HIEv User Manual

For software version 1.8 %%% Confirm SW version number

%%% Date April 2013 | v1.2

%%% TODO:

Replace all screen dumps

MAC or Windows?

What example data and login account should be used for screen dumps?

Should screen dumps show functions only available to Admins?

There has been a significant rise in the number of sensors and sensor networks used in environmental research in recent years. This growth has brought with it the challenge of managing sensor infrastructure and the data produced by the increasing numbers of deployed sensors.

The HIEv system was developed to address these challenges and specifically aims to:

* Ensure raw data is never lost
* Ensure that data can be used and interpreted in the future
* Allow researchers to make linkages between different types of data
* Make it easier for researchers to get access to the data they need
* Make it easier for technical officers to distribute the data
* Create entries in Research Data Australia (required by ANDS)

As a web application, HIEv is easily accessed from a variety of locations and platforms – all you need is a modern web browser and the URL of the server hosted by the research centre with which you are affiliated /collaborating.

Contents

[1 Overview 4](#_Toc351628659)

[1.1 Installing HIEv 4](#_Toc351628660)

[2 Logging in to the system 5](#_Toc351628661)

[3 The HIEv Main Screen 7](#_Toc351628662)

[3.1 Signing Out 8](#_Toc351628663)

[3.2 Changing Your User Settings 8](#_Toc351628664)

[3.2.1 Overview Tab 8](#_Toc351628665)

[3.2.2 Edit Details Tab 8](#_Toc351628666)

[3.2.3 Change Password Tab 9](#_Toc351628667)

[4 Managing Data Files 10](#_Toc351628668)

[4.1 The Dashboard Tab 10](#_Toc351628669)

[4.2 The Explore Data Tab and File Searching 10](#_Toc351628670)

[4.2.1 Sorting 11](#_Toc351628671)

[4.2.2 Searching 11](#_Toc351628672)

[4.3 The Cart 17](#_Toc351628673)

[4.3.1 Editing the Cart 17](#_Toc351628674)

[4.4 Viewing and Editing a File's Metadata 17](#_Toc351628675)

[4.5 Deleting a file 21](#_Toc351628676)

[5 Publishing Your Data 23](#_Toc351628677)

[5.1 Creating a Package 24](#_Toc351628678)

[5.2 Editing a Package’s metadata 25](#_Toc351628679)

[5.3 Publishing a Package 25](#_Toc351628680)

[5.4 Managing Published Packages 25](#_Toc351628681)

[5.5 Viewing Published data 25](#_Toc351628682)

[6 Downloading files 27](#_Toc351628683)

[7 Facilities and Experiments 28](#_Toc351628684)

[7.1 Creating and Editing Facilities 28](#_Toc351628685)

[7.2 Creating and Editing Experiments 29](#_Toc351628686)

[7.3 Setting Up Experiment Parameters 32](#_Toc351628687)

[8 Uploading Data files 35](#_Toc351628688)

[8.1 Uploading RAW TOA5 data files 37](#_Toc351628689)

[8.2 Automating the upload of data to HIEv 38](#_Toc351628690)

[9 System Administration 39](#_Toc351628691)

[9.1 The Users Tab 39](#_Toc351628692)

[9.2 The Access Requests tab 40](#_Toc351628693)

[9.3 The Column Mappings tab 41](#_Toc351628694)

[10 Modifying Tags, Column Mappings and Experiment Parameters 43](#_Toc351628695)

[11 Migrating data to a new system 45](#_Toc351628696)

[12 Revision History 46](#_Toc351628697)

[Appendix A - The Bagit format 47](#_Toc351628698)

[Appendix B - RIF-CS 48](#_Toc351628699)

[Appendix C - Data File Upload Scenarios 49](#_Toc351628700)

1. Overview

HIEv is designed to act as a central repository for environmental research data. Technicians can configure their field PCs to automatically push time-series data from sensors into HIEv, while researchers can use the system to discover and download the latest data available. Rich metadata is stored for physical infrastructure (“Facilities”), the Experiments that run at those facilities, as well as the individual files to support discovery and interpretation.

DC21 Diagram

All files in HIEv are grouped by Experiment, providing a convenient way to organise related files such as cleansed or gap-filled data, and analysis outputs.

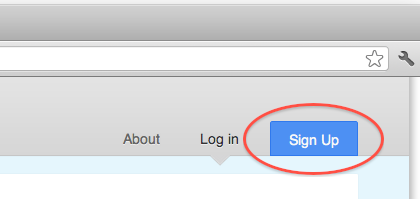
Once finalised, collections of data can be defined, described and published to ANDS. This enables researchers from outside the organisation that produced the data to discover it, and to request access to download a copy.

* 1. Installing HIEv

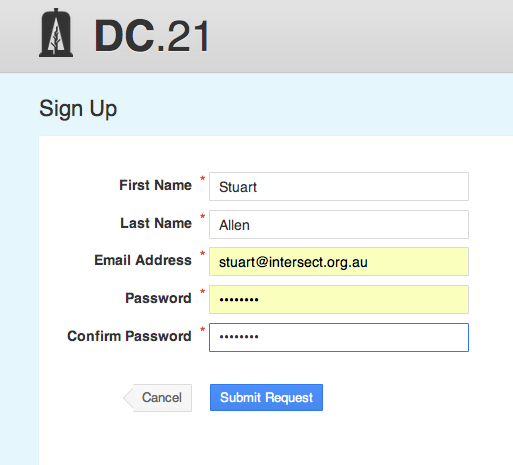
All instructions for installing and upgrading HIEv are held at [Deployment Guide](https://github.com/IntersectAustralia/dc21/wiki/Deployment-Guide) on the project’s [GitHub Wiki](https://github.com/IntersectAustralia/dc21/wiki).

