydata/.ssh/
$ chmod 700 /home/mydata/.ssh/
$ chmod 600 /home/mydata/.ssh/authorized_keys
$ usermod -a -G mytardis mydata
```

The MyData client will need to create subdirectories within the MyTardis staging area, and it will need to be able to write within those subdirectories. The "mytardis" web user should have read access to the staging data, but the "mydata" user should not have write access to the permanent storage location.

To ensure that MyTardis's Celery processes (running under the "mytardis" account) have access to the uploaded files via group ownership, we can set the "setgid" bit (chmod g+s or chmod 2770) on the staging directory so that all files created within staging can inherit the "mytardis" group.

Previous versions of MyData's documentation recommended using "umask" to ensure that files uploaded by MyData were group readable (so that MyTardis could verify them) and group writeable (so that MyTardis could move them to their permanent storage box). However, from v0.7.0, MyData explicitly sets permissions on files it uploads (and on subddirectories it creates), instead of assuming that umask has been configured to do this automatically.

N.B.: The test below was only possible because the MyData user submitting the request and the MyTardis administrator approving the request were the same person. Normally, the MyTardis administrator wouldn't have access to the MyData user's private key.

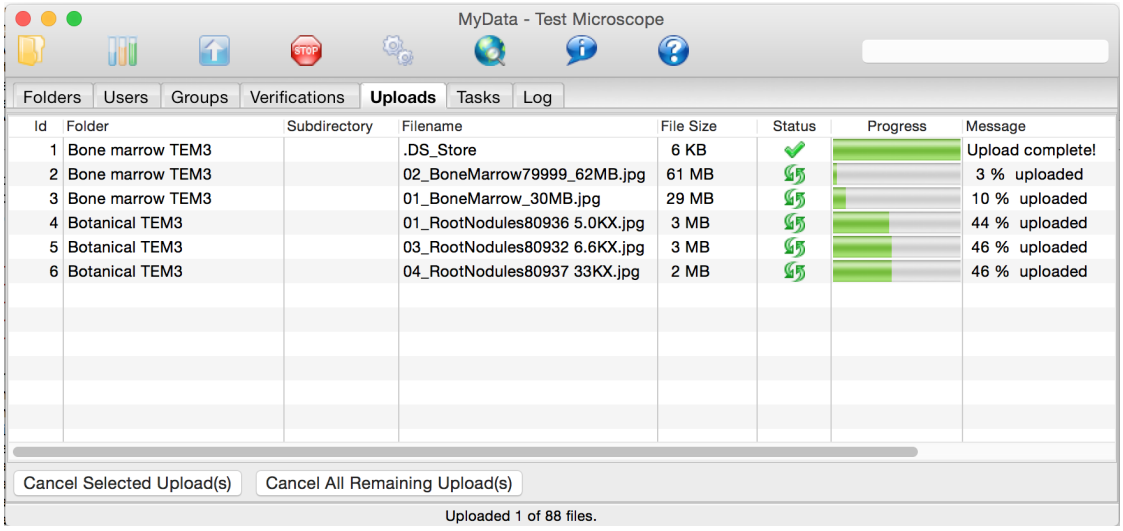
Because in this case, I had access to the private key generated by the MyData instance submitting the uploader registration request, I was able to test SSHing into the staging host from my MyData test machine using the SSH private

key which MyData generated in ~/.ssh/:

```
$ ssh -i ~/.ssh/MyData mydata@118.138.241.33
[mydata@118.138.241.33 ~]$ groups
mydata mytardis
[mydata@118.138.241.33 ~]$ ls -lh /mnt/sonas/market | grep MYTARDIS
drwx------ 403 mytardis mytardis 128K Nov 12 14:33 MYTARDIS_FILE_STORE
drwxrws---   3 mytardis mytardis  32K Nov 13 15:36 MYTARDIS_STAGING
[mydata@118.138.241.33 ~]$ touch /mnt/sonas/market/MYTARDIS_STAGING/test123.txt
[mydata@118.138.241.33 ~]$ rm /mnt/sonas/market/MYTARDIS_STAGING/test123.txt
```

Note the permissions above - being part of the "mytardis" group on this staging host allows the "mydata" user to write to the MYTARDIS_STAGING directory, but not to the MYTARDIS_FILE_STORE directory. The 's' in the "MYTARDIS_STAGING" directory permission ensures that all subdirectories created by the "mydata" user will inherit the MYTARDIS_STAGING directory's group ("mytardis"), instead of the mydata user's default group ("mydata").

Once uploads to staging have been approved, MyData can manage multiple uploads at once (5 by default):

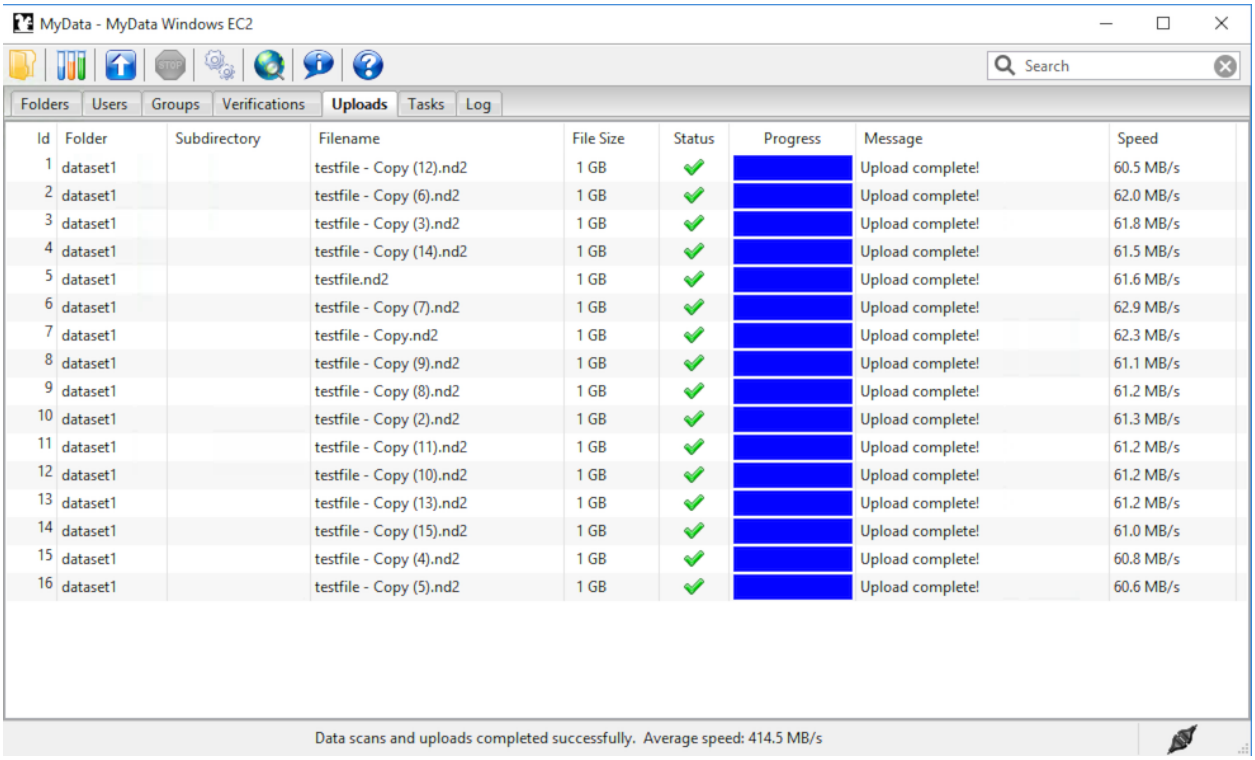


1.7 Upload Speed

How fast is MyData?

This page lists the key considerations for achieving faster uploads with MyData.

An example of MyData uploading at faster than Gigabit speeds is shown below:



Id	Folder	Subdirectory	Filename	File Size	Status	Progress	Message	Speed
1	dataset1		testfile - Copy (12).nd2	1 GB	✓		Upload complete!	60.5 MB/s
2	dataset1		testfile - Copy (6).nd2	1 GB	✓		Upload complete!	62.0 MB/s
3	dataset1		testfile - Copy (3).nd2	1 GB	✓		Upload complete!	61.8 MB/s
4	dataset1		testfile - Copy (14).nd2	1 GB	✓		Upload complete!	61.5 MB/s
5	dataset1		testfile.nd2	1 GB	✓		Upload complete!	61.6 MB/s
6	dataset1		testfile - Copy (7).nd2	1 GB	✓		Upload complete!	62.9 MB/s
7	dataset1		testfile - Copy.nd2	1 GB	✓		Upload complete!	62.3 MB/s
8	dataset1		testfile - Copy (9).nd2	1 GB	✓		Upload complete!	61.1 MB/s
9	dataset1		testfile - Copy (8).nd2	1 GB	✓		Upload complete!	61.2 MB/s
10	dataset1		testfile - Copy (2).nd2	1 GB	✓		Upload complete!	61.3 MB/s
11	dataset1		testfile - Copy (11).nd2	1 GB	✓		Upload complete!	61.2 MB/s
12	dataset1		testfile - Copy (10).nd2	1 GB	✓		Upload complete!	61.2 MB/s
13	dataset1		testfile - Copy (13).nd2	1 GB	✓		Upload complete!	61.2 MB/s
14	dataset1		testfile - Copy (15).nd2	1 GB	✓		Upload complete!	61.0 MB/s
15	dataset1		testfile - Copy (4).nd2	1 GB	✓		Upload complete!	60.8 MB/s
16	dataset1		testfile - Copy (5).nd2	1 GB	✓		Upload complete!	60.6 MB/s

An overall upload speed of 414.5 MB/s was achieved by MyData v0.7.0-beta1 while uploading sixteen 1 GB files. MyData was configured to run with 8 concurrent upload threads on a d2.8xlarge Amazon EC2 Windows virtual machine with 36 CPU cores, and was uploading to a d2.8xlarge EC2 Ubuntu virtual machine, also with 36 CPU cores, using MyData's SCP upload method.

The "Average speed" of 414.5 MB/s displayed in MyData's status bar was calculated from the total elapsed time, including the time taken to calculate an MD5 sum before each upload.

The "raw" upload speed (excluding elapsed time for MD5 calculations) can be determined by adding the individual upload speeds (shown in the "Speed" column) for the first 8 uploads (which were performed concurrently), giving 494 MB/s.

Each Amazon EC2 virtual machine had 36 *Intel(R) Xeon(R) CPU E5-2676* CPU cores.

1.7.1 Upload Methods

MyData currently offers two upload methods: POST (to MyTardis) and SCP. Upload methods which may be offered in the future include globus-url-copy, Robocopy to Samba/CIFS, and POSTing to object storage (S3 / Swift).

POST (to MyTardis) can be used with either HTTP or HTTPS, with HTTPS being recommend for all production MyTardis servers. The POST protocol can perform very well, e.g. for uploads to S3 object storage, but the POST uploads offered by MyTardis are usually restricted to small files due to server memory usage of Django / TastyPie / gunicorn. Furthermore, POST uploads from MyData don't allow concurrent upload threads because the "poster" Python module being used to display upload progress is not thread-safe.

SCP is the recommended upload method for MyData currently. Because of the current restrictions on POSTing large files to MyTardis, this document will focus exclusively on optimizing upload speed using the SCP upload method.

1.7.2 SCP Server Load

Overview

Modern top-of-the-line CPUs can encrypt and decrypt very quickly, but you shouldn't assume that your CPUs can encrypt / decrypt at wire speed, especially if the number of concurrent SCP uploads your server is handling is greater than the number of CPU cores on the server.

A simple benchmark

The following experiment, inspired by this blog article: https://blog.famzah.net/2015/06/26/openssh-ciphers-performance-benchmark-update-2015/ runs some encryption/decryption tests on a single-CPU *Intel Xeon E312xx* virtual machine, using the "aes128-gcm@openssh.com" cipher, while varying the number of concurrent SSH processes.

Here is the CPU model name and the number of CPU cores / processors:

```
$ grep "model name" /proc/cpuinfo
model name  : Intel Xeon E312xx (Sandy Bridge)

$ grep -c ^processor /proc/cpuinfo
1
```

Here's the SSH version we're using:

```
$ ssh -V
OpenSSH_6.6.1p1 Ubuntu-2ubuntu2.6, OpenSSL 1.0.1f 6 Jan 2014
```

Here is the bash script we will be running:

```
$ cat aes128-gcm
#!/bin/bash

# Inspired by https://blog.famzah.net/2015/06/26/openssh-ciphers-performance-
↪benchmark-update-2015/

num_threads=$1
cipher=aes128-gcm@openssh.com
dir=./"dd_${num_threads}_threads"
mkdir -p $dir
for thread in `seq 1 $num_threads`
do
  dd if=/dev/zero bs=4M count=256 2>$dir/dd_$thread | ssh -c $cipher localhost 'cat >␣
↪/dev/null' &
done

for job in `jobs -p`
do
  wait $job
done
```

Now let's run the script and view the results:

```
$ for numthreads in 1 2 4 8; do ./aes128-gcm $numthreads; done

$ find . -name "dd*" -type f -exec grep -H copied {} \; | sort
./dd_1_threads/dd_1:1073741824 bytes (1.1 GB) copied, 7.53023 s, 143 MB/s
./dd_2_threads/dd_1:1073741824 bytes (1.1 GB) copied, 15.4659 s, 69.4 MB/s
```

(continues on next page)

(continued from previous page)

```
./dd_2_threads/dd_2:1073741824 bytes (1.1 GB) copied, 15.4916 s, 69.3 MB/s
./dd_4_threads/dd_1:1073741824 bytes (1.1 GB) copied, 31.5267 s, 34.1 MB/s
./dd_4_threads/dd_2:1073741824 bytes (1.1 GB) copied, 31.6224 s, 34.0 MB/s
./dd_4_threads/dd_3:1073741824 bytes (1.1 GB) copied, 31.6511 s, 33.9 MB/s
./dd_4_threads/dd_4:1073741824 bytes (1.1 GB) copied, 31.7058 s, 33.9 MB/s
./dd_8_threads/dd_1:1073741824 bytes (1.1 GB) copied, 64.7115 s, 16.6 MB/s
./dd_8_threads/dd_2:1073741824 bytes (1.1 GB) copied, 65.2428 s, 16.5 MB/s
./dd_8_threads/dd_3:1073741824 bytes (1.1 GB) copied, 65.3309 s, 16.4 MB/s
./dd_8_threads/dd_4:1073741824 bytes (1.1 GB) copied, 65.1312 s, 16.5 MB/s
./dd_8_threads/dd_5:1073741824 bytes (1.1 GB) copied, 65.3107 s, 16.4 MB/s
./dd_8_threads/dd_6:1073741824 bytes (1.1 GB) copied, 65.2225 s, 16.5 MB/s
./dd_8_threads/dd_7:1073741824 bytes (1.1 GB) copied, 65.2411 s, 16.5 MB/s
./dd_8_threads/dd_8:1073741824 bytes (1.1 GB) copied, 65.1053 s, 16.5 MB/s
```

It is clear that as we increase the number of concurrent SSH processes from 1 to 8, the encryption / decryption speed decreases significantly.

Recommendations

1. Use a monitoring tool like Nagios to check the number of concurrent SSH (or SCP) processes on your SCP server(s), and consider load balancing e.g. using HAProxy.
2. Check MyData's max_upload_threads setting configured by your users (visible to MyTardis administrators in the UploaderSettings model), and ensure that users are not trying to run more upload threads than the number of CPUs on their machine.
3. The "scp_hostname" storage box attribute configured by MyTardis administrators for MyData uploads doesn't need to be the same as your MyTardis server's hostname. You can use a different server with more CPUs and with a more recent version of OpenSSH, as long as it can mount the same storage as your MyTardis server (e.g. using NFS).

1.7.3 Max Upload Threads

Overview

MyData can be configured to upload multiple files concurrently. The maximum number of concurrent uploads can be configured in the Advanced tab of MyData's Settings dialog.

Recommendations

1. Do not set MyData's maximum upload threads to be greater than the number of CPU cores available on the SCP server(s) MyData is uploading to.
2. Do not set MyData's maximum upload threads to be greater than the number of CPU cores available on the client machine running MyData.
3. If multiple CPU cores are available on both the client machine and on the SCP server(s), running multiple concurrent upload threads in MyData can improve overall throughput when single-channel SCP speed is limited by an encryption bottleneck.

1.7.4 SSHFS Mounts

Overview

If encryption/decryption is a bottleneck, using SSHFS storage mounts on your SCP server can slow down write speeds.

Recommendations

1. Run some write speed tests using "dd":

```
$ dd if=/dev/zero of=/NFS_mount/test.img bs=1G count=1 oflag=dsync
1+0 records in
1+0 records out
1073741824 bytes (1.1 GB) copied, 5.67731 s, 189 MB/s

$ dd if=/dev/zero of=/SSHFS_mount/test.img bs=1G count=1 oflag=dsync
1+0 records in
1+0 records out
1073741824 bytes (1.1 GB) copied, 19.1225 s, 56.2 MB/s
```

2. Try different ciphers with SSHFS, e.g. "-o Ciphers=aes128-gcm@openssh.com". The aes128-gcm@openssh.com is usually one of the fastest if you have AES-NI. If you have really old CPUs without AES-NI, then the fastest ciphers are usually the "arcfour" family. See "man ssh_config" for a full list of Ciphers available to your SSH version. After changing the cipher (and restarting SSHFS if necessary), run "dd" again:

```
$ dd if=/dev/zero of=/SSHFS_mount/test.img bs=1G count=1 oflag=dsync
1+0 records in
1+0 records out
1073741825 bytes (1.1 GB) copied, 14.4593 s, 74.3 MB/s
```

1.7.5 SSH/SCP Ciphers

Overview

A cipher is an algorithm for encrypting or decrypting data. If you are using recent top-of-the-line PCs at both ends of your SCP transfer and you are operating on a Gigabit (or slower) network, then it doesn't matter which cipher you use for SCP transfers - the default cipher should easily be able to encrypt at "wire speed", i.e. as fast as your Network can transfer the data.

However, if you have older / cheaper CPUs on at least one end of your SCP transfer and/or a fast network (Gigabit or 10 Gigabit), then encryption and/or decryption could easily become a bottleneck, and using the best cipher (and a recent OpenSSH version) can make a big difference.

Recommendations

1. On your SCP server, you can run a benchmark like this one: https://blog.famzah.net/2015/06/26/openssh-ciphers-performance-benchmark-update-2015/ to determine which ciphers perform best for you. If you have AES-NI, then the fastest ciphers are usually aes128-gcm@openssh.com and aes256-gcm@openssh.com. If you have old CPUs without AES-NI, then the fastest ciphers are the "arcfour" ciphers. Here are some results from an *Intel Xeon E312xx (Sandy Bridge)* single-CPU virtual machine:

Cipher	Speed
aes128-gcm@openssh.com	140 MB/s
aes256-gcm@openssh.com	133 MB/s
aes128-ctr	103 MB/s
arcfour	82.3 MB/s
blowfish-cbc	35.0 MB/s

2. If you are running MyData v0.7.0 or later, you can set the cipher in MyData.cfg. From v0.7.0 onwards, MyData's default cipher on Windows is aes128-gcm@openssh.com,aes128-ctr. Having multiple ciphers separated by a comma means that the SSH / SCP client will request the first one, and if the server rejects it, then the second one will be used. On Mac and Linux, MyData doesn't bundle its own SSH / SCP binaries, so the default cipher is aes128-ctr, which is available in older versions of OpenSSH.
3. MyTardis administrators can set the scp_hostname storage box attribute for MyData uploads, so if you want MyData to upload to an SCP server with a more recent OpenSSH version than what you have on your MyTardis server, supporting additional ciphers, that is no problem.

1.7.6 Hardware-Accelerated Encryption (AES-NI)

Overview

Modern CPUs offer hardware-accelerated AES encryption (AES-NI), which makes encryption/decryption must faster, especially when using the AES ciphers. The aes128-gcm@openssh.com and aes256-gcm@openssh.com are usually the fastest ciphers on machines on AES-NI hardware. If using older SSH versions which do not support these ciphers, aes128-ctr, aes192-ctr and aes256-ctr also perform very well on AES-NI hardware. On older CPUs which do not support AES-NI, the fastest ciphers are usually arcfour, arcfour128, arcfour256 and blowfish-cbc. Running a benchmark like the one in the following blog articles can help to determine if AES-NI is working (AES ciphers should be fast) or if it is not supported (in which case the arcfour and blowfish ciphers may perform better than the AES ciphers).

- https://blog.famzah.net/2015/06/26/openssh-ciphers-performance-benchmark-update-2015/

On Linux, you can determine if AES encryption is supported by your CPU using:

```
$ cat /proc/cpuinfo | grep aes
```

Whilst this is the simplest way, it is not guaranteed to be accurate. Intel says:

> "The Linux /proc/cpuinfo/ command does not accurately detect if Intel® AES-NI is enabled or disabled on the hardware. CPUID (http://www.etallen.com/cpuid/) tool can be used to make accurate determination." https://software.intel.com/sites/default/files/m/d/4/1/d/8/AES-NI_Java_Linux_Testing_Configuration_Case_Study.pdf

On Windows, you can use one of the following tools to check whether your CPU(s) have AES-NI support:

- http://www.cpuid.com/softwares/cpu-z.html
- https://www.grc.com/securable.htm

However, having hardware-support for AES-NI doesn't necessarily means that your SSH/SCP software supports it!

On Linux, it is generally a safe bet that if hardware support is available, then AES-NI will be available in the installed OpenSSH software.

However on Windows, only some SSH/SCP clients claim to support AES-NI:

- https://en.wikipedia.org/wiki/Comparison_of_SSH_clients#Features

And of those SSH/SCP clients which do claim to support it, some of them don't offer the full range of ciphers available in the latest OpenSSH versions. For example, not many Windows SSH/SCP clients (except for Cygwin OpenSSH) support aes128-gcm@openssh.com and aes256-gcm@openssh.com. The best way to determine whether AES-NI is working is to compare speeds between an AES cipher which is supported by the SSH/SCP client (e.g. aes128-ctr) with one of the older ciphers (e.g. arcfour or blowfish-cbc). If the AES cipher doesn't perform significantly better than the arcfour or blowfish-cbc, or if you are getting encryption speeds well below 100 MB/s, then AES-NI probably isn't working.

Recommendations

1. Run some encryption benchmarks like those in the blog article linked below to isolate encryption speed (as distinct from storage I/O speed or network bandwidth). - https://blog.famzah.net/2015/06/26/openssh-ciphers-performance-benchmark-update-2015/

1.7.7 Lots of Tiny Files

Overview

MyData is not very efficient at uploading thousands of tiny files. For each file it finds, it does a MyTardis API query to check whether the file has already been uploaded, then it calculates the file's MD5 sum, then it calls MyTardis's API again to create a DataFile record.

Future versions of MyData may perform combined API queries for groups of files, and upload them with a single call to SCP or SFTP. The challenge here is that asking MyTardis whether a group of files needs to be uploaded can result in "yes", "no" or "some of them".

Recommendations

1. If you have thousands of tiny files you want to upload, then it is more efficient to create a ZIP or TAR archive before uploading them.
2. If you find that MyData is taking a long time to verify previous uploads of a large number of tiny files, you could try the following: (i) Move folders of previously-uploaded files outside of the directory being scanned by MyData. (ii) Use MyData's "Ignore datasets older than" filter to ignore dataset folders with old created dates. (iii) Measure how long it takes to get a basic response from your MyTardis API, using https://mytardis.example.com/api/v1/?format=json - and if it is slow, consider putting more grunt (CPUs / gunicorn processes) behind your MyTardis API. (iv) If using MyData v0.6.3 or later, try adjusting max_verification_threads in your MyData.cfg

1.7.8 MD5 Checksums

Overview

Whilst it is best to check for bottlenecks on your servers (MyTardis and SCP) first (because they affect all of your users), you should also consider whether MyData's MD5 checksum calculation before each upload is adding significant overhead. This depends on the CPUs on the MyData client machine.

Recommendations

1. Ask any users experiencing slow MyData uploads to check MyData's Uploads view and report whether they see the "Calculating MD5 checksum" message and progress bar for significant durations.

2. Where MD5 checksums are slow, consider running MyData on a more up-to-date PC if possible.

3. If using MyData v0.7.0 or later, try setting fake_md5_sums to True in MyData.cfg to skip the MD5 sum calculation in order to measure the overall difference in upload speed. Don't forgot to change it back to False or remove it from MyData.cfg when you have finished testing!

4. Request (from the MyData developers) MD5 sum calculations in parallel with uploads. MyData can already upload with a fake MD5 sum, but it doesn't yet have the functionality to update the DataFile record with the corrected MD5 sum when available.

1.7.9 MyData v0.7.0 Enhancements

Overview

There are number of enhancements in MyData v0.7.0 which improve upload speeds. The most significant enhancement for upload speed is the scrapping of MyData's file chunking. Prior to v0.7.0, MyData split large files up into chunks and uploaded one at a time, and then joined them together on the SCP server. This added significant overhead, so it has been removed in v0.7.0.

The potential gotchas of upgrading to v0.7.0 are that aborted partial uploads cannot be resumed, progress updates might not be as smooth, and your MyTardis administrator will need to ugrade your MyTardis server to use the latest version of of MyData's server-side app, available at https://github.com/mytardis/mytardis-app-mydata. Also, if your MyTardis server's filesystem uses caching (e.g. SSHFS), then it's possible for MyData's progress queries to get inconsistent results from the MyTardis API, depending on which web worker node responds to the query.

Recommendations

1. Please help with beta testing MyData v0.7.0 beta and let us know what you think of its performance and report any bugs, either at https://github.com/mytardis/mydata/issues or at store.star.help@monash.edu. It is available at https://github.com/mytardis/mydata/releases

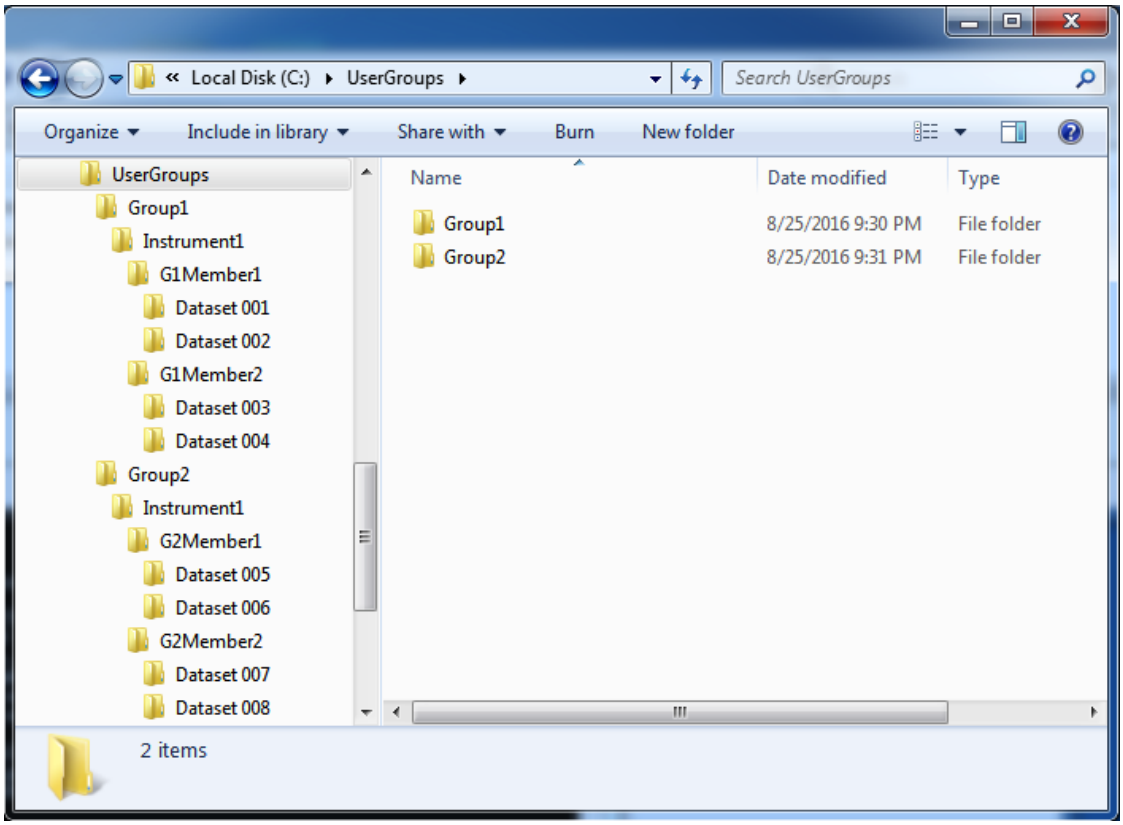
1.8 User Groups

Assigning access to datasets to user groups is an alternative to assigning access to individual users. A folder structure of "User Group/Instrument/Data Owner Full Name/Dataset" is supported for this purpose. As well as being used by MyData, this folder structure can be used to copy / sync data to a shared network drive. Each instrument PC will only have one instrument folder, being the name of that instrument, but when data from multiple instrument PCs is copied to the shared network drive, multiple instrument folders can appear alongside each other. The "Data Owner Full Name" folder is usually the name of the person who collected the data, but it is really just a way of grouping datasets into MyTardis experiments, i.e. it is not used to assign access control.