1. Logging in to the system

To begin using HIEv, enter the system URL %%% What is it? into your web browser. Before you can login you are required to have a system account. You can apply for an account by clicking the blue "Sign Up" button on the top left of the screen:

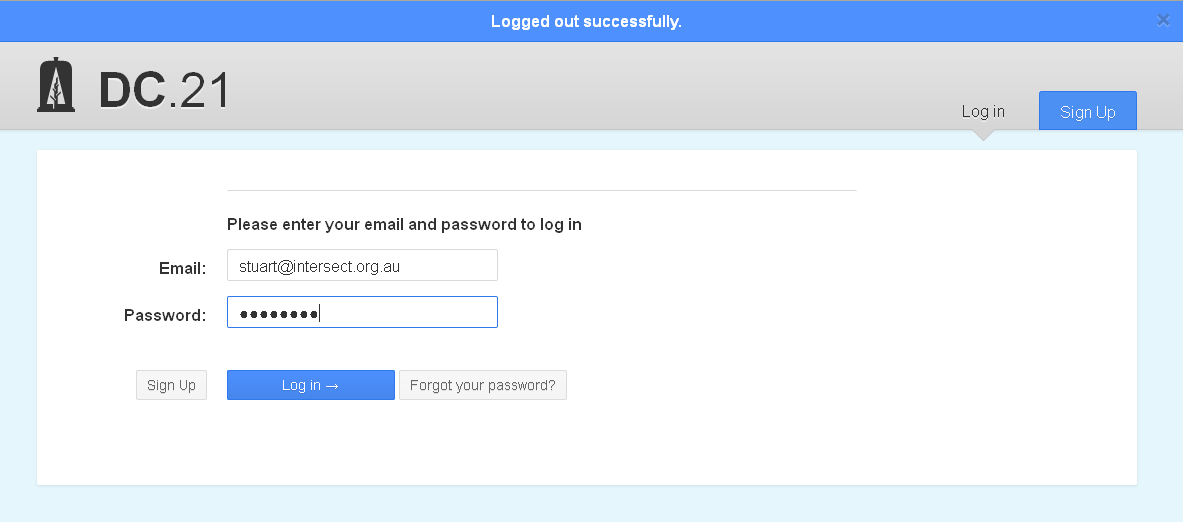


This will take you to a form where you will be requested to enter your first name, last name, email address and chosen password. (Note: Passwords must be between 6-20 characters and contain at least one each of: an uppercase letter; a lowercase letter; a digit and a symbol.)



Once you have filled out the form and clicked "Submit Request" an email will be sent to the system administrator who will either approve or deny your request for access. If your request is approved you will receive an email informing you that you can now login using the password you entered on the original sign up form.

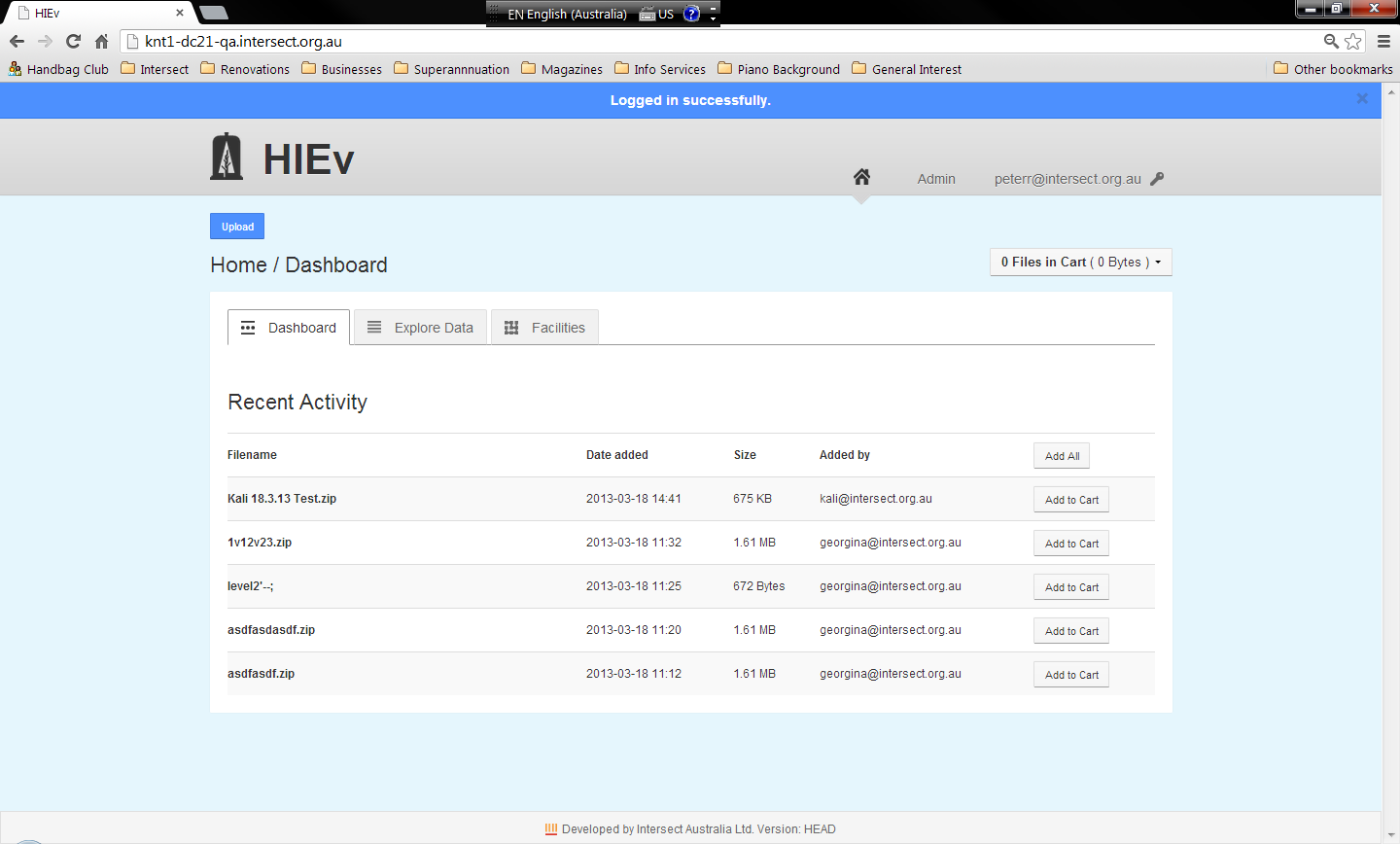
To see the login form, make sure you have the "Log in" tab selected on the top right. Next enter your Email address and password and click the blue "Log in" button below:



Once you have logged in you will be taken to the main screen for the HIEv application.

1. The HIEv Main Screen

The Main Screen consists of the following parts:

Figure : HIEv Main Screen

%%% Add reference numbers to this diagram when the final cut & paste is done.

Most of these components are common to many screens in the HIEv system.

|  |  |  |
| --- | --- | --- |
| 1 | Home button | Click to show the Home/Dashboard (as shown in the view above). |
| 2 | Admin button | Click to access Admin functions (see Chapter ). This button is only present if you have Administrator permissions. |
| 3 | Login ID | This is your login name. Click to open a drop down menu of user operations. (See sections 3.1 and 3.2) |
| 4 | Action button | In many screens, there is an action button at the top left corner. It is often an Upload button, which allows you to upload new data files to HIEv. (See Chapter for more information.) However, it may also be a button for another function which is more relevant to the data being displayed in the current view. |
| 5 | Cart status box | The HIEv web interface allows you to add files to a Cart, which operates like an e-Commerce shopping cart. Click in the Cart Status to open a drop down menu of Cart functions. (See section ) |
| 6 | Working area | This contents of this work area changes as you perform HIEv operations. |
| 7 | Version indicator | This shows the version of HIEv which you are accessing. |

* 1. Signing Out

Click on your login ID at the top right of the screen to see a drop down menu. Click on Sign Out to finish your HIEv session.

* 1. Changing Your User Settings

Click on your login ID at the top right of the screen to see a drop down menu. Click on Settings to access the following three tabs.

When finished, click on the Home button to return to the HIEv Main Screen.

* + 1. Overview Tab

This tab displays a summary of your user information.

%%% Screen dump of tab

|  |  |
| --- | --- |
| User Name: | Your valid email address which you use for logging on. |
| First Name Last Name: | Your name. |
| API Token: | A string of characters which you can use as an authorisation token in scripts which you write to make use of the HTML API for HIEv. See the API definition on the GitHUB WIKI for DC21/HIEv for instructions on using the HTML API.  Clicking on Generate Token will cause a token to be displayed in this field. Copy and paste it into the required place in your API scripts.  Clicking on Re-generate Token will cause the current token to be invalidated and a new token to be generated and displayed. You must replace the token value in your API scripts with this new token so that your API scripts continue to work. This button is only displayed if a valid token is available.  Clicking on Delete Token will invalidate the displayed token. Your API scripts will no longer work. This button is only displayed if a valid token is available. |

* + 1. Edit Details Tab

The Edit Details tab allows you to update your First Name and Last Name. Click Cancel to return to the Overview Tab without accepting changes, and click Update to store the changed values you’ve entered.

%%% Screen dump of tab

* + 1. Change Password Tab

Use this tab to change your HIEv logon password. You must correctly enter your current password and the values you enter for New password and Confirm new password must be identical for your password change request to be processed. Click Cancel to return to the Overview Tab without accepting changes, and click Update to store the changed password you’ve entered.

%%% Screen dump of tab

1. Managing Data Files

%%% DC21-514 mentions new mouse hover function over truncated fields. However, it probably doesn’t require documentation.

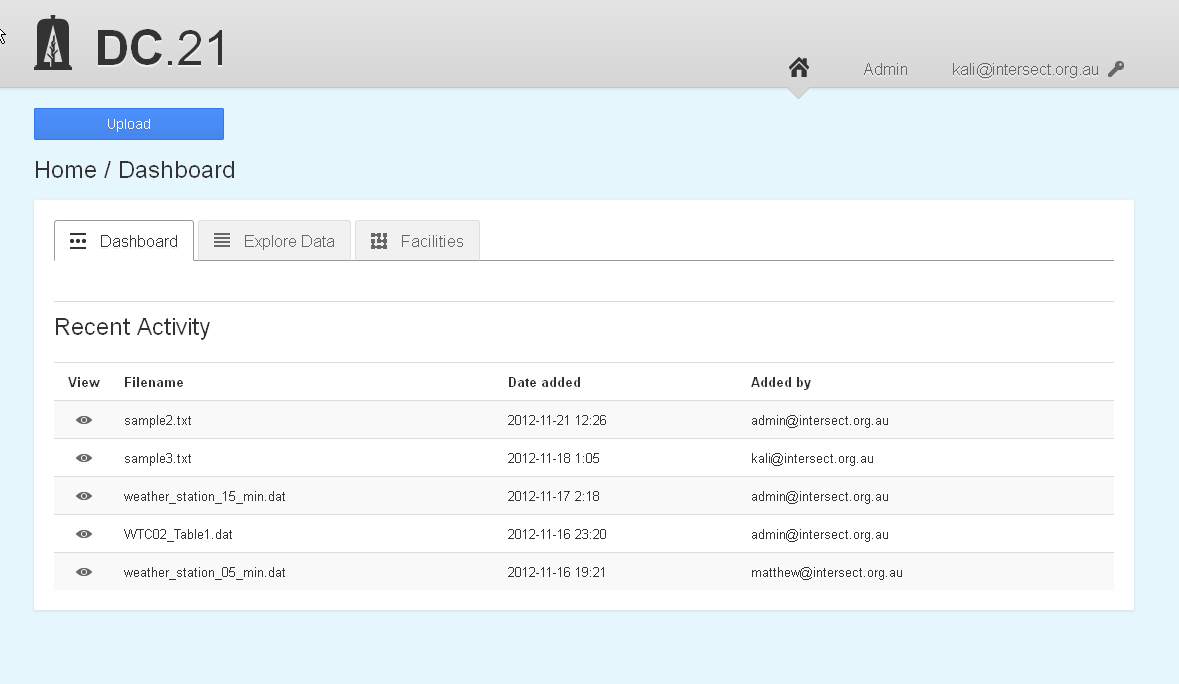
The key views of the files loaded into the HIEv database are the Dashboard and Explore Data views. These views allow you to perform the key functions of the HIEv system.

Access these views by clicking on their respective tabs on the HIEv Home Screen.

* 1. The Dashboard Tab

The default tab on the Home Screen is the Dashboard tab. It shows a list of the five files which have most recently been uploaded by the users of the HIEv system.