For more information, see the "Folder Structure - User Group / Instrument / Full Name / Dataset" section in http://mydata.readthedocs.org/en/latest/settings.html#advanced

1.8.1 Data Uploads from Instrument PCs

When using User Groups, the primary data directory used with MyData could look like this:

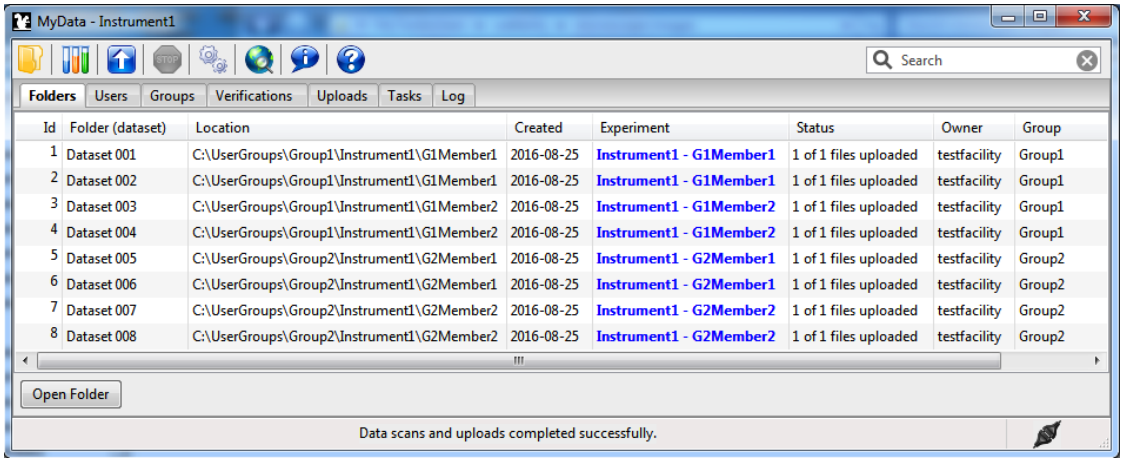


The first folder level within C:\UserGroups ("Group1", "Group2" etc.) is a user group defined in MyTardis. The actual group names in MyTardis may have an additional prefix ("TestFacility-") prepended to the folder name, i.e. "TestFacility-Group1", "TestFacility-Group2" etc.

The second folder level within C:\UserGroups ("Instrument 1") is the name of the data collection instrument. This folder may seem redundant, because all of the data on each instrument PC is by definition, on the same instrument PC (e.g. "Instrument 1"), but this folder level becomes useful when data from multiple instrument PCs is synced to a shared network drive.

The third folder level within C:\UserGroups ("G1Member1", "G1Member2", "G2Member1", "G2Member2" etc.) is usually the full name of the researcher who owns the data, but in some cases it is just an arbitrary collection of datasets. This corresponds to an experiment in MyTardis, which is a collection of datasets which can be made accessible to a particular user or to a group (e.g. "TestFacility-Group1").

The fourth folder level within C:\UserGroups ("Dataset001" etc.) is mapped to a MyTardis dataset.

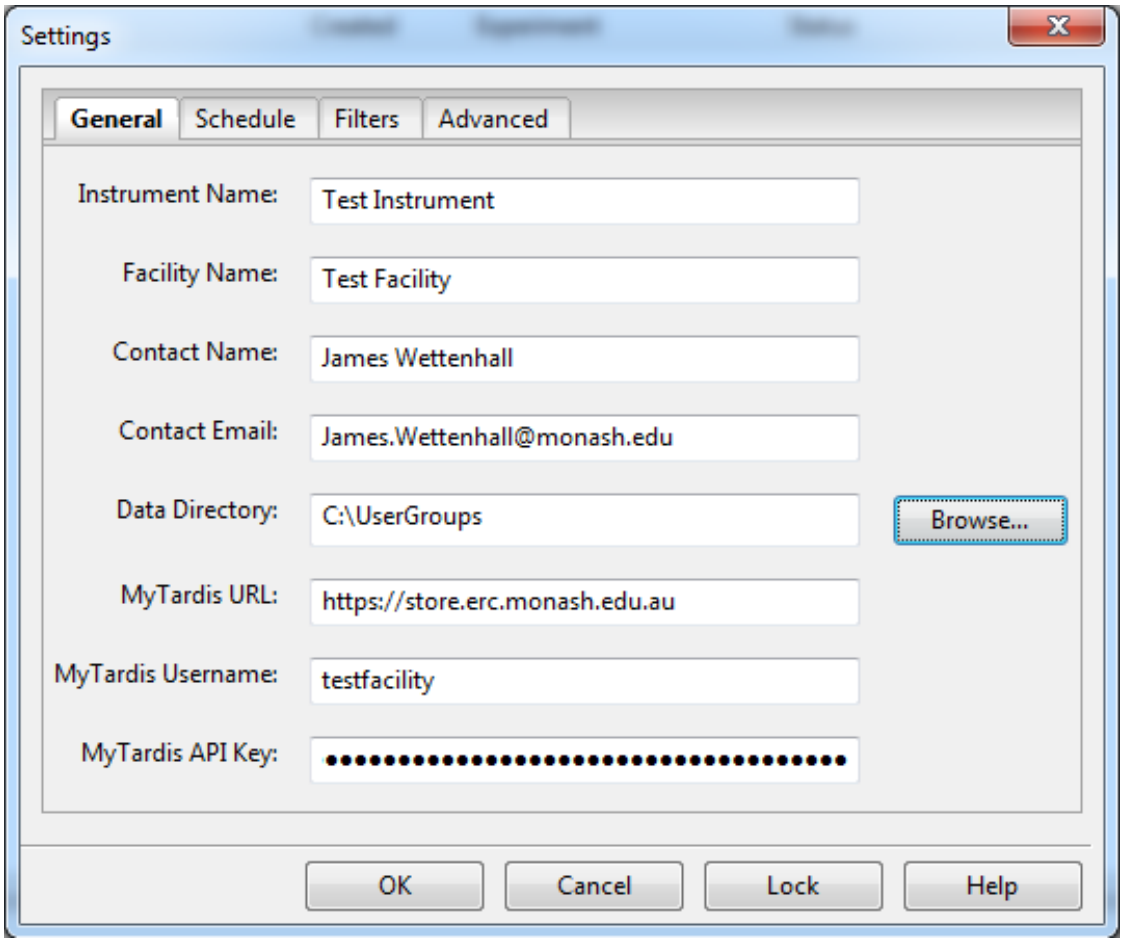


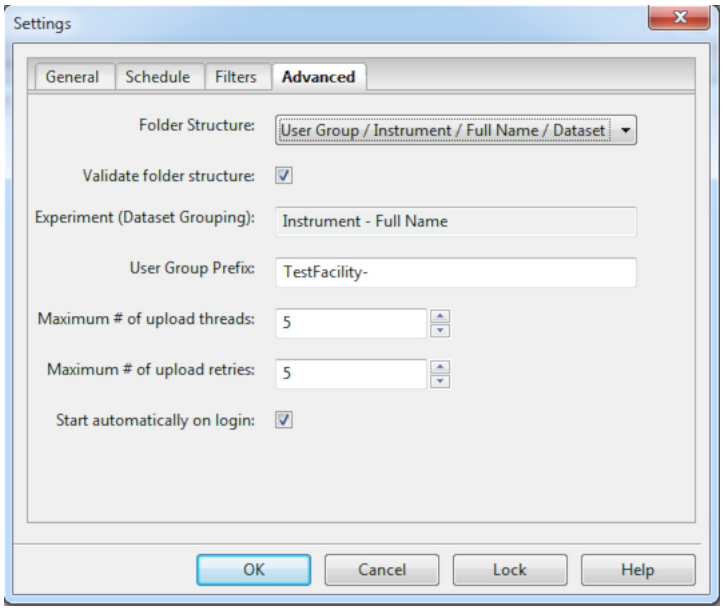
The MyData screenshot shows the 8 datasets found within the C:\UserGroups directory on the "Instrument 1" PC. MyData counts the number of files within each dataset folder on the local disk, then counts the number of files previously uploaded to VicNode / MyTardis for that dataset (if any), and then uploads any datafiles which are not already available on VicNode / MyTardis.

Whilst MyData can recognize old data in a well-defined folder structure as described above, MyData is generally intended to be used to upload recently acquired data. An option to ignore old datasets (older than 6 months) has recently been implemented in MyData.

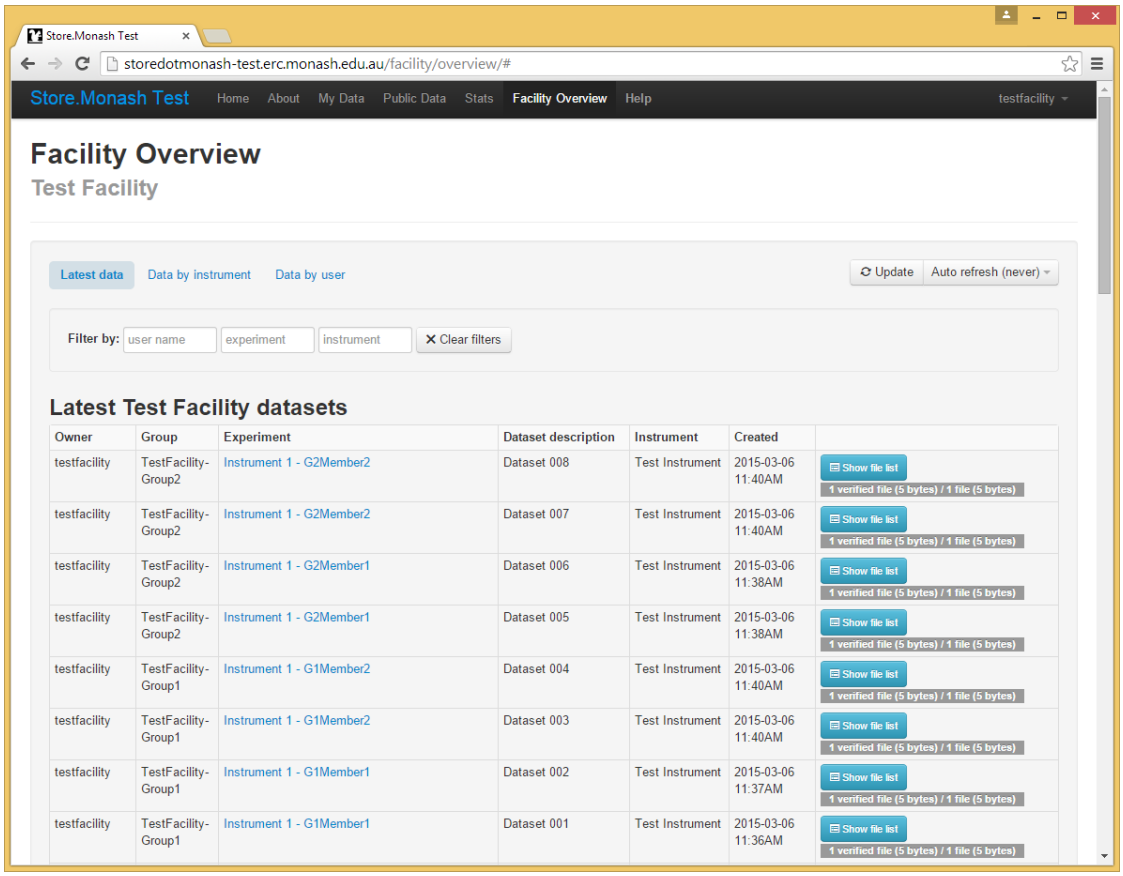
1.8.2 Data Management in MyTardis for Facility Managers

The first time MyData is run on a new instrument PC, some configuration is required - MyDatas Settings dialog is shown below. Typically a facility role account in MyTardis ("testfacility" in this case) is used to upload data). Once the data has been uploaded, access (and ownership) can be granted to individual users within MyTardis. In the case of the User Group folder structure, MyData will attempt to automatically grant read access to each dataset to all users within the data set's user group (e.g. "TestFacility-Group1").



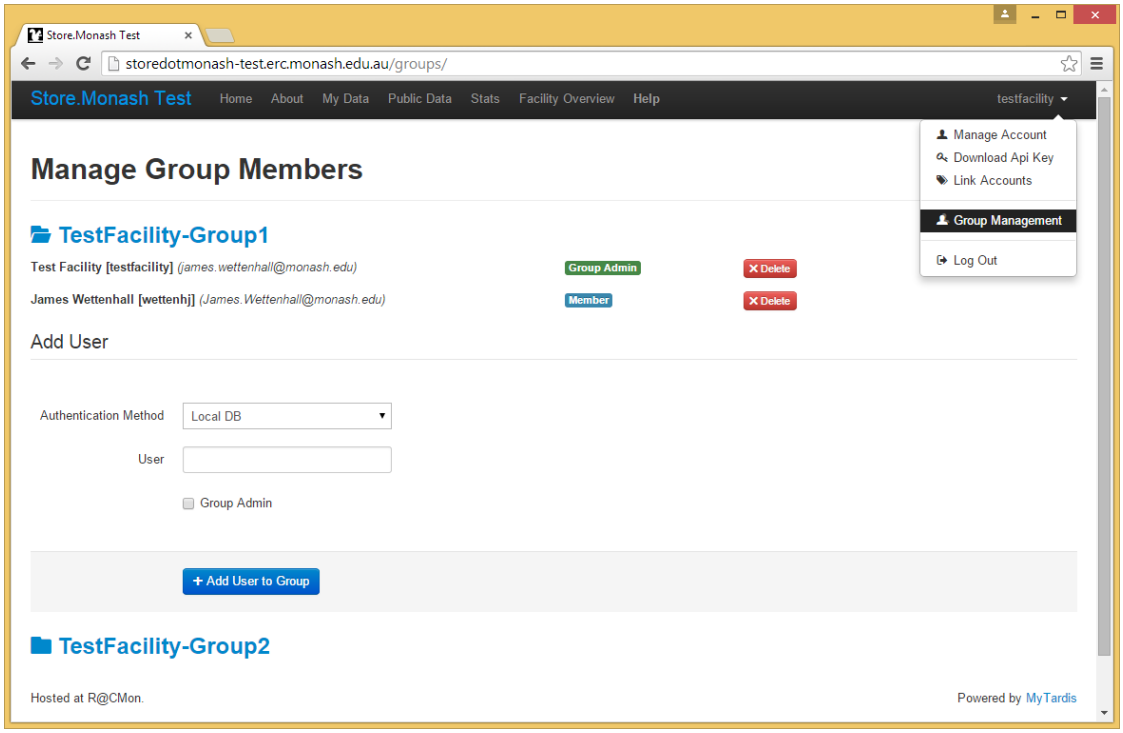


The "testfacility" account in this MyTardis instance is associated with a facility record in MyTardis's database, which means that MyTardis's Facility Overview will be accessible when logged into MyTardis as "testfacility". The Facility Overview lists recently uploaded datasets. The number of verified files in each dataset is the number of files which have been uploaded and confirmed to have the correct file size and MD5 checksum.

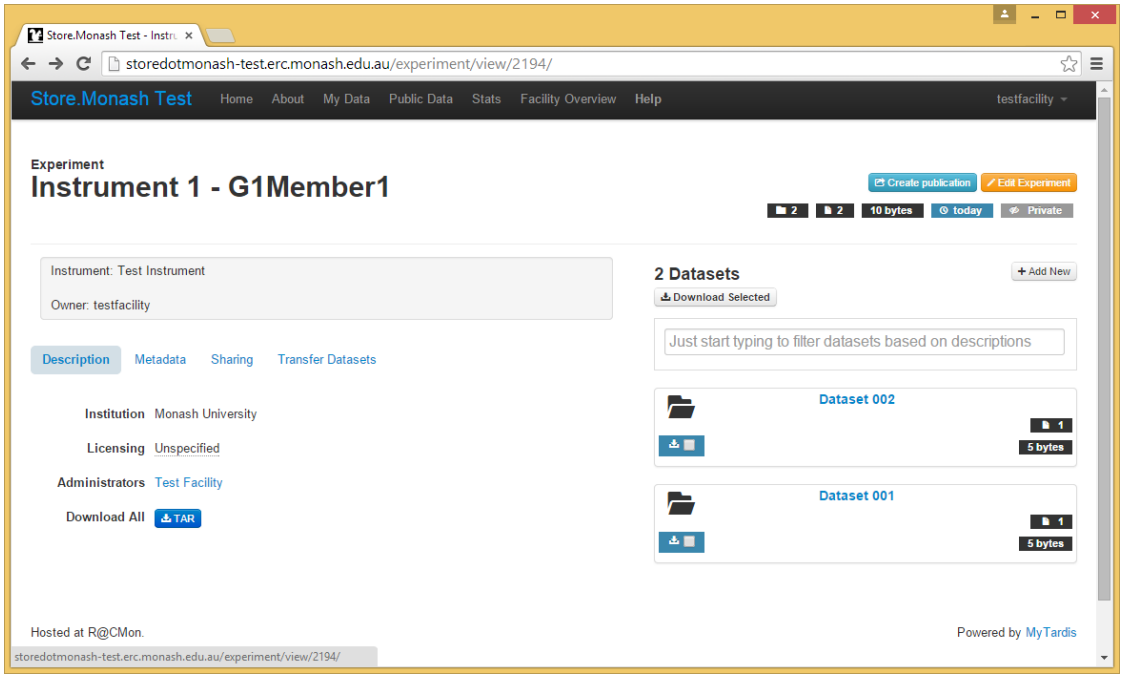


1.8.3 User and Group Management in MyTardis for Facility Managers

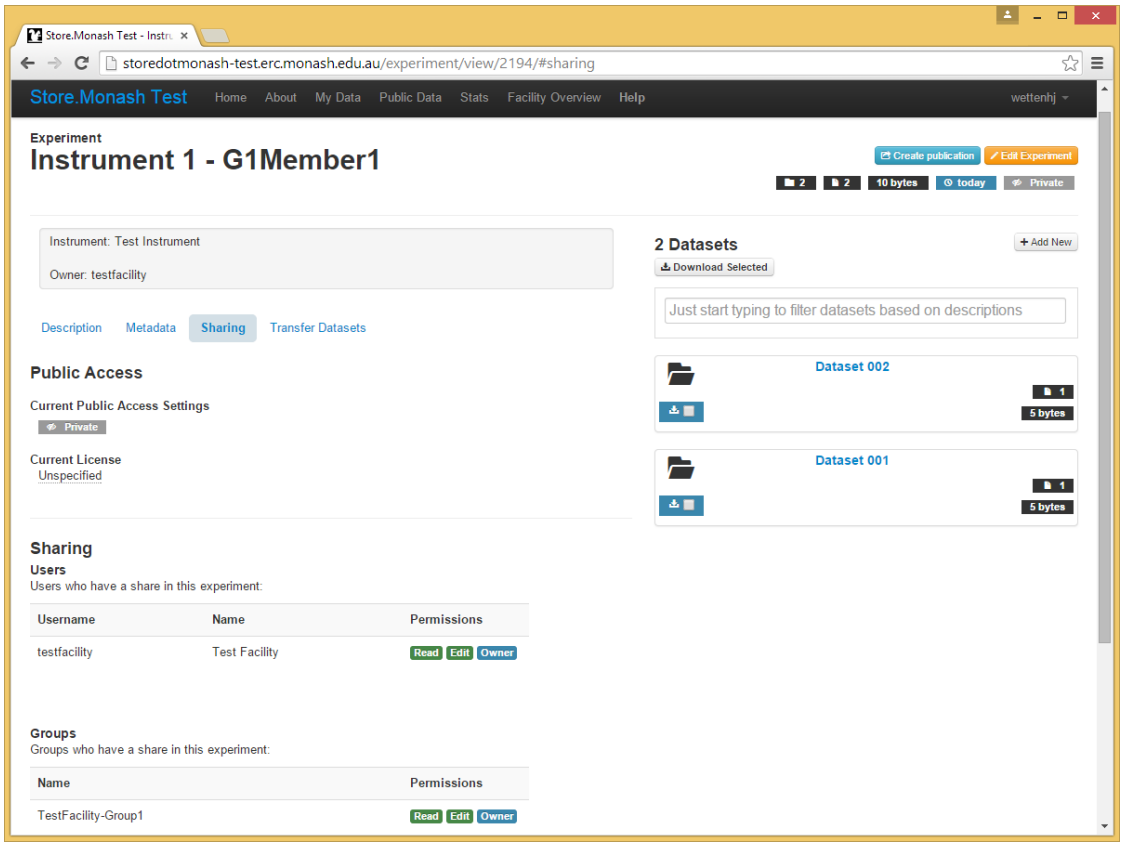
The “testfacility” account in this MyTardis instance is a group administrator for the “TestFacility-Group1” group, which means that they can view all members of that group by selecting Group Management from the drop-down menu available by clicking on the “testfacility” username in the upper-right corner of MyTardis.



Clicking on an experiment from the Facility Overview page (or from the My Data page or from the Home page) allows you to determine which users its datasets are accessible to. In this case, the "Instrument 1 - G1Member1" experiment is owned by "testfacility" and is accessible by the "TestFacility-Group1" group. Users can be granted access (or revoked access) using the Change User Sharing and Change Group Sharing buttons.



Researchers can log into MyTardis and view all experiments which their user group has access to. User "wettenhj" has access to the experiment "Instrument 1 - G1Member1" (below), because he is a member of the "TestFacility-Group1" group.



1.9 Mac OS X Walkthrough

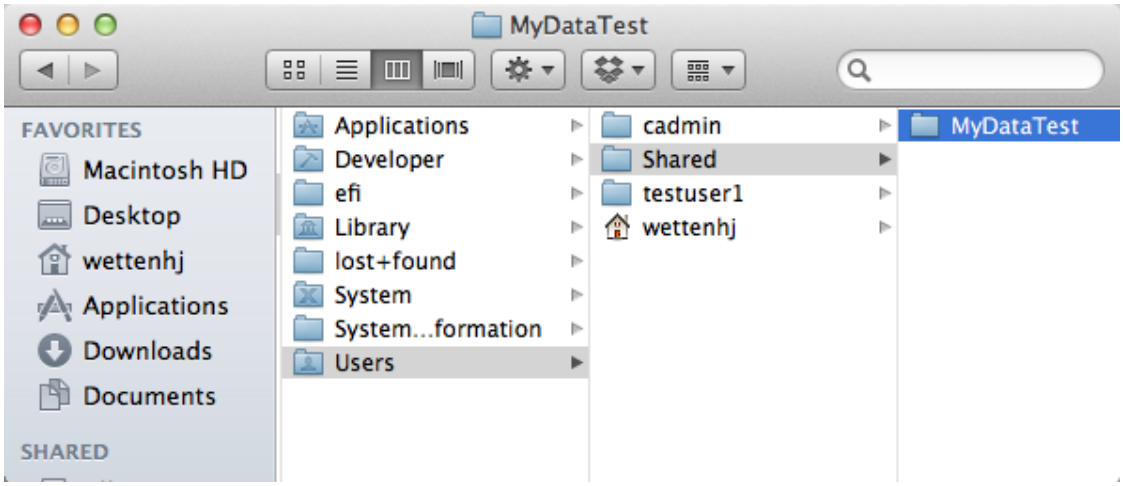
MyData development is primarily targeting Windows, which is the OS of choice for most data-collection instrument PCs. This document aims to demonstrate that MyData can also run on Mac OS X.

On Mac OS X, after downloading and opening the disk image (DMG) file, drag the MyData application into your Applications folder and launch it. (You can then eject the disk image.)

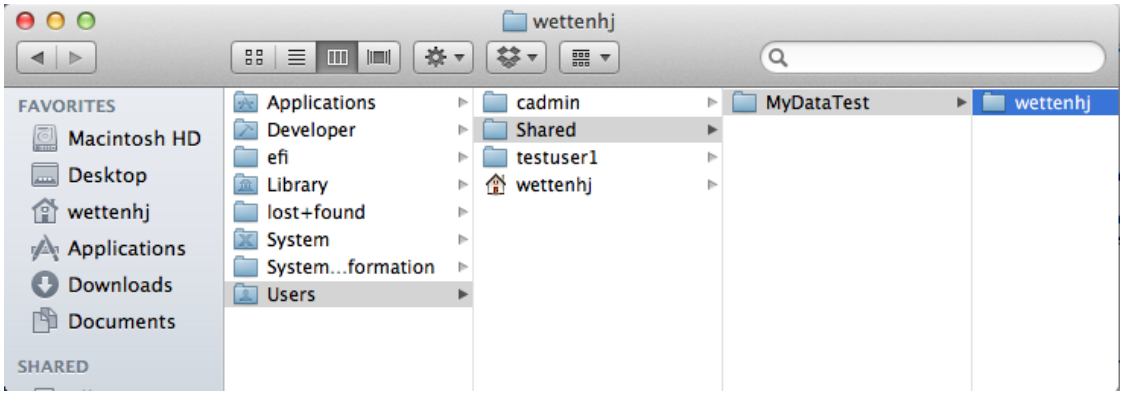
A test MyTardis site is available for authorized MyData testers. Contact Store.Star Help at store.star.help@monash.edu.au if you would like to register for testing MyData against this MyTardis test site or if you would like assistance with setting up an alternative test site for MyData. After registering as an authorized

MyData tester, you will receive a MyTardis username and password.

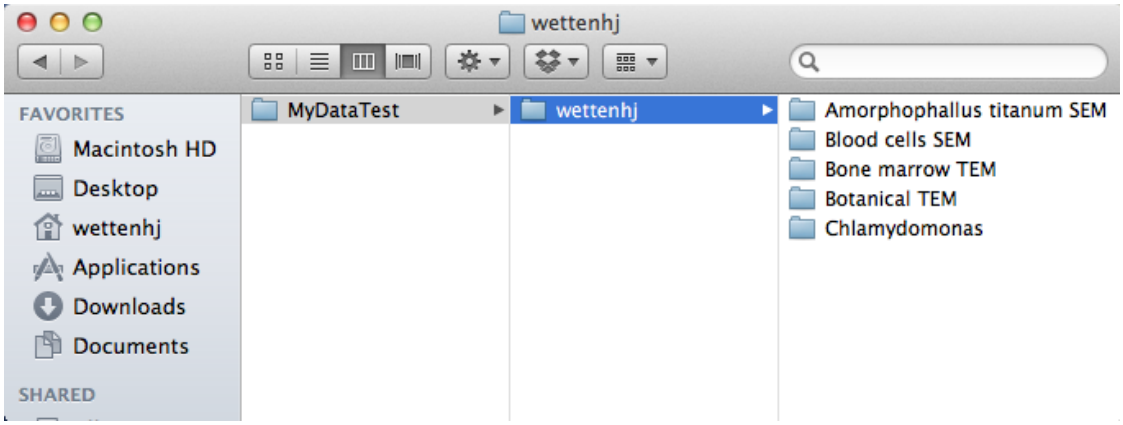
Choose a folder where you would like to store your data. I chose "/Users/Shared/MyDataTest":



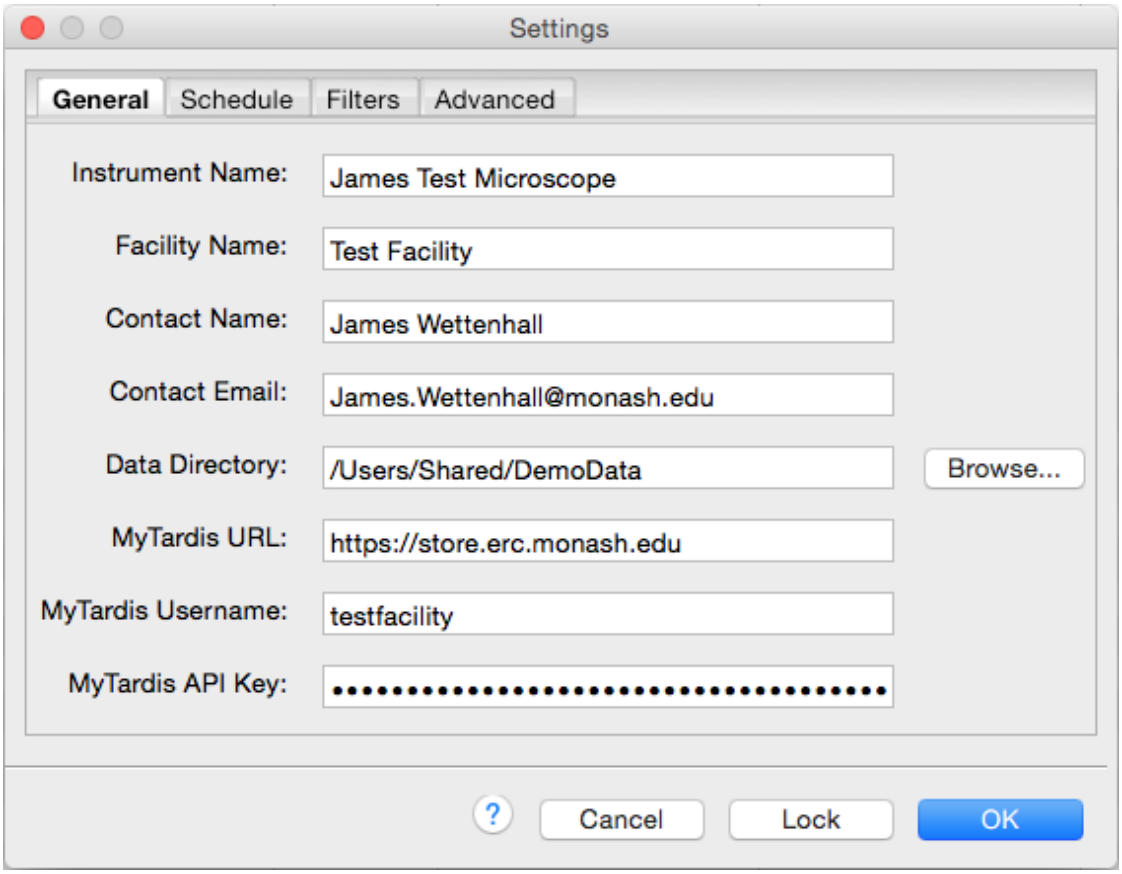
Create a folder whose name matches your MyTardis username (e.g. "wettenhj"):



Put your data within your user folder, ensuring that all datafiles are grouped within folders, which will become datasets in MyTardis:



Launch MyData, and enter some basic settings in MyData's Settings dialog (below). Each field within the Settings dialog is described here: http://mydata.readthedocs.org/en/latest/settings.html



1.9.1 Starting MyData's Scan and Upload Processes

Depending on the "schedule type" you configure in MyData's Settings dialog, MyData can scan for data and attempt to upload it as soon as you click "OK" on the Settings dialog or whenever you press the refresh icon on MyData's toolbar, or whenever you launch MyData.