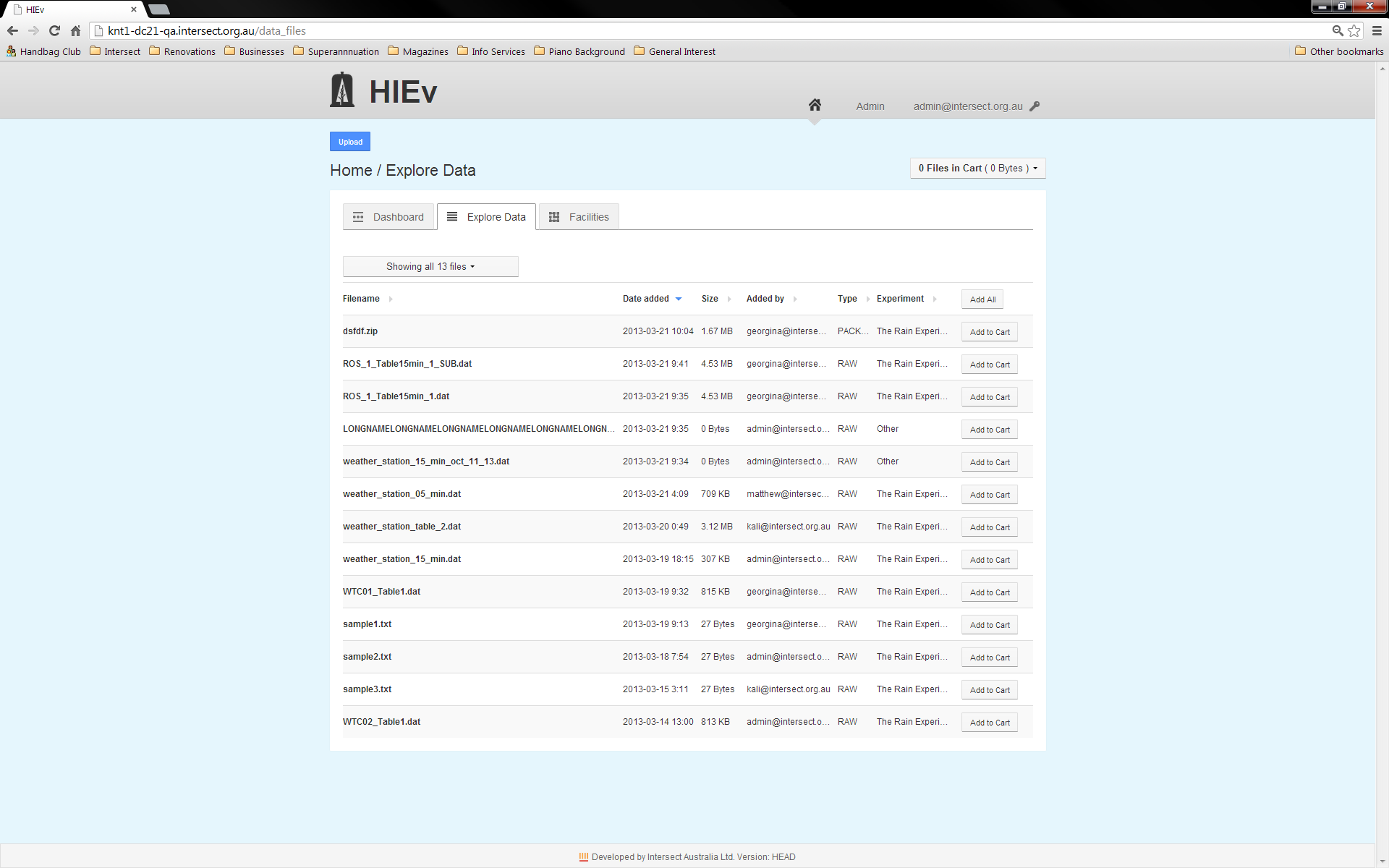
The operation of the Dashboard tab is similar to the operation of the Explore Data tab, except that searching and sorting functions are not supported. Therefore, please see the description of the Explore Data tab for more information.



%%% Are there ever other sections besides “Recent Activity”?

* 1. The Explore Data Tab and File Searching

The Explore Data tab provide the main data management functions of the HIEv system. The initial view shows all data files which have been uploaded. If there are more than fit on one screen, only the first 30 files will be shown, and the subsequent files can be shown by paging through the data using the page number buttons.



%%% Annotated screen dump

* + 1. Sorting

Click on the heading of any column in the file list in this view to sort the files into increasing order for that column. Click again to reverse the sort order.

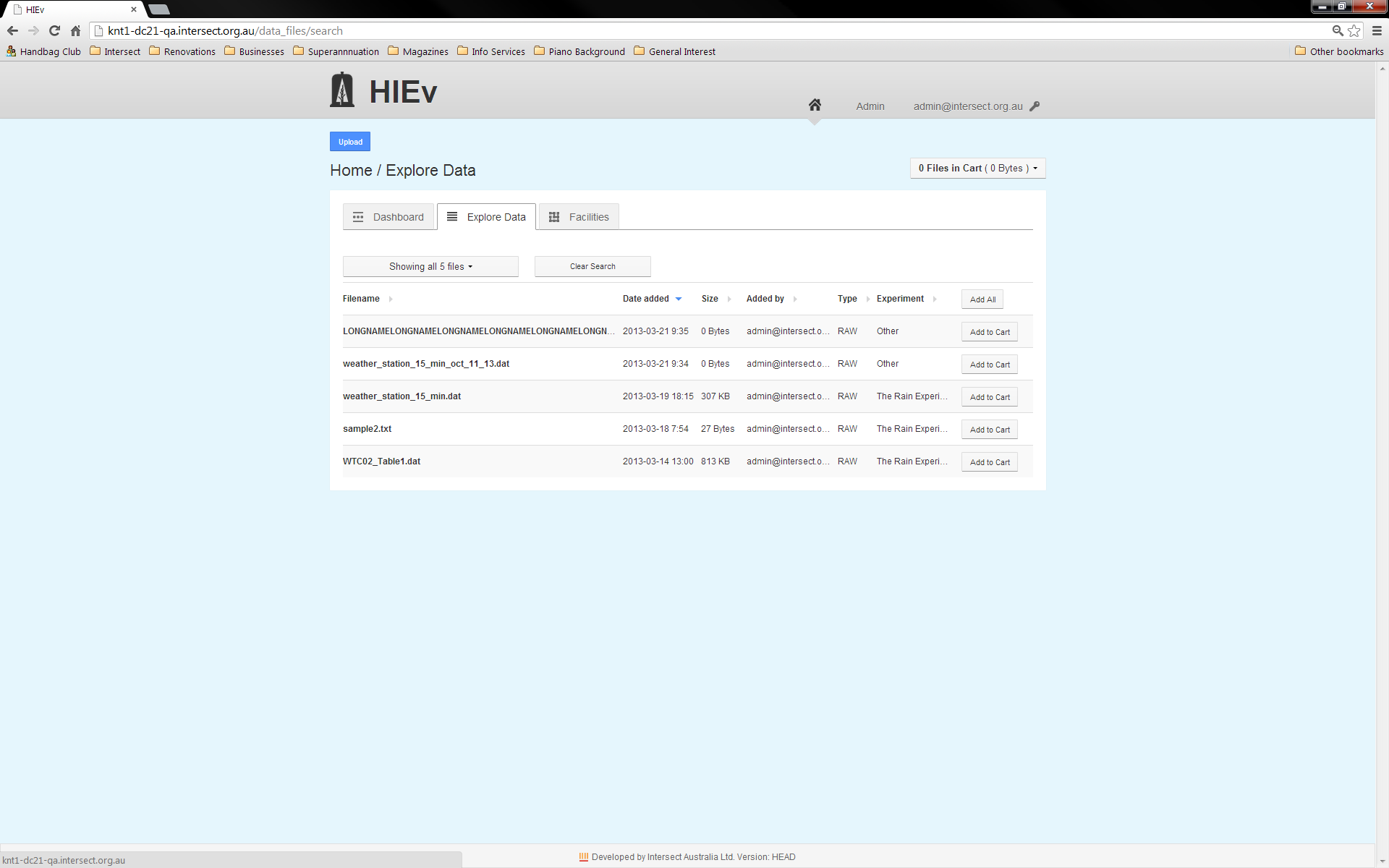
A triangular up or down arrow to right of any file list column heading indicates the active sort order. A grey right-pointing arrow indicates that the file list is unsorted in that column.

Re-sorting the data always resets the display to the first page of the file list.

Sorting can be done by only one file list column at a time.

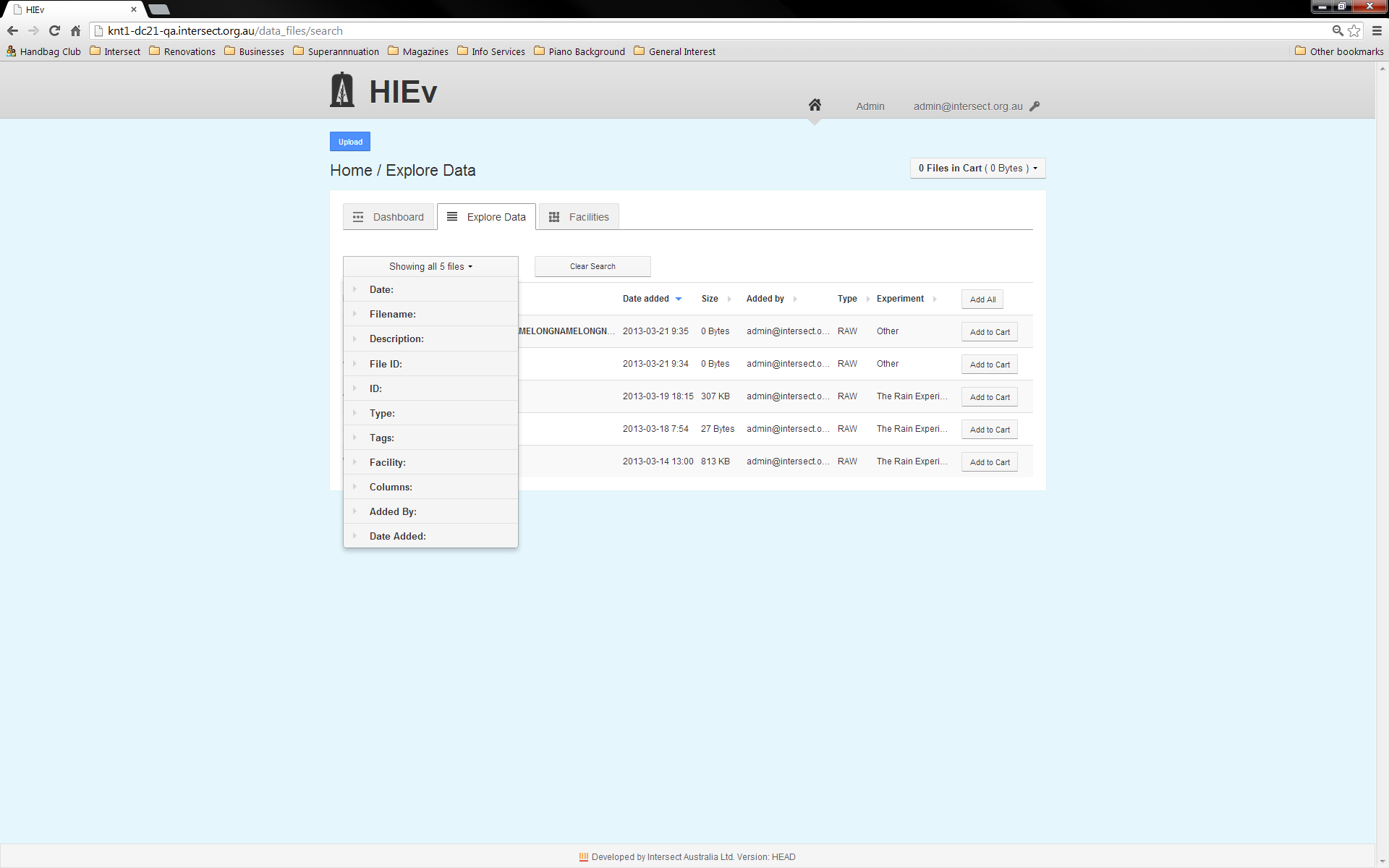
* + 1. Searching

When the number of files uploaded to your system becomes large, finding the file you are interested in may become difficult. The Search facility is provided to assist. It supports searching for data files using the metadata that was supplied at the time each file was uploaded.



The screen above shows the Explore Data tab when a search is active, restricting the number of files shown. Note the Clear Search button, which is only present when a search is active. Click this button to return to displaying all files.

To change the search conditions, click on the Showing ... files button. (The exact text of this button changes, depending on how many files are presently shown in the file list.) The search parameters box will be shown.



If there is a search active, one or more of the search categories in this menu will be expanded when you first display it.

Click on the metadata field you wish to search by to expand its display its search parameters.