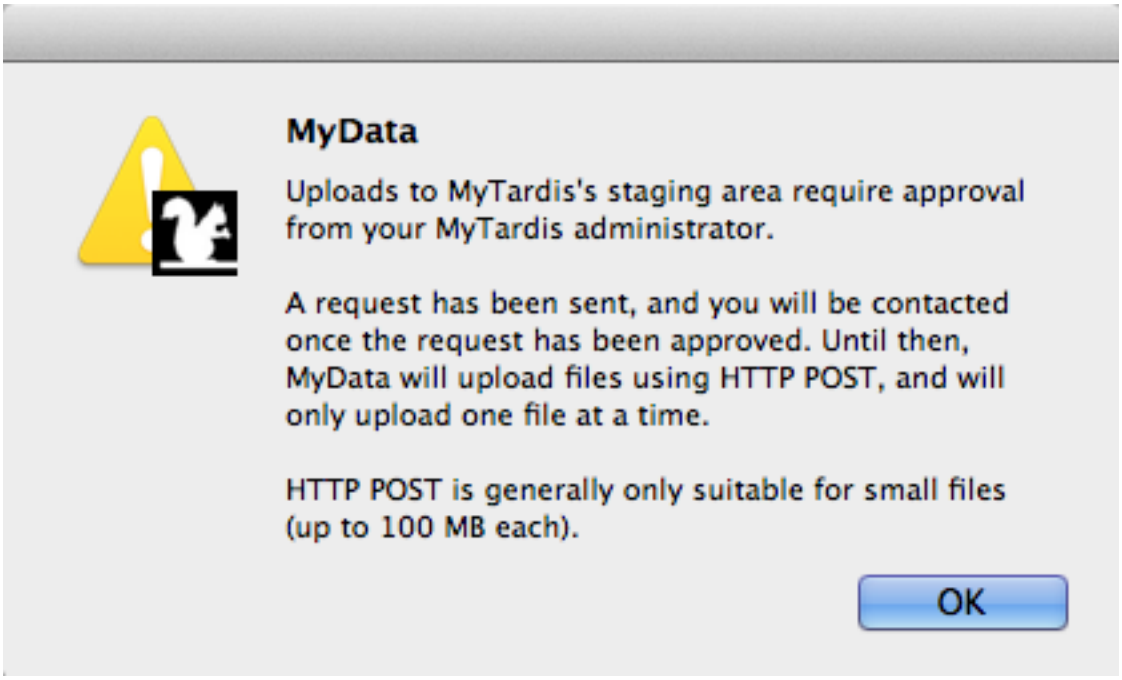
1.9.2 MyData's Upload Methods

MyData supports two upload methods - HTTP POST and SCP via staging.

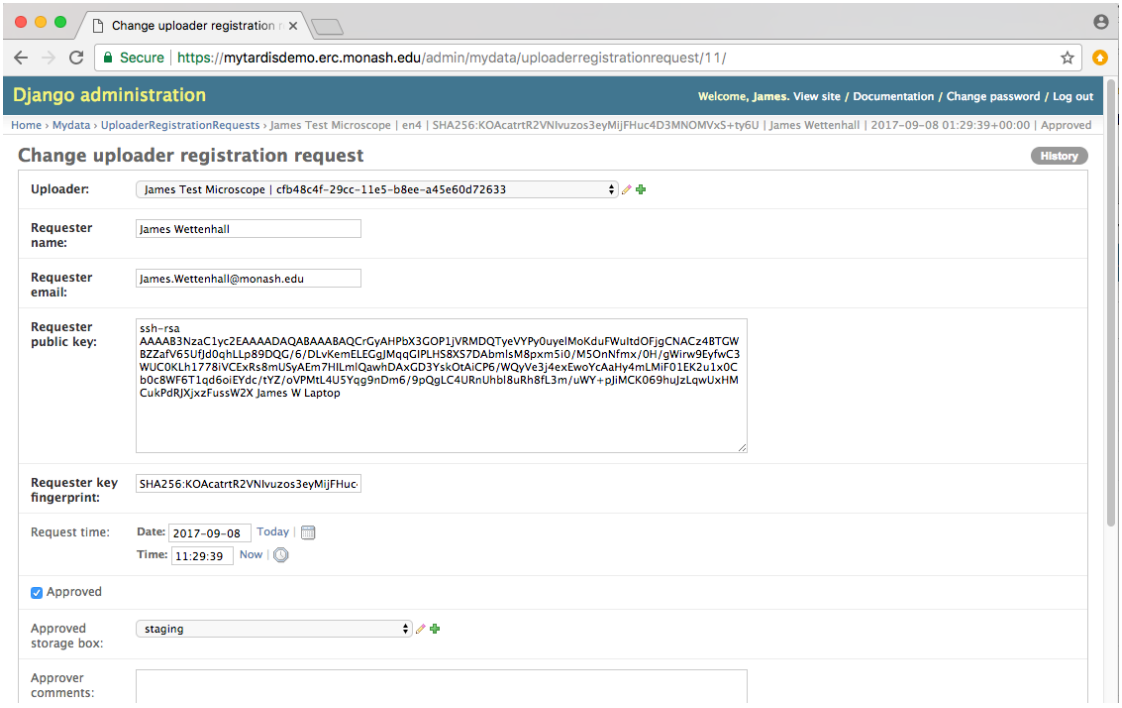
The HTTP POST upload method is only intended to be used for quick demos and for small data files. Uploading large files with HTTP POST can put significant strain on the MyTardis server's resources (particularly memory usage). The only advantage of HTTP POST is that it is easy to configure. As long as you have access to a suitable MyTardis role account (e.g. "testfacility") and know its API key, then you can begin uploading from MyData straight away, although you should begin by testing small files only.

SCP via staging is MyData's preferred upload method. MyData will automatically use this method as soon as it become available, but uploads via staging need to be approved by a MyTardis administrator. MyData generates an SSH key pair the first time it runs and sends the public key to the MyTardis server in a request for the ability to upload via staging. The MyTardis administrator needs to approve the request and put the public key in a suitable authorized keys file on the staging server (which could be the same as the MyTardis server). For example, the public key could be put in "/home/mydata/.ssh/authorized_keys" on the staging server.

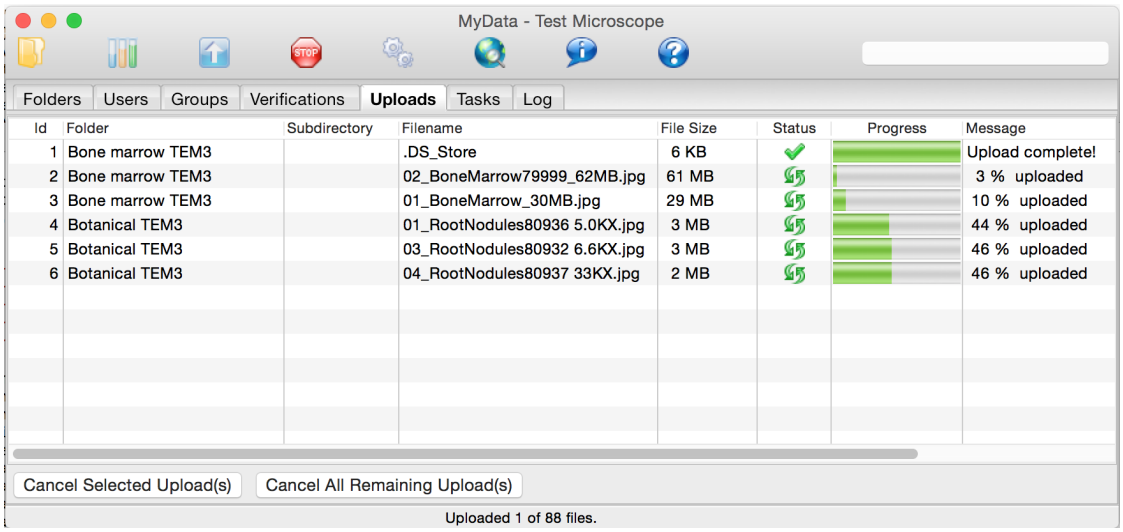
The first time a user runs MyData, they wil see a warning indicating that MyData's preferred upload method (SCP via staging) hasn't yet been approved by the MyTardis administartor:



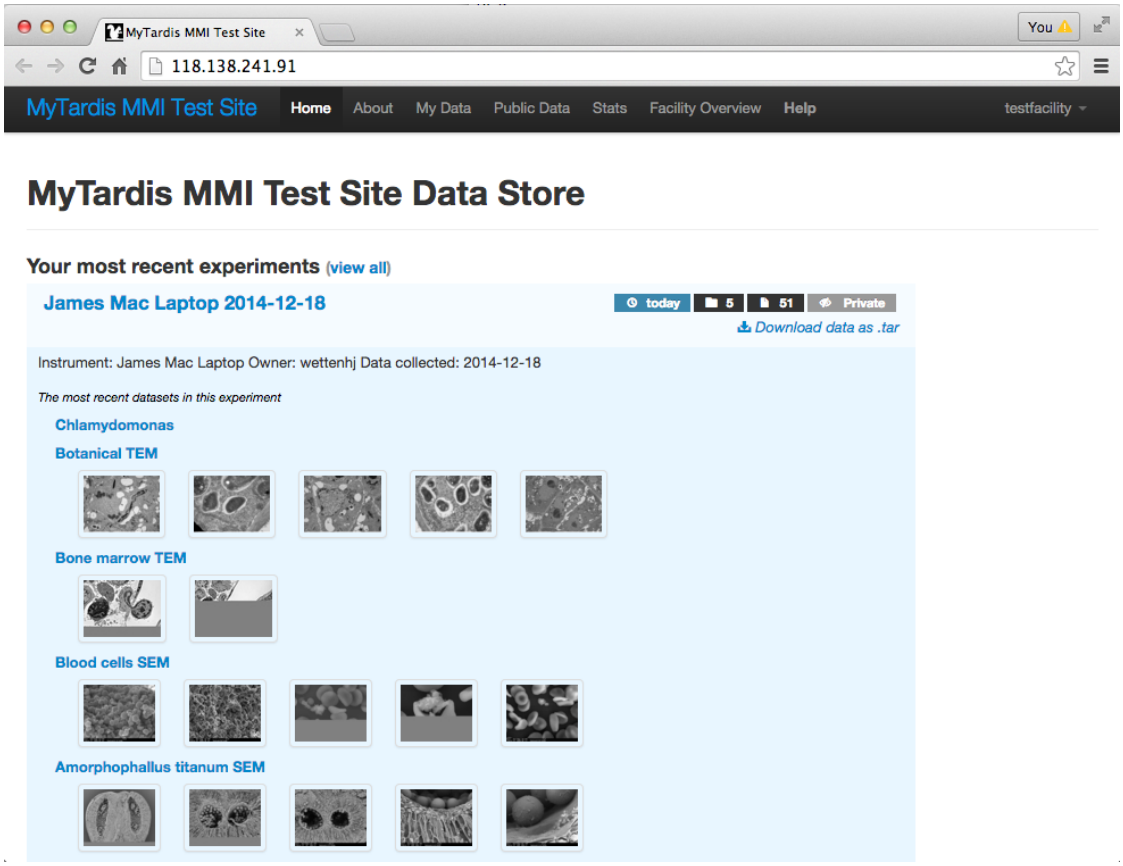
The MyTardis administrator can approve the request in the Django Admin interface (after adding the public key to the appropriate /home/mydata/.ssh/authorized_keys file):



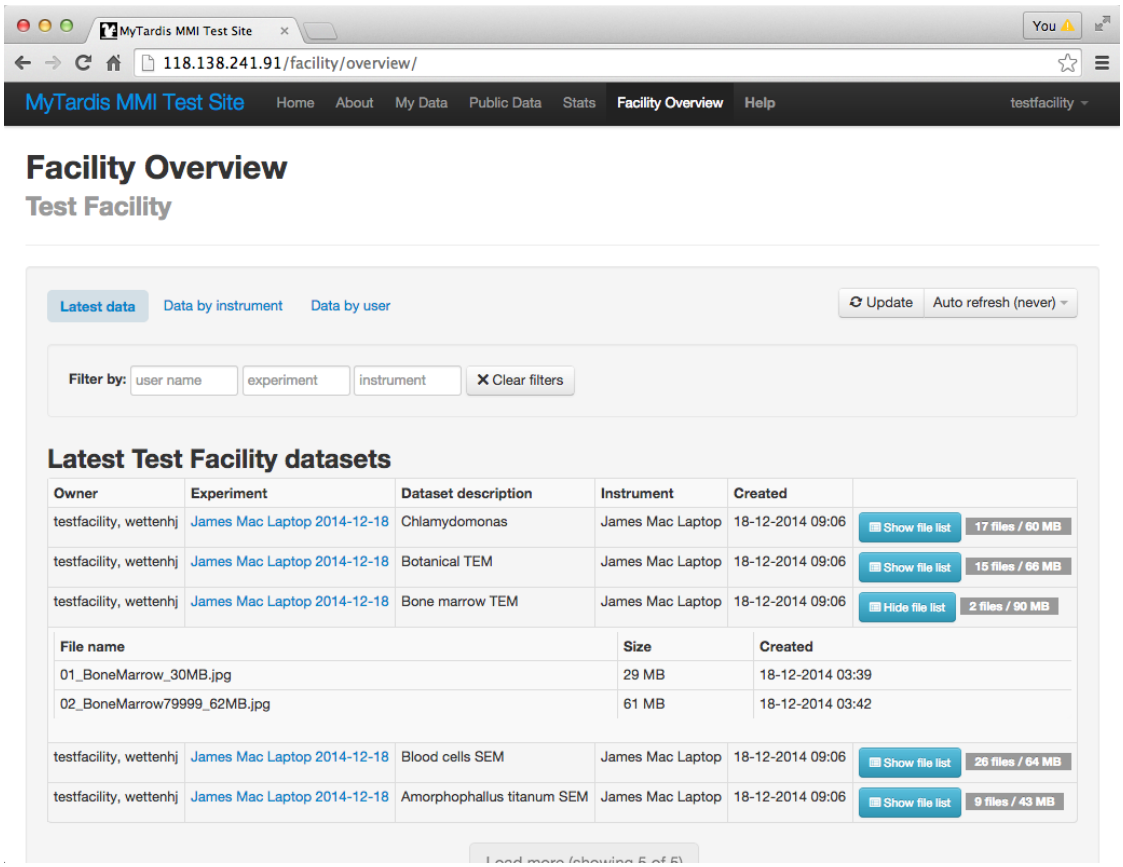
Once uploads to staging have been approved, MyData can manage multiple uploads at once (5 by default):



By clicking on the web browser icon on MyData's toolbar, you can view the uploaded data in MyTardis in your web browser. The data will be jointly owned by the facility role account (e.g. "testfacility") and by the MyTardis user whose username (e.g. "wettenhj") was used to name the folder containing the datasets. MyTardis allows grouping datasets together into experiments. MyData uses the instrument name (e.g. "James Mac Laptop") and the date of creation of the dataset folders (e.g. "2014-12-18") to define a default experiment for the datasets it uploads:



If you are authorized to log into MyTardis's web interface as a facility manager, you can view the data in MyTardis's new Facility Overview. Note the two owners - the facility role account ("testfacility") and the individual user ("wettenhj") who collected the data:



1.10 MyData Tutorial

1.10.1 Recent Changes

Previous versions of this tutorial used a MyTardis demo server which provided an easy way to install a local MyTardis test server running an appropriate version of MyTardis for MyData. Now that MyData is compatible with MyTardis's official "develop" branch, it is no longer necessary to provide easy ways to install the unofficial fork of MyTardis which was previously used with MyData.

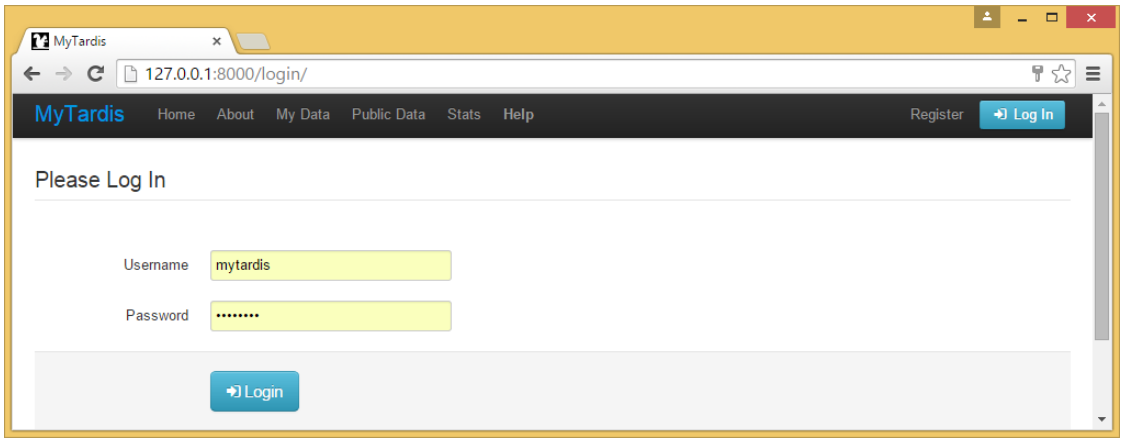
It is still possible to run through this tutorial with a local MyTardis test server. Testing MyData's staging uploads is best done with a real remote MyTardis server. But the majority of MyData's functionality (as described below) can be tested against a local MyTardis test server.

Installing MyTardis and running a local test server is beyond the scope of this tutorial. For more information, see https://github.com/mytardis/mytardis/blob/develop/build.sh and ask for help if needed.

For anyone wishing to work through this tutorial interactively, it will be assumed that you know how to set up a MyTardis test server, accessible at http://127.0.0.1:8000/

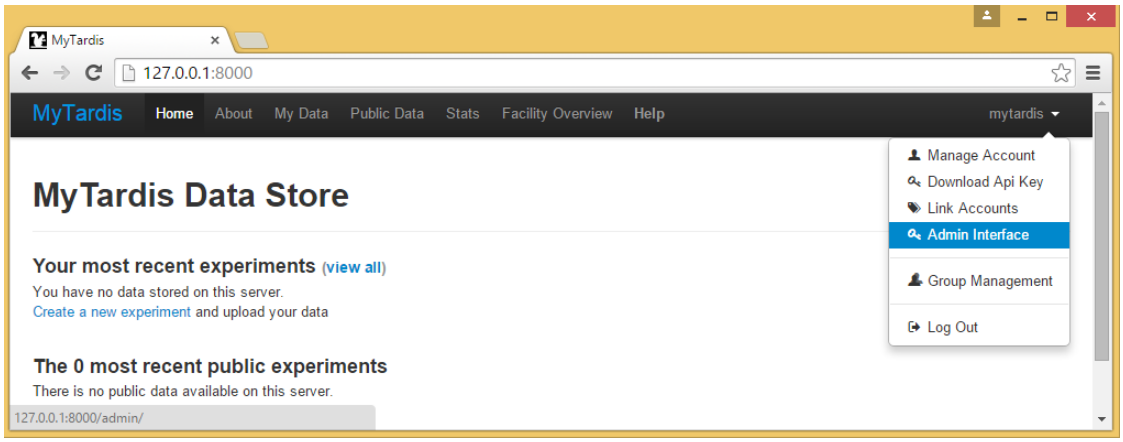
1.10.2 Logging into the MyTardis Test Server as a MyTardis administrator

Click the "Log In" button in the upper right corner, and log in with username "mytardis" and password "mytardis".

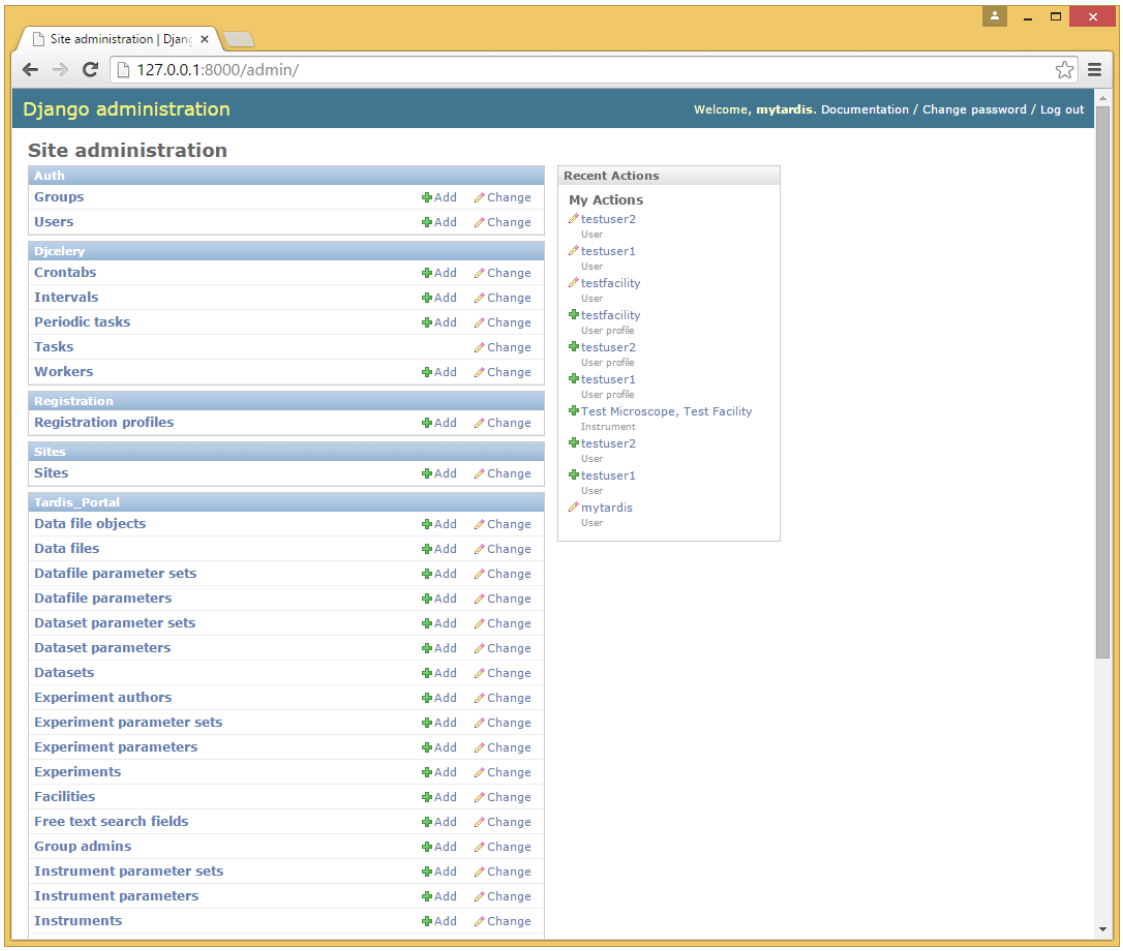


Accessing MyTardis's Django Admin Interface

The "mytardis" account in this test server is a super administrator, i.e. it can do anything, including accessing MyTardis's Django Admin interface from the menu item shown below.

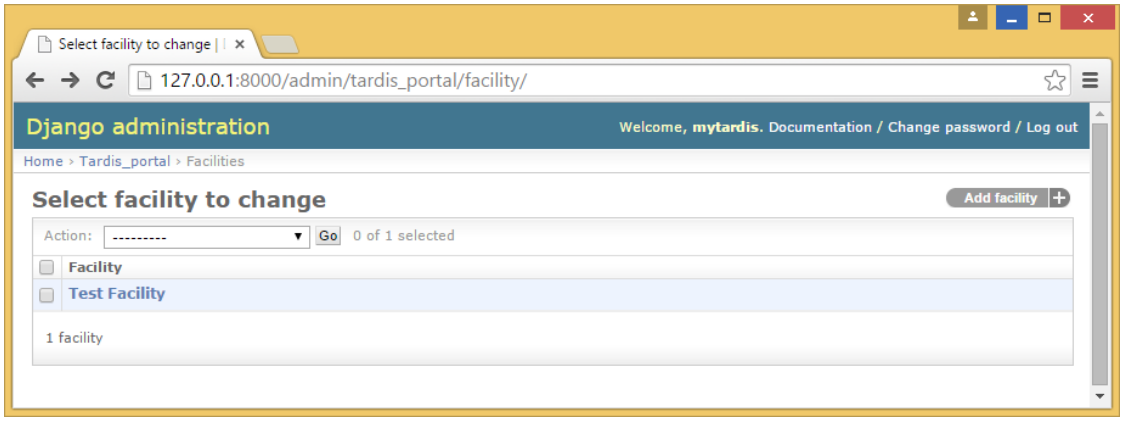


MyTardis's Django Admin interface looks similar to many other Django applications' admin interfaces. Keep in mind that this interface is extremely powerful, so if you are not careful, you could delete database records without any way to recover them!

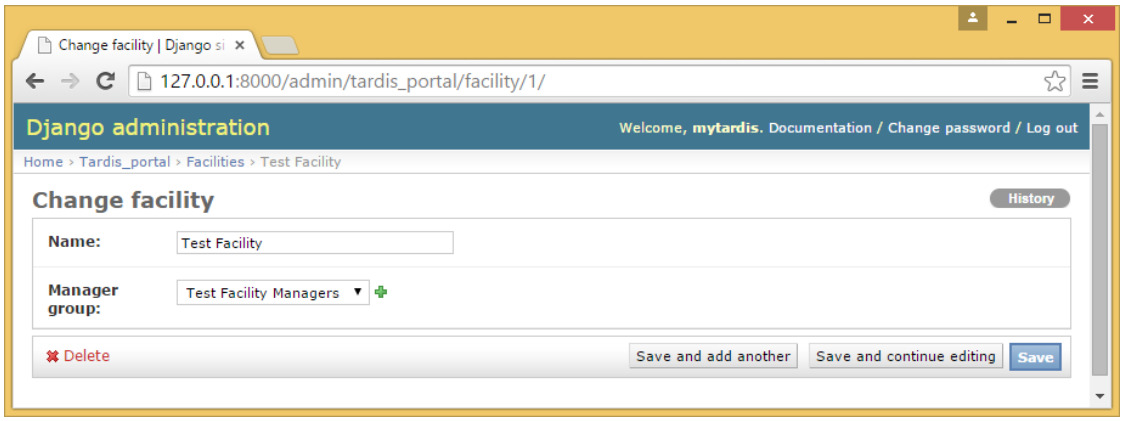


Facilities Registered in MyTardis

From the Django Admin interface, click on "Facilities" to see what facilities are available in this Test Server. There is only one facility, named "Test Facility".

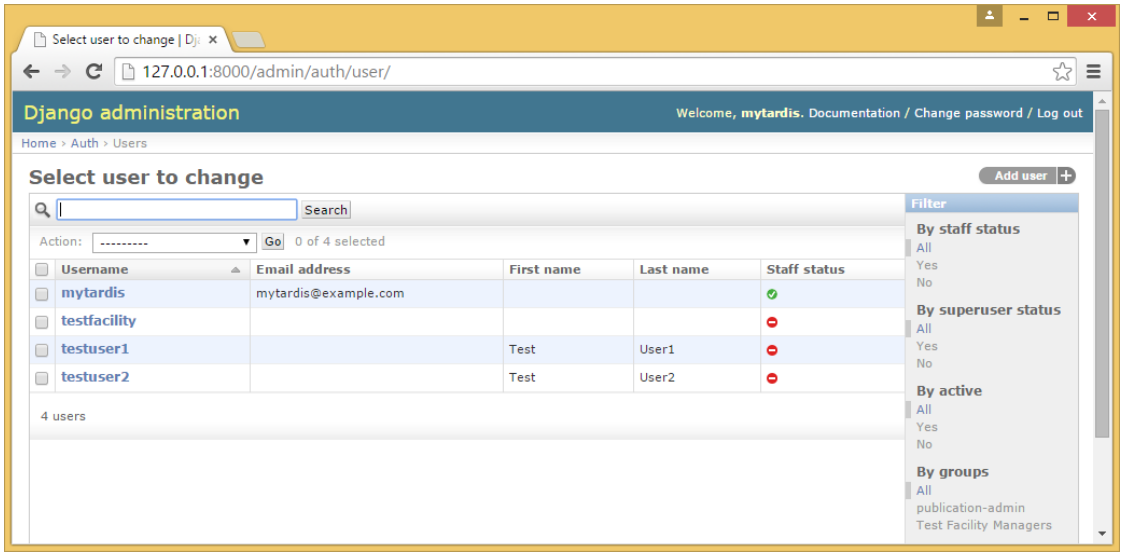


Click on the "Test Facility" facility record to see the properties of the facility, including the "Test Facility Managers" user group assigned to the "Manager group" field of the facility record.

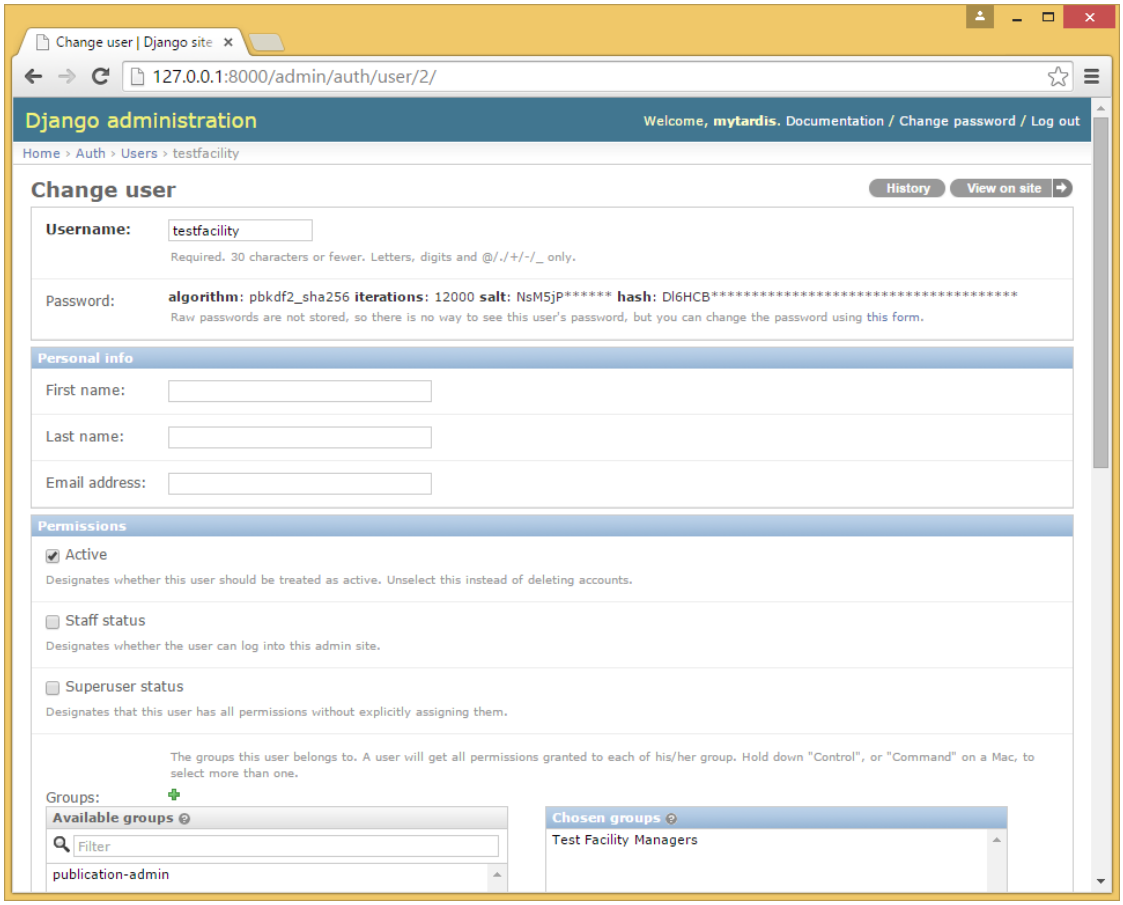


User Accounts in MyTardis

From the Django Admin interface, click on Users to see the user accounts available in this MyTardis server. The "mytardis" administrator is the only account which can access the Django Admin interface.

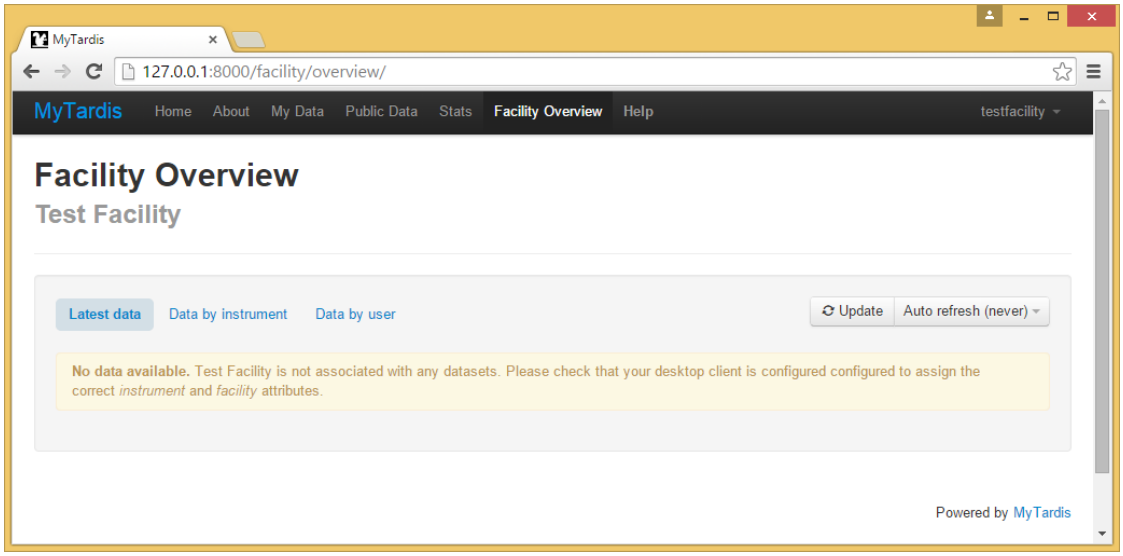


Click on the "testfacility" user account to see its attributes. Note that this account is a member of the "Test Facility" facility record's manager group, named "Test Facility Managers".



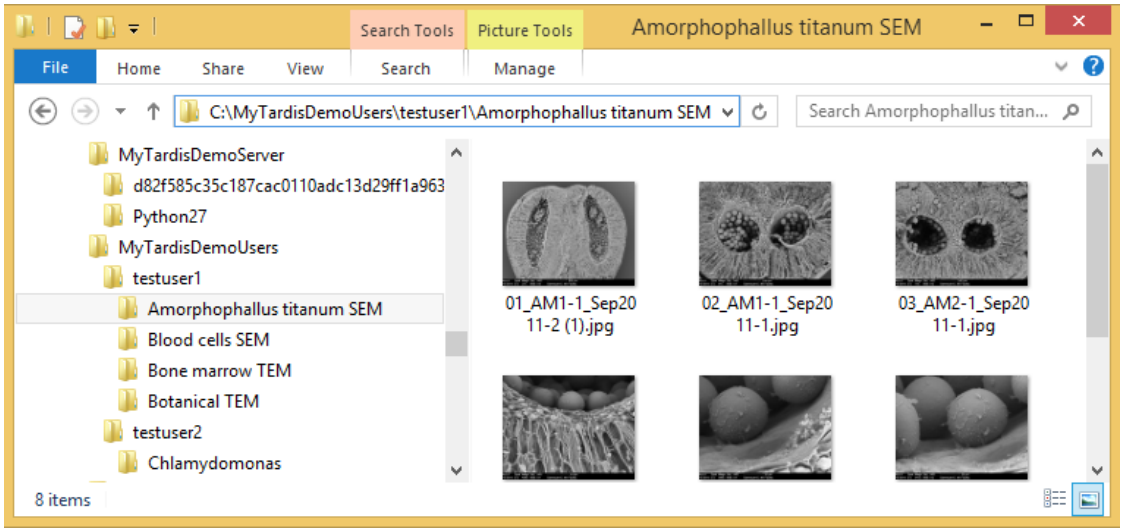
1.10.3 Logging into the MyTardis Test Server as a MyTardis facility manager

Log out of the Django Admin interface, and then return to the original URL in your web browser's address bar, i.e. http://127.0.0.1:8000/, not http://127.0.0.1:8000/admin/. Then log in with username "testfacility" and password "testfacility", and click on the "Facility Overview" section link in the navigation bar at the top of the MyTardis home page. Since we haven't uploaded any data yet, no data will appear in the Facility Overview, but we can confirm that the "testfacility" account has access to the Facility Overview for the "Test Facility" facility.