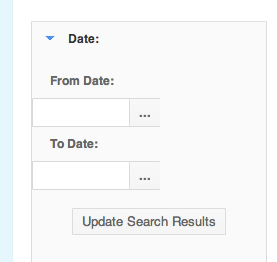
If you specify more than one search condition using more than one metadata field, the file list will display only those files which satisfy all of the conditions you specify.

* + - 1. Regular Expressions

%%% They’re in, so give a few useful examples, and a link to the full definition.

* + - 1. Restricting by data Date

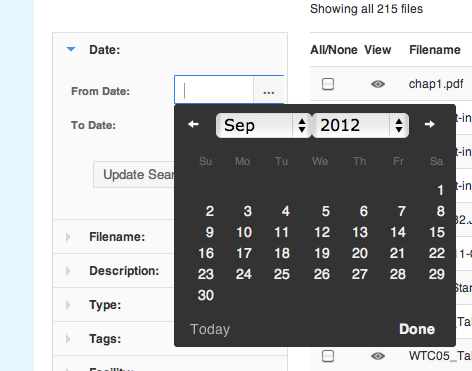
The **Date** field allows you to search for files based on the start and end date specified in the file’s metadata:



A date can be entered in either the **From Date**, **To Date** or both. If only a **From Date** is specified, all files containing data for after that date will be included. If only a **To Date** is specified, all files containing data for before that date will be included.

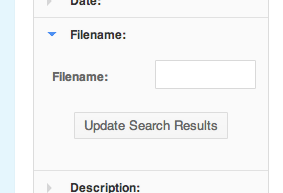
If you restrict by date, files which have no dates in their metadata will not be displayed.

Dates can be typed into the **From Date** and **To Date** fields in *YYYY-MM-DD* format or by clicking on the ellipsis to the right and selecting a day from the calendar that is displayed:



* + - 1. Restricting by Filename substring

The **Filename** field allows you to search for files based on their filename:

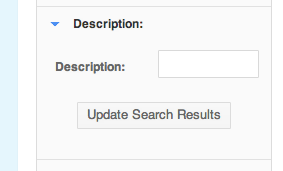


%%% Regular expressions

Once the search results are updated, only files that contain the supplied text string in their filename will be returned. This will include partial matches. For example, if the string "txt" was entered, only files that have "txt" in their name would be shown, regardless of where within the file name that substring occurs.

* + - 1. Restricting by Description substring

The **Description** field allows you to search for files based on their free-form text descriptions:

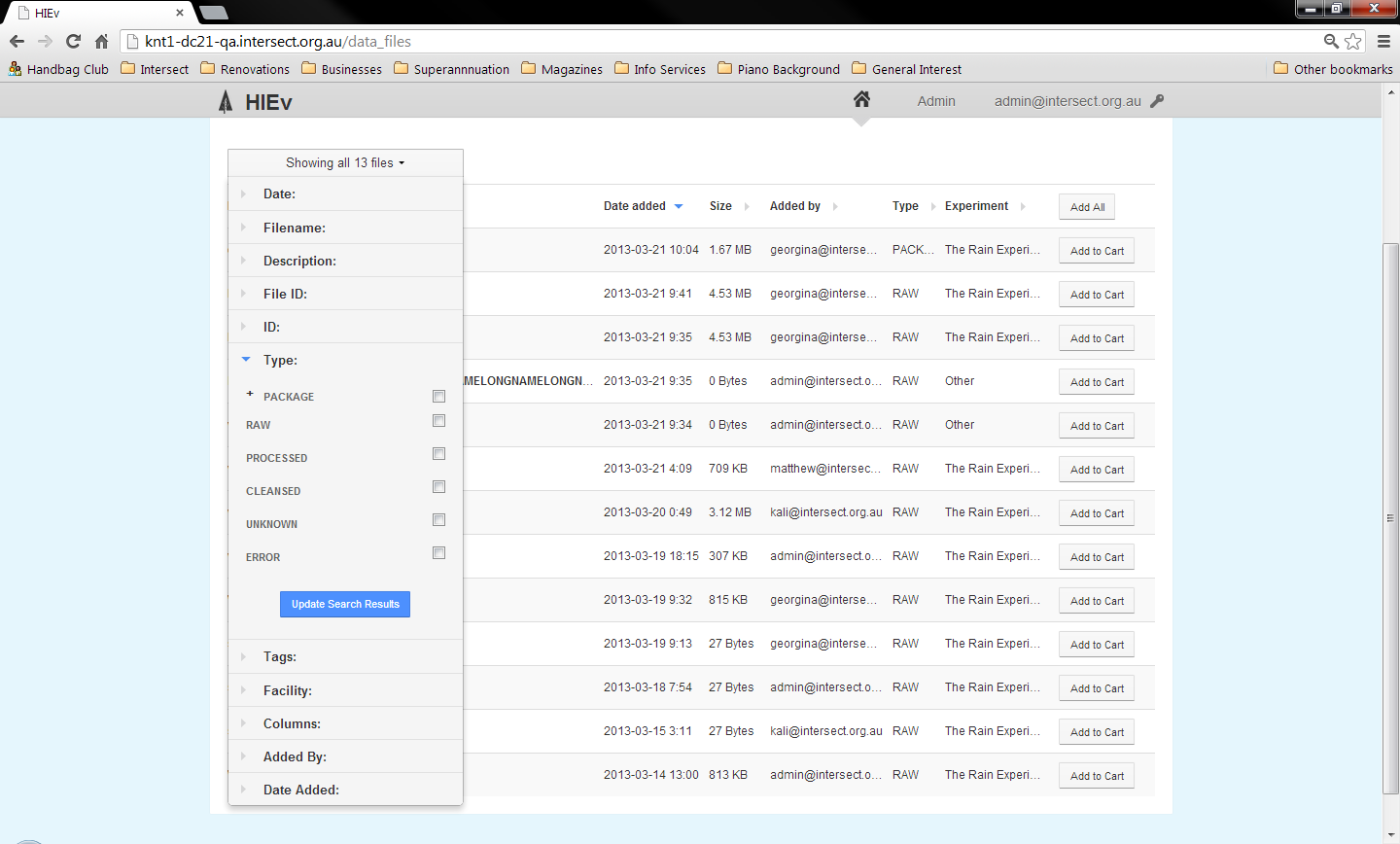


%%% Regular expressions

Like the filename search, the string supplied only needs to be a partial match with the description of a file for it to be included in the search results.