1.10.4 Obtaining the demo data

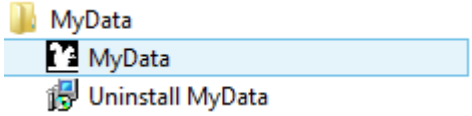
Download MyTardisDemoData.zip and extract it in "C:" to create the "C:\MyTardisDemoUsers" folder shown below:



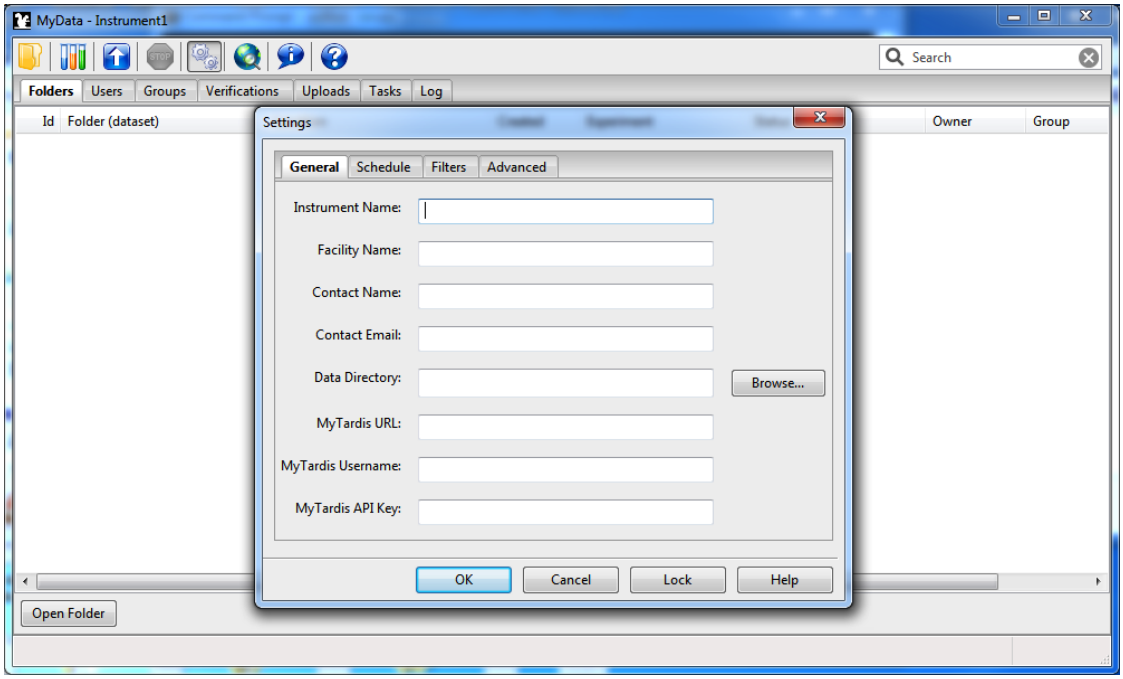
1.10.5 Launching MyData

MyData can be downloaded from here: http://mydata.readthedocs.org/en/latest/download.html

Open the downloaded executable and proceed through the setup wizard to install MyData. A shortcut to MyData will then be available in the Start Menu (or the Start Screen if not using a Start Menu):

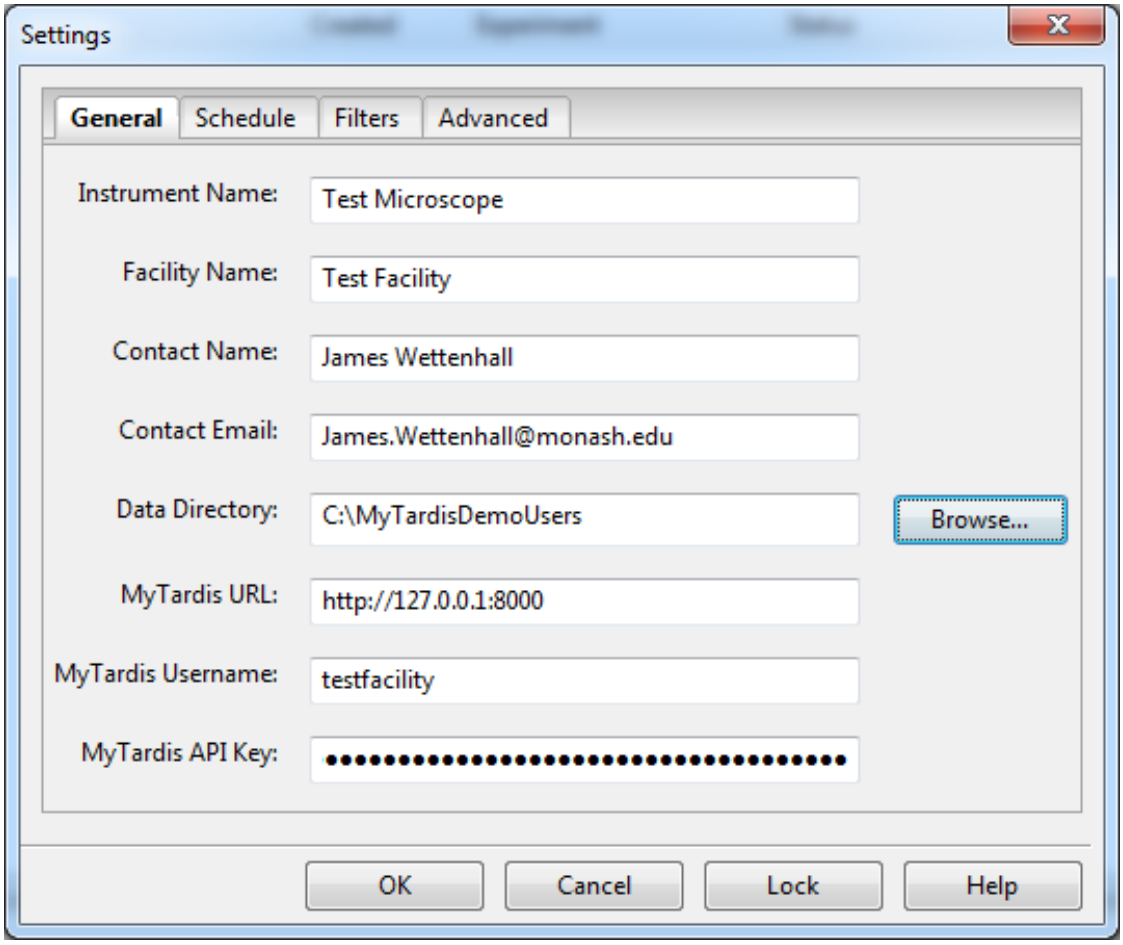


The first time you launch MyData, its settings dialog will appear automatically and appear blank:

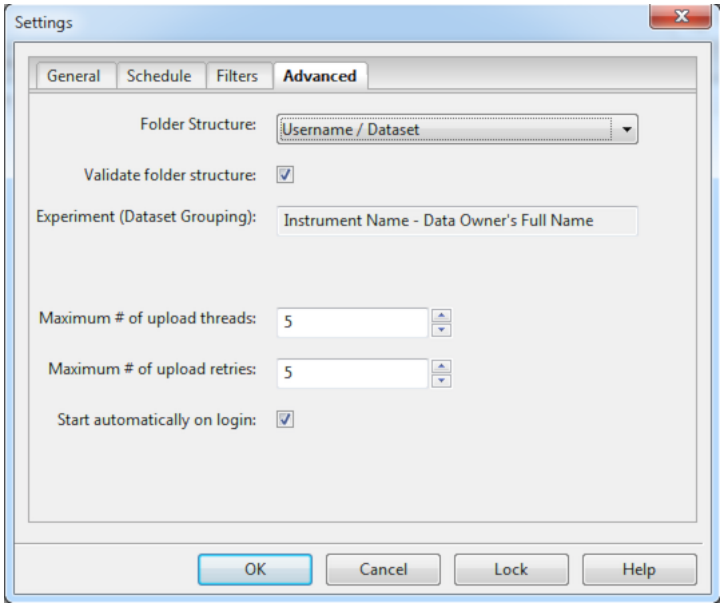


1.10.6 Downloading and installing the demo configuration for MyData

Download MyDataDemo.cfg onto your Desktop and drag and drop it onto MyData's settings dialog, which should automatically populate the fields in MyData's settings dialog.

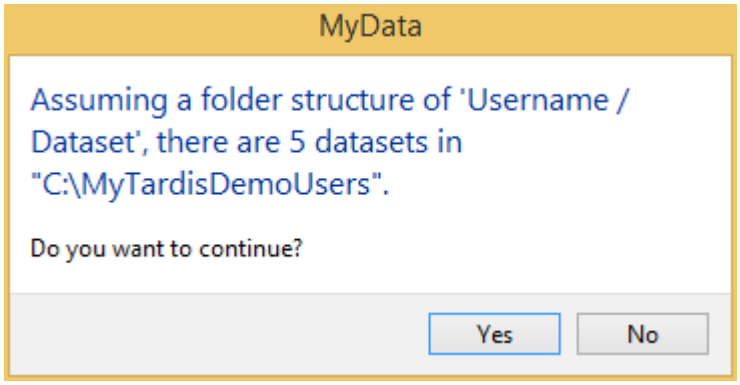


The Advanced tab of MyData's settings dialog contains additional settings:



1.10.7 MyData's Settings Validation

After clicking "OK" on the settings dialog, MyData will validate the settings and inform the user of any problems it finds. When running in interactive mode, MyData will then inform the user of how many datasets it has counted within the data directory and ask the user to confirm that they want to continue.

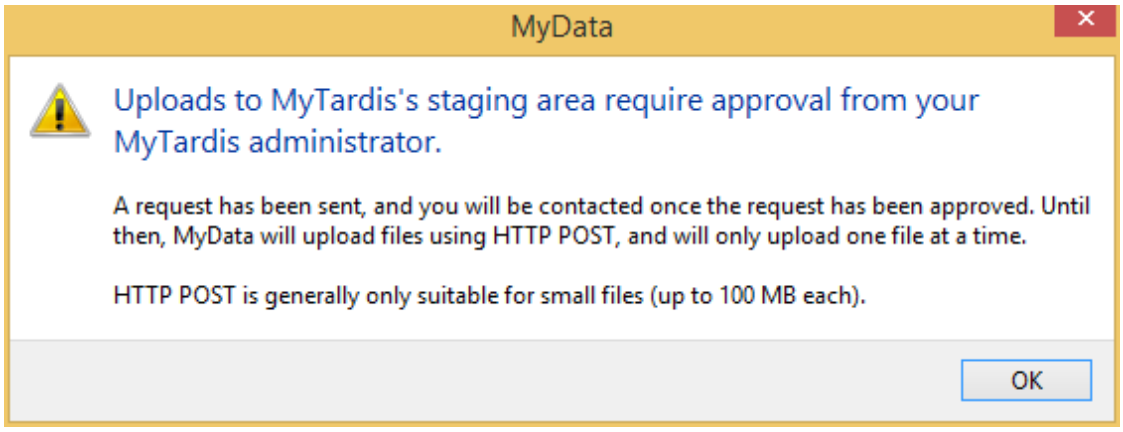


1.10.8 MyData's Upload Methods

MyData offers two upload methods:

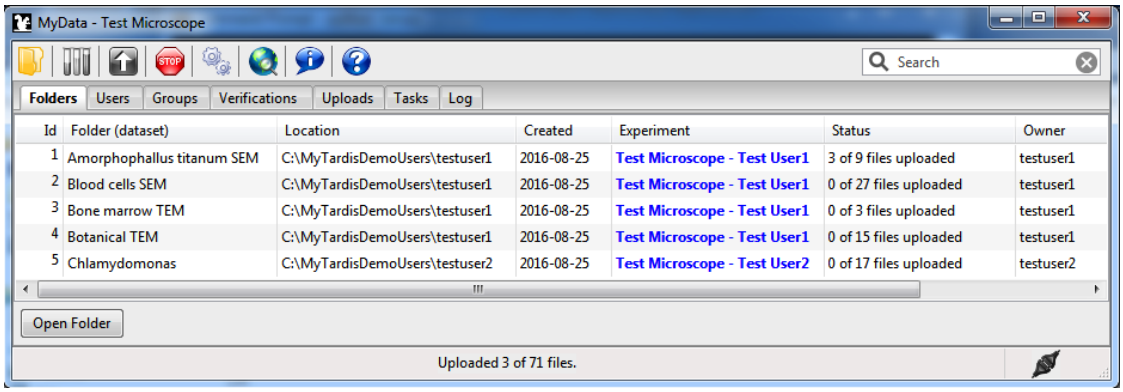
- HTTP POST
- SCP to Staging

The second method ("SCP to Staging") can handle much larger datafiles and supports multiple concurent upload threads, however it is slightly more complicated to set up, so we won't be covering it in this tutorial. Instead, we will stick with MyData's default upload method ("HTTP POST") and ignore the warning dialog below.



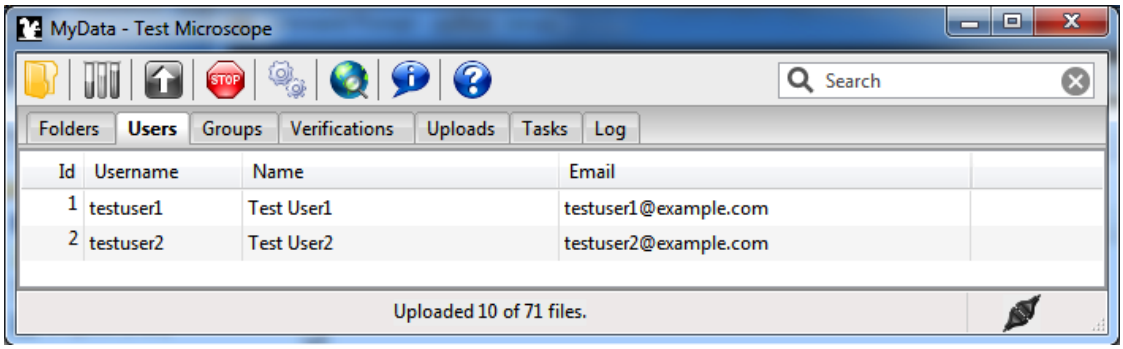
1.10.9 MyData's Folders View

MyData's Folders view lists all of the dataset folders which will be scanned for files to upload to MyTardis. For each folder, MyData displays a count of the total number of files in that folder, and the number of files which have already been uploaded to MyTardis. MyData is stateless, i.e. it won't remember how many files were confirmed to be on MyTardis last time it was run, so each count will begin at zero and then increment by one as each file is confirmed to be available on MyTardis.



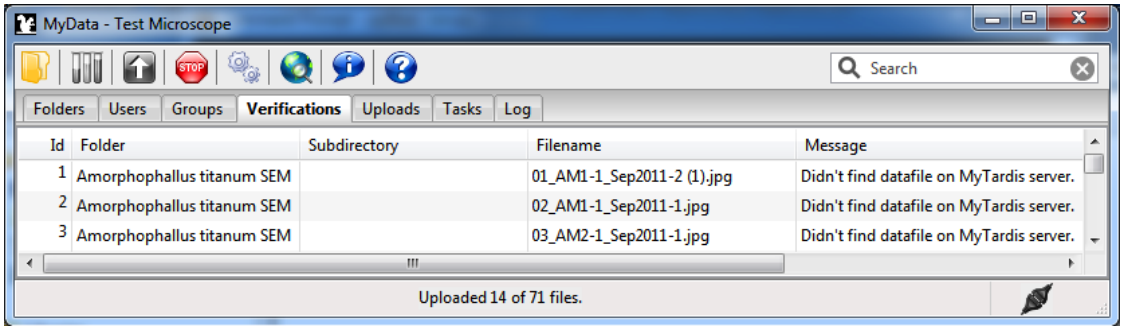
1.10.10 MyData's Users View

MyData's Users view (below) displays the result of MyData's attempt to map the user folder names ("testuser1" and "testuser2") to MyTardis user accounts. In this case, both user folder names have been successfully mapped to user accounts on our MyTardis Test Server, but no email address has been recorded for either account in MyTardis. Many queries MyData performs against MyTardis will only work if the MyTardis account you entered in MyData's settings dialog ("testfacility") has sufficient permissions assigned to it, as shown on the Django Admin's user account attributes page for the "testfacility" account. In this case, the "testfacility" account can access other users' email addresses because it is a member of a Facility Managers group in MyTardis.



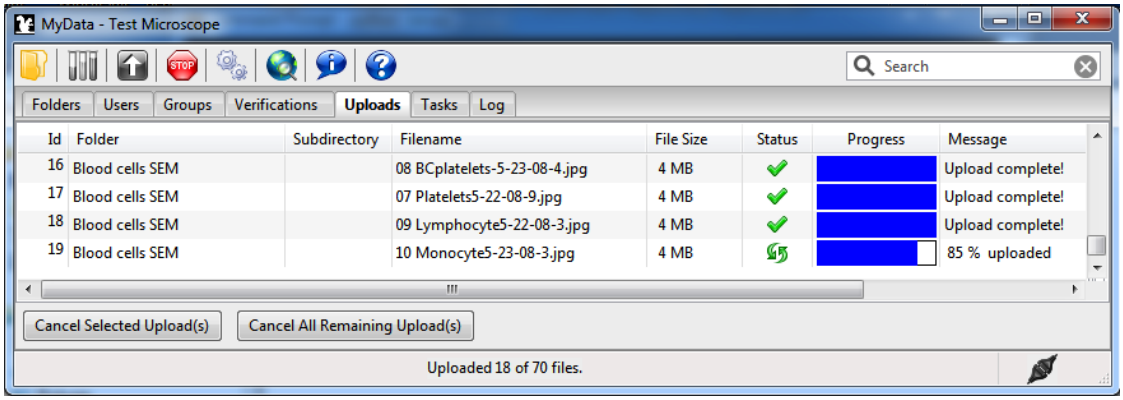
1.10.11 MyData's Verifications View

MyData's Verifications view (below) shows MyData's attempts to verify whether each datafile is available on the MyTardis server, or whether it needs to be uploaded.



1.10.12 MyData's Uploads View

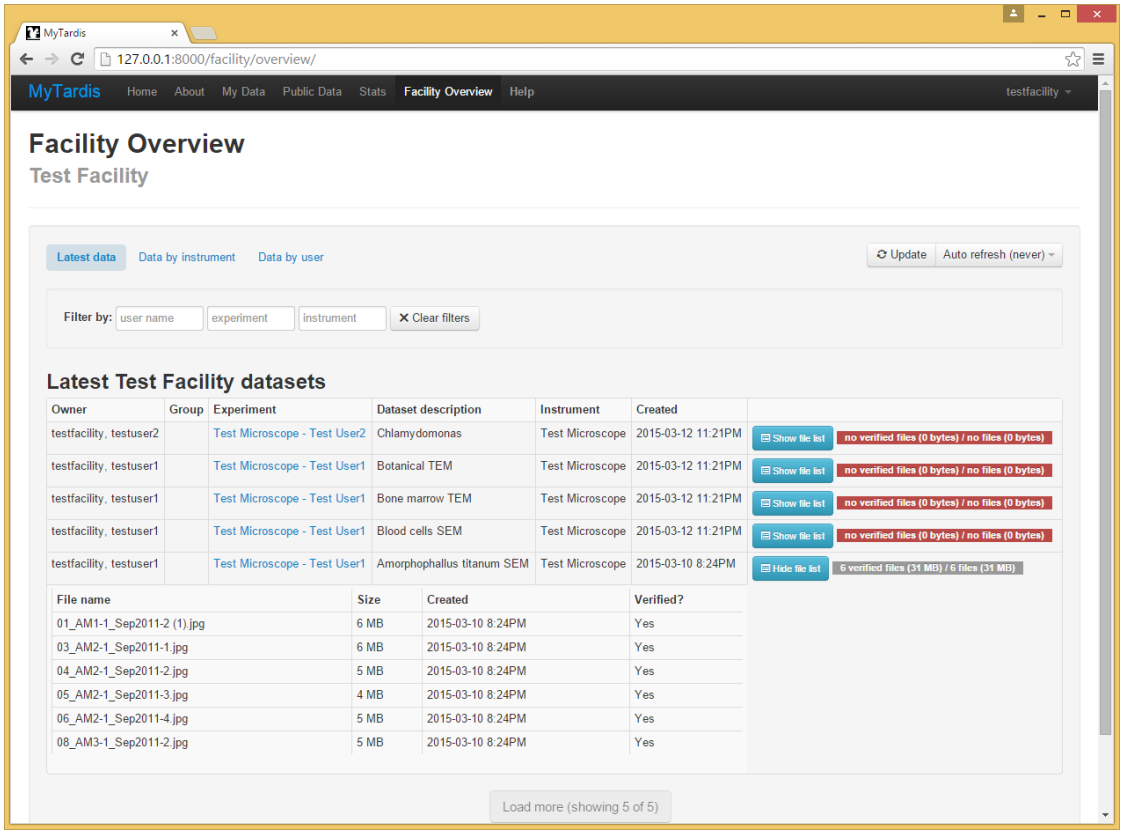
MyData's Uploads view (below) shows MyData's upload progress. The default HTTP POST method only supports one concurrent upload, whereas the "SCP to Staging" upload method supports multiple concurrent uploads.



1.10.13 Monitoring MyData Uploads in MyTardis's Facility Overview

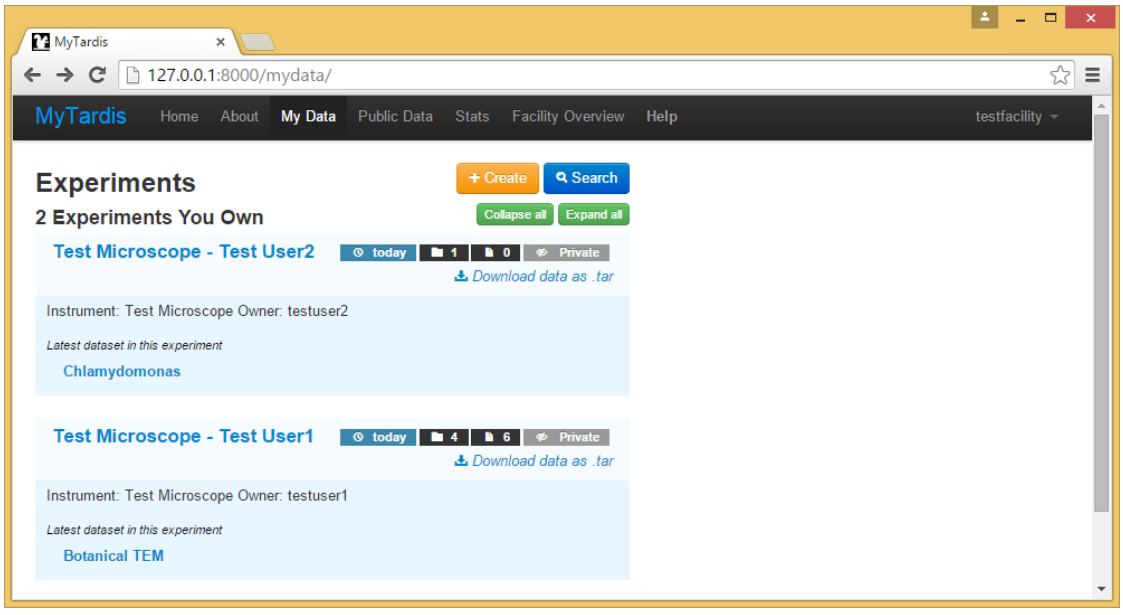
After some of the datafiles have completed uploading, you can check back in your web browser to see the datafiles in MyTardis's Facility Overview (below). You should be logged into MyTardis as the "testfacility" account (username "testfacility", password "testfacility").

For the test server, we are using the CELERY_ALWAYS_EAGER setting which means that datafiles will be verified immediately, instead of as a background task. This explains why the number of verified datafiles below is always equal to the total number of datafiles for each dataset. In the screenshot below, only 6 datafiles have been uploaded from the "Amorphophallus Titanum SEM" dataset, and no datafiles have been uploaded from the other datasets yet.



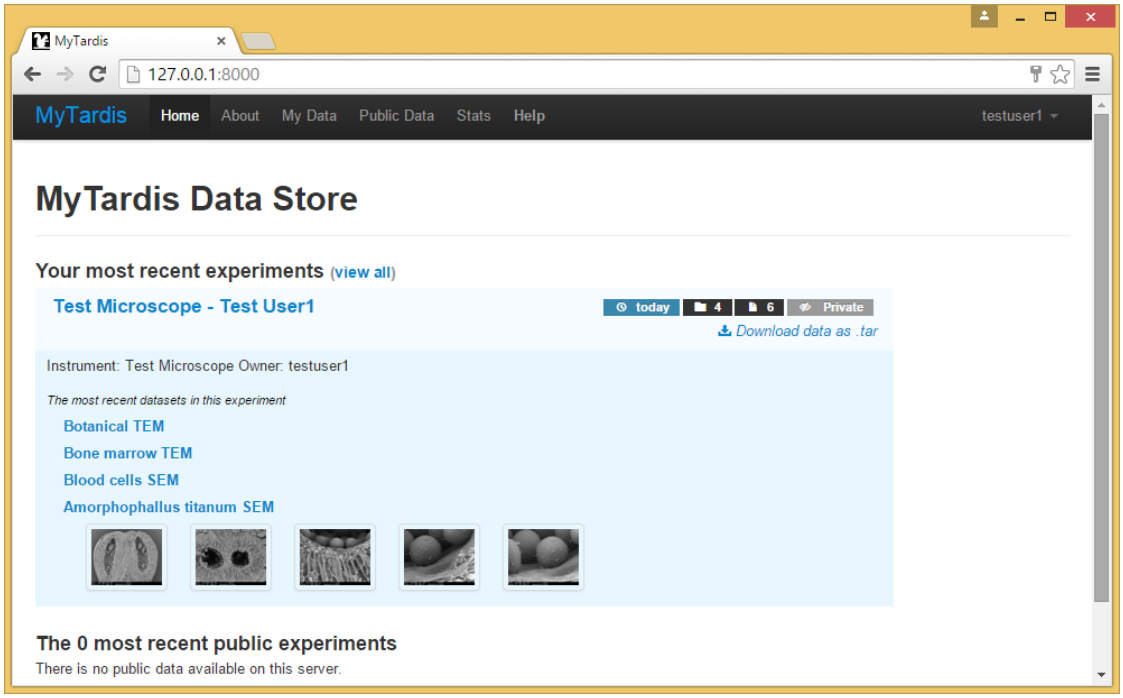
1.10.14 MyTardis's "My Data" View from a Facility Manager's Perspective

While logged in as "testfacility" (an account whose credentials could be shared amongst the managers of "Test Facility"), click on "My Data" to see all of the "experiments" (dataset collections) created by MyData while running at that facility. MyData's default dataset grouping uses the instrument name ("Test Microscope") and the user's full name (e.g. "Test User1") to define a MyTardis "experiment" record, as seen in MyTardis's "My Data" view below.

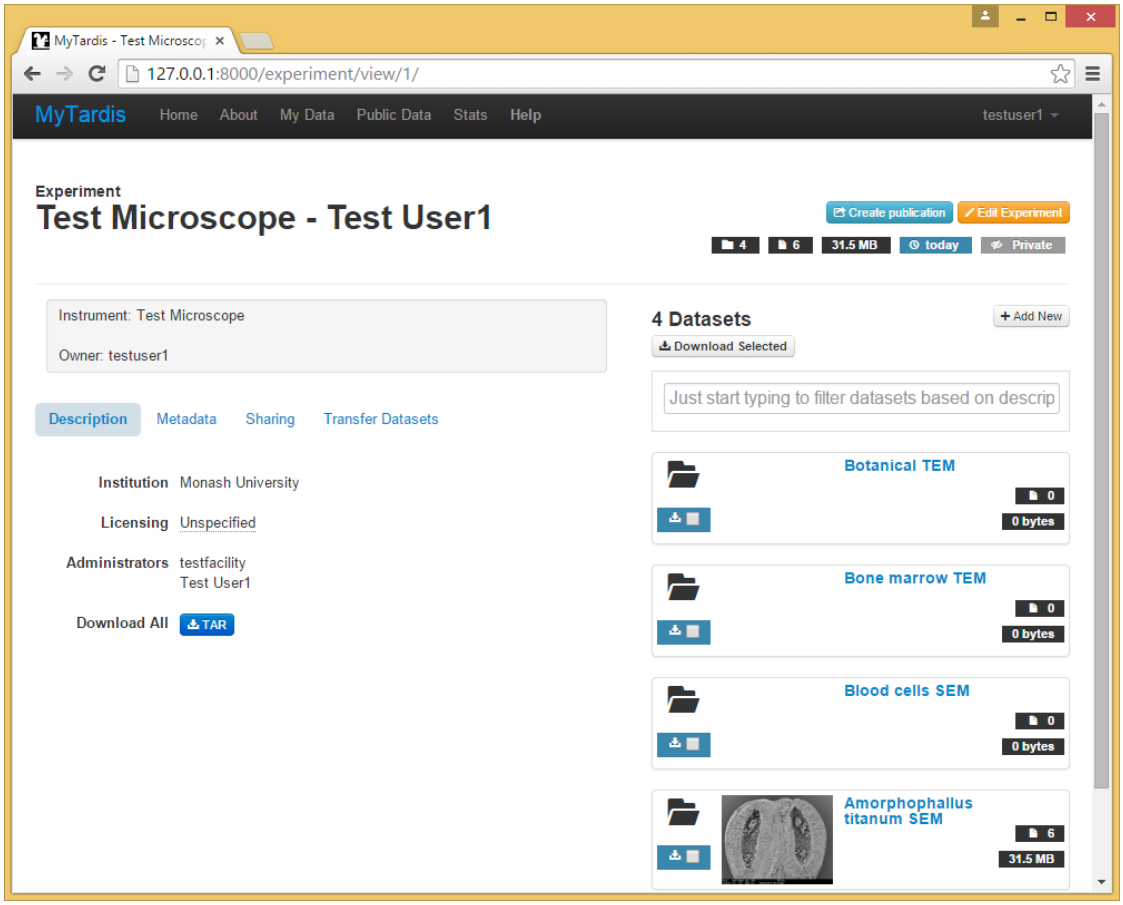


1.10.15 MyTardis from a Facility User's Perspective

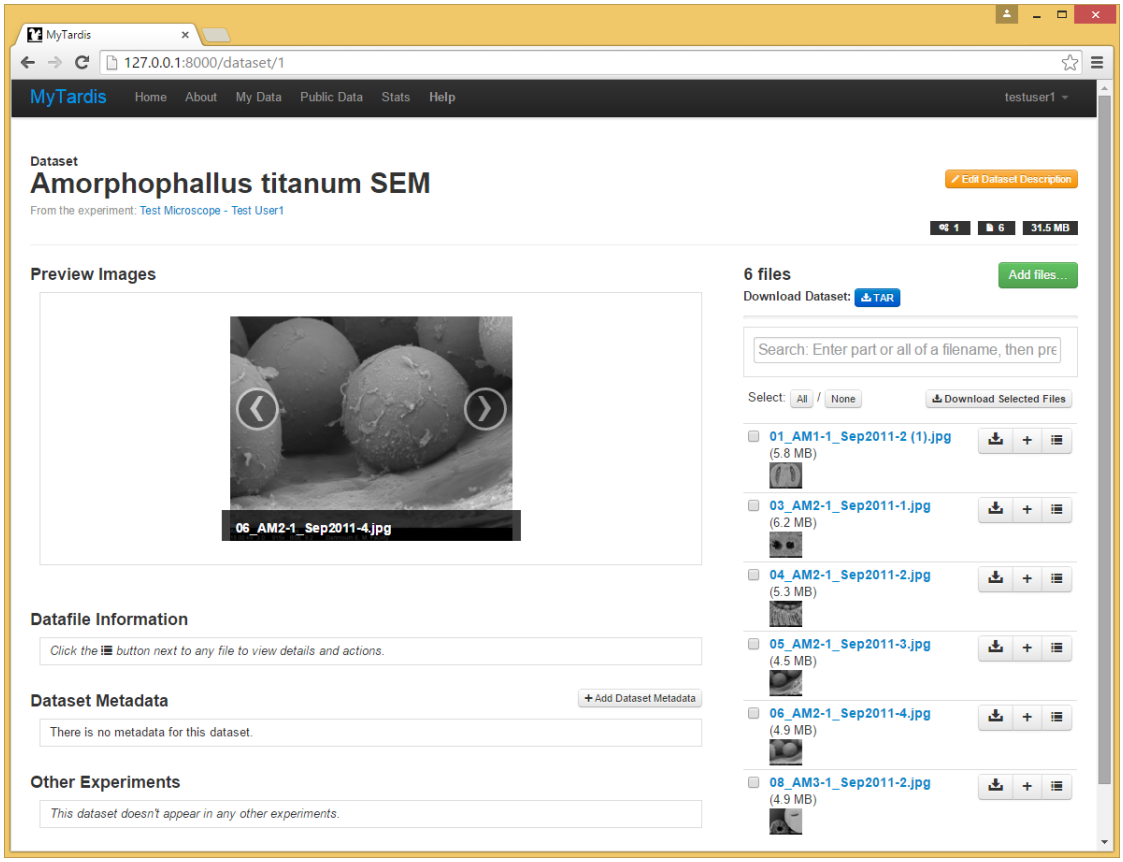
Log out of MyTardis, and log back in with the username "testuser1" and password "testuser1". Now you only see the data collected by user "testuser1", not the data collected by "testuser2". The "Test User1" in the experiment (dataset group) names may seem redundant here, but users can share their experiments with other users, so it would be confusing if all of the shared experiments were just given a default name of "Test Microscope".



Click on the "Test Microscope - Test User1" experiment to see the datasets included in that experiment:



Click on the "Amorphophallus Titanum SEM Dataset" to see the datafiles in that dataset:



1.11 License

MyData's source code is licensed under GPL v3 (see below).

MyData distributions include some commercial icons which are only intended to be used within MyData, i.e. they not covered by the GPL v3 license. See: https://github.com/mytardis/mydata/tree/master/mydata/media/Aha-Soft