* + - 1. Restricting by file Type

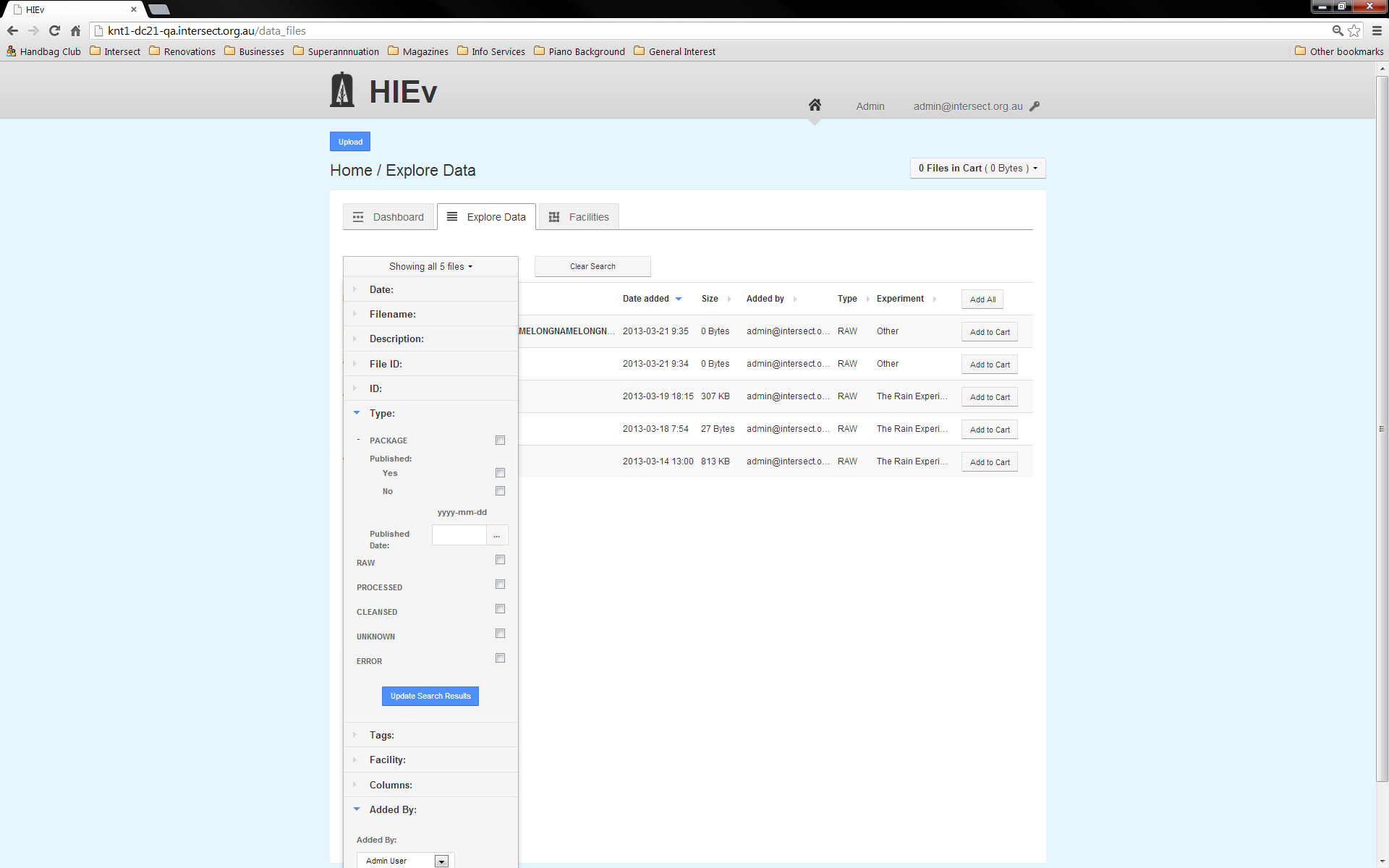
The **Type** interface allows you to search for files based on their specified type:



The set of possible types is displayed as a list of checkboxes. Selecting none of the checkboxes is the same as selecting them all - files will not be filtered based on their type. Once at least one checkbox has been selected, only files of that type will be returned in the search results. More than one type can be selected.

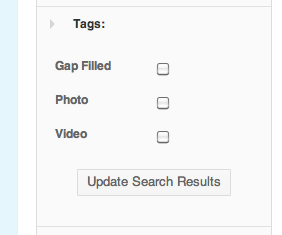
Click on the + sign to the left of the Package option to open further search conditions for Packages. (Clicking on the word Package sets its checkbox.) Select the Yes or No checkbox to display only files which are, or are not, Published. To show a Package file regardless of its Publish status, leave both checkboxes unchecked. If a Published Date is entered, then only files Published on that date will be displayed. Again, click on the ellipses to the right of the Published Date box to open a date entry dialog.

%%% The operation of this package selection functionality is buggy, and it’s not clear if it will be fixed prior to end of Sprint 8. Delaying until the situation is clear.



* + - 1. Restricting by Tag

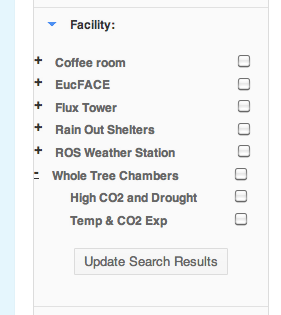
The **Tags** interface allows you to search on the tags that have been given to a file:



Like the Type interface, selecting none of the available checkboxes means that files will be returned in the search results regardless of the tags they have. Once a checkbox is selected, only files that have the corresponding tag will returned. More than one tag can be selected at any given time.

* + - 1. Restricting by Facility

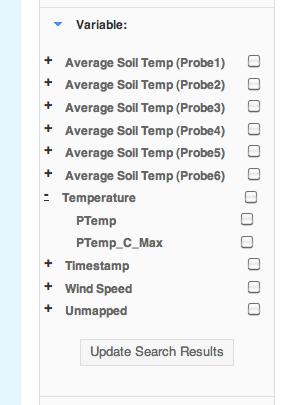
The **Facility** interface allows you to search for files based on the facility or experiment that produced the file:



The interface is a two-level hierarchy of checkboxes. The top level represents all the facilities in the system and the second level represents the experiments that are running, or have run, at each of the facilities. Selecting a facility selects all of the experiments for that facility. If only specific experiments are required, clicking on the plus sign to the left of a facility will expand the hierarchy and allow individual experiments to be selected or deselected.