```
GNU GENERAL PUBLIC LICENSE
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copyleft license for
software and other kinds of works.

The licenses for most software and other practical works are designed
to take away your freedom to share and change the works.  By contrast,
the GNU General Public License is intended to guarantee your freedom to
share and change all versions of a program--to make sure it remains free
software for all its users.  We, the Free Software Foundation, use the
GNU General Public License for most of our software; it applies also to
any other work released this way by its authors.  You can apply it to
```

(continues on next page)

(continued from previous page)

```
your programs, too.

When we speak of free software, we are referring to freedom, not
price.  Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
them if you wish), that you receive source code or can get it if you
want it, that you can change the software or use pieces of it in new
free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you
these rights or asking you to surrender the rights.  Therefore, you have
certain responsibilities if you distribute copies of the software, or if
you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether
gratis or for a fee, you must pass on to the recipients the same
freedoms that you received.  You must make sure that they, too, receive
or can get the source code.  And you must show them these terms so they
know their rights.

Developers that use the GNU GPL protect your rights with two steps:
(1) assert copyright on the software, and (2) offer you this License
giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains
that there is no warranty for this free software.  For both users' and
authors' sake, the GPL requires that modified versions be marked as
changed, so that their problems will not be attributed erroneously to
authors of previous versions.

Some devices are designed to deny users access to install or run
modified versions of the software inside them, although the manufacturer
can do so.  This is fundamentally incompatible with the aim of
protecting users' freedom to change the software.  The systematic
pattern of such abuse occurs in the area of products for individuals to
use, which is precisely where it is most unacceptable.  Therefore, we
have designed this version of the GPL to prohibit the practice for those
products.  If such problems arise substantially in other domains, we
stand ready to extend this provision to those domains in future versions
of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents.
States should not allow patents to restrict development and use of
software on general-purpose computers, but in those that do, we wish to
avoid the special danger that patents applied to a free program could
make it effectively proprietary.  To prevent this, the GPL assures that
patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and
modification follow.

                       TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.
```

(continues on next page)

(continued from previous page)

```
"Copyright" also means copyright-like laws that apply to other kinds of
works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this
License.  Each licensee is addressed as "you".  "Licensees" and
"recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work
in a fashion requiring copyright permission, other than the making of an
exact copy.  The resulting work is called a "modified version" of the
earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based
on the Program.

To "propagate" a work means to do anything with it that, without
permission, would make you directly or secondarily liable for
infringement under applicable copyright law, except executing it on a
computer or modifying a private copy.  Propagation includes copying,
distribution (with or without modification), making available to the
public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other
parties to make or receive copies.  Mere interaction with a user through
a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices"
to the extent that it includes a convenient and prominently visible
feature that (1) displays an appropriate copyright notice, and (2)
tells the user that there is no warranty for the work (except to the
extent that warranties are provided), that licensees may convey the
work under this License, and how to view a copy of this License.  If
the interface presents a list of user commands or options, such as a
menu, a prominent item in the list meets this criterion.

1. Source Code.

The "source code" for a work means the preferred form of the work
for making modifications to it.  "Object code" means any non-source
form of a work.

A "Standard Interface" means an interface that either is an official
standard defined by a recognized standards body, or, in the case of
interfaces specified for a particular programming language, one that
is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other
than the work as a whole, that (a) is included in the normal form of
packaging a Major Component, but which is not part of that Major
Component, and (b) serves only to enable use of the work with that
Major Component, or to implement a Standard Interface for which an
implementation is available to the public in source code form.  A
"Major Component", in this context, means a major essential component
(kernel, window system, and so on) of the specific operating system
(if any) on which the executable work runs, or a compiler used to
produce the work, or an object code interpreter used to run it.
```

(continues on next page)

(continued from previous page)

```
The "Corresponding Source" for a work in object code form means all
the source code needed to generate, install, and (for an executable
work) run the object code and to modify the work, including scripts to
control those activities.  However, it does not include the work's
System Libraries, or general-purpose tools or generally available free
programs which are used unmodified in performing those activities but
which are not part of the work.  For example, Corresponding Source
includes interface definition files associated with source files for
the work, and the source code for shared libraries and dynamically
linked subprograms that the work is specifically designed to require,
such as by intimate data communication or control flow between those
subprograms and other parts of the work.

The Corresponding Source need not include anything that users
can regenerate automatically from other parts of the Corresponding
Source.

The Corresponding Source for a work in source code form is that
same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of
copyright on the Program, and are irrevocable provided the stated
conditions are met.  This License explicitly affirms your unlimited
permission to run the unmodified Program.  The output from running a
covered work is covered by this License only if the output, given its
content, constitutes a covered work.  This License acknowledges your
rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not
convey, without conditions so long as your license otherwise remains
in force.  You may convey covered works to others for the sole purpose
of having them make modifications exclusively for you, or provide you
with facilities for running those works, provided that you comply with
the terms of this License in conveying all material for which you do
not control copyright.  Those thus making or running the covered works
for you must do so exclusively on your behalf, under your direction
and control, on terms that prohibit them from making any copies of
your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under
the conditions stated below.  Sublicensing is not allowed; section 10
makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological
measure under any applicable law fulfilling obligations under article
11 of the WIPO copyright treaty adopted on 20 December 1996, or
similar laws prohibiting or restricting circumvention of such
measures.

When you convey a covered work, you waive any legal power to forbid
circumvention of technological measures to the extent such circumvention
is effected by exercising rights under this License with respect to
the covered work, and you disclaim any intention to limit operation or
```

(continues on next page)

(continued from previous page)

```
modification of the work as a means of enforcing, against the work's
users, your or third parties' legal rights to forbid circumvention of
technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you
receive it, in any medium, provided that you conspicuously and
appropriately publish on each copy an appropriate copyright notice;
keep intact all notices stating that this License and any
non-permissive terms added in accord with section 7 apply to the code;
keep intact all notices of the absence of any warranty; and give all
recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey,
and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to
produce it from the Program, in the form of source code under the
terms of section 4, provided that you also meet all of these conditions:

  a) The work must carry prominent notices stating that you modified
  it, and giving a relevant date.

  b) The work must carry prominent notices stating that it is
  released under this License and any conditions added under section
  7.  This requirement modifies the requirement in section 4 to
  "keep intact all notices".

  c) You must license the entire work, as a whole, under this
  License to anyone who comes into possession of a copy.  This
  License will therefore apply, along with any applicable section 7
  additional terms, to the whole of the work, and all its parts,
  regardless of how they are packaged.  This License gives no
  permission to license the work in any other way, but it does not
  invalidate such permission if you have separately received it.

  d) If the work has interactive user interfaces, each must display
  Appropriate Legal Notices; however, if the Program has interactive
  interfaces that do not display Appropriate Legal Notices, your
  work need not make them do so.

A compilation of a covered work with other separate and independent
works, which are not by their nature extensions of the covered work,
and which are not combined with it such as to form a larger program,
in or on a volume of a storage or distribution medium, is called an
"aggregate" if the compilation and its resulting copyright are not
used to limit the access or legal rights of the compilation's users
beyond what the individual works permit.  Inclusion of a covered work
in an aggregate does not cause this License to apply to the other
parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms
```

(continues on next page)

(continued from previous page)

```
of sections 4 and 5, provided that you also convey the
machine-readable Corresponding Source under the terms of this License,
in one of these ways:

  a) Convey the object code in, or embodied in, a physical product
  (including a physical distribution medium), accompanied by the
  Corresponding Source fixed on a durable physical medium
  customarily used for software interchange.

  b) Convey the object code in, or embodied in, a physical product
  (including a physical distribution medium), accompanied by a
  written offer, valid for at least three years and valid for as
  long as you offer spare parts or customer support for that product
  model, to give anyone who possesses the object code either (1) a
  copy of the Corresponding Source for all the software in the
  product that is covered by this License, on a durable physical
  medium customarily used for software interchange, for a price no
  more than your reasonable cost of physically performing this
  conveying of source, or (2) access to copy the
  Corresponding Source from a network server at no charge.

  c) Convey individual copies of the object code with a copy of the
  written offer to provide the Corresponding Source.  This
  alternative is allowed only occasionally and noncommercially, and
  only if you received the object code with such an offer, in accord
  with subsection 6b.

  d) Convey the object code by offering access from a designated
  place (gratis or for a charge), and offer equivalent access to the
  Corresponding Source in the same way through the same place at no
  further charge.  You need not require recipients to copy the
  Corresponding Source along with the object code.  If the place to
  copy the object code is a network server, the Corresponding Source
  may be on a different server (operated by you or a third party)
  that supports equivalent copying facilities, provided you maintain
  clear directions next to the object code saying where to find the
  Corresponding Source.  Regardless of what server hosts the
  Corresponding Source, you remain obligated to ensure that it is
  available for as long as needed to satisfy these requirements.

  e) Convey the object code using peer-to-peer transmission, provided
  you inform other peers where the object code and Corresponding
  Source of the work are being offered to the general public at no
  charge under subsection 6d.

A separable portion of the object code, whose source code is excluded
from the Corresponding Source as a System Library, need not be
included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any
tangible personal property which is normally used for personal, family,
or household purposes, or (2) anything designed or sold for incorporation
into a dwelling.  In determining whether a product is a consumer product,
doubtful cases shall be resolved in favor of coverage.  For a particular
product received by a particular user, "normally used" refers to a
typical or common use of that class of product, regardless of the status
of the particular user or of the way in which the particular user
```

(continues on next page)

(continued from previous page)

```
actually uses, or expects or is expected to use, the product.  A product
is a consumer product regardless of whether the product has substantial
commercial, industrial or non-consumer uses, unless such uses represent
the only significant mode of use of the product.

"Installation Information" for a User Product means any methods,
procedures, authorization keys, or other information required to install
and execute modified versions of a covered work in that User Product from
a modified version of its Corresponding Source.  The information must
suffice to ensure that the continued functioning of the modified object
code is in no case prevented or interfered with solely because
modification has been made.

If you convey an object code work under this section in, or with, or
specifically for use in, a User Product, and the conveying occurs as
part of a transaction in which the right of possession and use of the
User Product is transferred to the recipient in perpetuity or for a
fixed term (regardless of how the transaction is characterized), the
Corresponding Source conveyed under this section must be accompanied
by the Installation Information.  But this requirement does not apply
if neither you nor any third party retains the ability to install
modified object code on the User Product (for example, the work has
been installed in ROM).

The requirement to provide Installation Information does not include a
requirement to continue to provide support service, warranty, or updates
for a work that has been modified or installed by the recipient, or for
the User Product in which it has been modified or installed.  Access to a
network may be denied when the modification itself materially and
adversely affects the operation of the network or violates the rules and
protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided,
in accord with this section must be in a format that is publicly
documented (and with an implementation available to the public in
source code form), and must require no special password or key for
unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this
License by making exceptions from one or more of its conditions.
Additional permissions that are applicable to the entire Program shall
be treated as though they were included in this License, to the extent
that they are valid under applicable law.  If additional permissions
apply only to part of the Program, that part may be used separately
under those permissions, but the entire Program remains governed by
this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option
remove any additional permissions from that copy, or from any part of
it.  (Additional permissions may be written to require their own
removal in certain cases when you modify the work.)  You may place
additional permissions on material, added by you to a covered work,
for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you
```

(continues on next page)

(continued from previous page)

```
add to a covered work, you may (if authorized by the copyright holders of
that material) supplement the terms of this License with terms:

  a) Disclaiming warranty or limiting liability differently from the
  terms of sections 15 and 16 of this License; or

  b) Requiring preservation of specified reasonable legal notices or
  author attributions in that material or in the Appropriate Legal
  Notices displayed by works containing it; or

  c) Prohibiting misrepresentation of the origin of that material, or
  requiring that modified versions of such material be marked in
  reasonable ways as different from the original version; or

  d) Limiting the use for publicity purposes of names of licensors or
  authors of the material; or

  e) Declining to grant rights under trademark law for use of some
  trade names, trademarks, or service marks; or

  f) Requiring indemnification of licensors and authors of that
  material by anyone who conveys the material (or modified versions of
  it) with contractual assumptions of liability to the recipient, for
  any liability that these contractual assumptions directly impose on
  those licensors and authors.

All other non-permissive additional terms are considered "further
restrictions" within the meaning of section 10.  If the Program as you
received it, or any part of it, contains a notice stating that it is
governed by this License along with a term that is a further
restriction, you may remove that term.  If a license document contains
a further restriction but permits relicensing or conveying under this
License, you may add to a covered work material governed by the terms
of that license document, provided that the further restriction does
not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you
must place, in the relevant source files, a statement of the
additional terms that apply to those files, or a notice indicating
where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the
form of a separately written license, or stated as exceptions;
the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly
provided under this License.  Any attempt otherwise to propagate or
modify it is void, and will automatically terminate your rights under
this License (including any patent licenses granted under the third
paragraph of section 11).

However, if you cease all violation of this License, then your
license from a particular copyright holder is reinstated (a)
provisionally, unless and until the copyright holder explicitly and
finally terminates your license, and (b) permanently, if the copyright
```

(continues on next page)

(continued from previous page)

```
holder fails to notify you of the violation by some reasonable means
prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is
reinstated permanently if the copyright holder notifies you of the
violation by some reasonable means, this is the first time you have
received notice of violation of this License (for any work) from that
copyright holder, and you cure the violation prior to 30 days after
your receipt of the notice.

Termination of your rights under this section does not terminate the
licenses of parties who have received copies or rights from you under
this License.  If your rights have been terminated and not permanently
reinstated, you do not qualify to receive new licenses for the same
material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or
run a copy of the Program.  Ancillary propagation of a covered work
occurring solely as a consequence of using peer-to-peer transmission
to receive a copy likewise does not require acceptance.  However,
nothing other than this License grants you permission to propagate or
modify any covered work.  These actions infringe copyright if you do
not accept this License.  Therefore, by modifying or propagating a
covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically
receives a license from the original licensors, to run, modify and
propagate that work, subject to this License.  You are not responsible
for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an
organization, or substantially all assets of one, or subdividing an
organization, or merging organizations.  If propagation of a covered
work results from an entity transaction, each party to that
transaction who receives a copy of the work also receives whatever
licenses to the work the party's predecessor in interest had or could
give under the previous paragraph, plus a right to possession of the
Corresponding Source of the work from the predecessor in interest, if
the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the
rights granted or affirmed under this License.  For example, you may
not impose a license fee, royalty, or other charge for exercise of
rights granted under this License, and you may not initiate litigation
(including a cross-claim or counterclaim in a lawsuit) alleging that
any patent claim is infringed by making, using, selling, offering for
sale, or importing the Program or any portion of it.

11. Patents.

A "contributor" is a copyright holder who authorizes use under this
License of the Program or a work on which the Program is based.  The
work thus licensed is called the contributor's "contributor version".
```

(continues on next page)

(continued from previous page)

```
A contributor's "essential patent claims" are all patent claims
owned or controlled by the contributor, whether already acquired or
hereafter acquired, that would be infringed by some manner, permitted
by this License, of making, using, or selling its contributor version,
but do not include claims that would be infringed only as a
consequence of further modification of the contributor version.  For
purposes of this definition, "control" includes the right to grant
patent sublicenses in a manner consistent with the requirements of
this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free
patent license under the contributor's essential patent claims, to
make, use, sell, offer for sale, import and otherwise run, modify and
propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express
agreement or commitment, however denominated, not to enforce a patent
(such as an express permission to practice a patent or covenant not to
sue for patent infringement).  To "grant" such a patent license to a
party means to make such an agreement or commitment not to enforce a
patent against the party.

If you convey a covered work, knowingly relying on a patent license,
and the Corresponding Source of the work is not available for anyone
to copy, free of charge and under the terms of this License, through a
publicly available network server or other readily accessible means,
then you must either (1) cause the Corresponding Source to be so
available, or (2) arrange to deprive yourself of the benefit of the
patent license for this particular work, or (3) arrange, in a manner
consistent with the requirements of this License, to extend the patent
license to downstream recipients.  "Knowingly relying" means you have
actual knowledge that, but for the patent license, your conveying the
covered work in a country, or your recipient's use of the covered work
in a country, would infringe one or more identifiable patents in that
country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or
arrangement, you convey, or propagate by procuring conveyance of, a
covered work, and grant a patent license to some of the parties
receiving the covered work authorizing them to use, propagate, modify
or convey a specific copy of the covered work, then the patent license
you grant is automatically extended to all recipients of the covered
work and works based on it.

A patent license is "discriminatory" if it does not include within
the scope of its coverage, prohibits the exercise of, or is
conditioned on the non-exercise of one or more of the rights that are
specifically granted under this License.  You may not convey a covered
work if you are a party to an arrangement with a third party that is
in the business of distributing software, under which you make payment
to the third party based on the extent of your activity of conveying
the work, and under which the third party grants, to any of the
parties who would receive the covered work from you, a discriminatory
patent license (a) in connection with copies of the covered work
conveyed by you (or copies made from those copies), or (b) primarily
for and in connection with specific products or compilations that
```

(continues on next page)

(continued from previous page)

```
contain the covered work, unless you entered into that arrangement,
or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting
any implied license or other defenses to infringement that may
otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not
excuse you from the conditions of this License.  If you cannot convey a
covered work so as to satisfy simultaneously your obligations under this
License and any other pertinent obligations, then as a consequence you may
not convey it at all.  For example, if you agree to terms that obligate you
to collect a royalty for further conveying from those to whom you convey
the Program, the only way you could satisfy both those terms and this
License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have
permission to link or combine any covered work with a work licensed
under version 3 of the GNU Affero General Public License into a single
combined work, and to convey the resulting work.  The terms of this
License will continue to apply to the part which is the covered work,
but the special requirements of the GNU Affero General Public License,
section 13, concerning interaction through a network will apply to the
combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of
the GNU General Public License from time to time.  Such new versions will
be similar in spirit to the present version, but may differ in detail to
address new problems or concerns.

Each version is given a distinguishing version number.  If the
Program specifies that a certain numbered version of the GNU General
Public License "or any later version" applies to it, you have the
option of following the terms and conditions either of that numbered
version or of any later version published by the Free Software
Foundation.  If the Program does not specify a version number of the
GNU General Public License, you may choose any version ever published
by the Free Software Foundation.

If the Program specifies that a proxy can decide which future
versions of the GNU General Public License can be used, that proxy's
public statement of acceptance of a version permanently authorizes you
to choose that version for the Program.

Later license versions may give you additional or different
permissions.  However, no additional obligations are imposed on any
author or copyright holder as a result of your choosing to follow a
later version.

15. Disclaimer of Warranty.
```

(continues on next page)

(continued from previous page)

```
THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY
APPLICABLE LAW.  EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT
HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY
OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO,
THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE.  THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM
IS WITH YOU.  SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF
ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS
THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY
GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE
USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF
DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD
PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS),
EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF
SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided
above cannot be given local legal effect according to their terms,
reviewing courts shall apply local law that most closely approximates
an absolute waiver of all civil liability in connection with the
Program, unless a warranty or assumption of liability accompanies a
copy of the Program in return for a fee.

                     END OF TERMS AND CONDITIONS

            How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest
possible use to the public, the best way to achieve this is to make it
free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program.  It is safest
to attach them to the start of each source file to most effectively
state the exclusion of warranty; and each file should have at least
the "copyright" line and a pointer to where the full notice is found.

  <one line to give the program's name and a brief idea of what it does.>
  Copyright (C) <year>  <name of author>

  This program is free software: you can redistribute it and/or modify
  it under the terms of the GNU General Public License as published by
  the Free Software Foundation, either version 3 of the License, or
  (at your option) any later version.

  This program is distributed in the hope that it will be useful,
  but WITHOUT ANY WARRANTY; without even the implied warranty of
  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
  GNU General Public License for more details.
```

(continues on next page)

(continued from previous page)

```
  You should have received a copy of the GNU General Public License
  along with this program.  If not, see <http://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program
into proprietary programs.  If your program is a subroutine library, you
may consider it more useful to permit linking proprietary applications with
the library.  If this is what you want to do, use the GNU Lesser General
Public License instead of this License.  But first, please read
<http://www.gnu.org/philosophy/why-not-lgpl.html>.
```

CHAPTER 2

Indices and tables

- genindex
- modindex
- search