* + - 1. Restricting by data Columns

The **Columns** interface allows you to search for TOA5 format data files that contain the specified columns:



Like the Facility interface, the variable is a two-level hierarchy of checkboxes. The top level contains all the standardised variable names that column headers are mapped to. There is also a special top-level group called **Unmapped** that contains all the headers that are not mapped to a standard name.

Selecting the checkbox for a standardised top-level variable name will select all the variables (TOA5 column names) that are mapped to it. Clicking the plus sign to the left of the standardised variable will show all the columns mapped to it and allow you to select them individually.

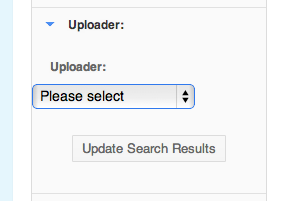
* + - 1. Restricting by other things too %%%

%%%There are File ID and ID as well, and possibly more.

%%% ID uses regular expressions.

* + - 1. Restricting by person who added the file

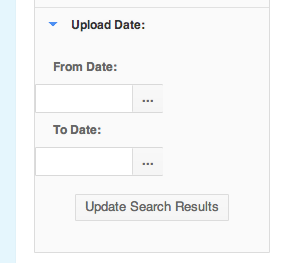
The **Added By** field allows you to search for files that were uploaded by a specific user:



The user must be selected from a list of a users registered in the system.

* + - 1. Restricting by Upload Date

The **Date Added** interface allows you to search for files that were uploaded between a specified set of dates:



Like the **Date** field, the **Date Added** interface allows you to select a **From date** and a **To Date**. If both dates are specified only files uploaded between those dates will be included in the search results. If only a **From Date** is specified, all files uploaded after that date will be included. If only a **To Date** is specified, all uploaded before that date will be included.

* 1. The Cart

The Cart operates like an e-Commerce shopping cart. HIEv provides functions for adding files to the Cart and for doing operations, such as Downloading and Publishing, on all files in the Cart. See more information about Downloading in Chapter and Publishing in Chapter .

Note There is one Cart per user account. If two people use the same login account, their operations on the Cart will interfere with one another.

Add a file to the Cart by clicking on any Add to Cart button for that file. There are Add to Cart buttons in multiple places, including the Dashboard file list and the Explore Data file lists.

The Cart status box shows the number of files in your Cart and the total size of these files. Click on this Cart status box, which appears at the top right of many HIEv screens, to show a dropdown menu of operations which can be performed on the files in the Cart. These operations are:

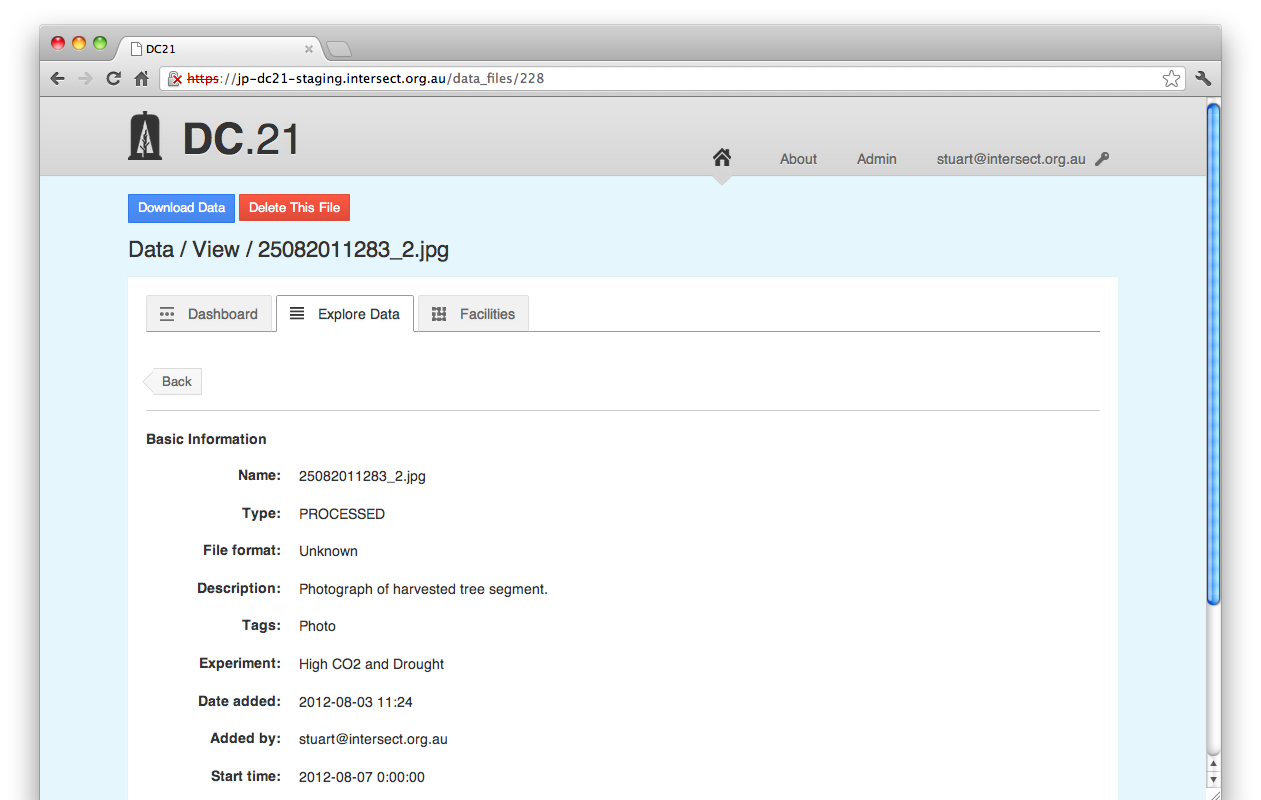
|  |  |
| --- | --- |
| Download | Click on this option to download data files to your local computer. See Chapter . |
| Package | Click on this option to create a publishable Package containing all files in the Cart. See section . |
| Clear cart | Click on this option to remove all files from the Cart. |
| Edit cart | Click on this option to view the Cart contents, remove individual files from the Cart, download all files in the Cart or create a publishable Package using all files in the Cart. See section for more details. |

* + 1. Editing the Cart

%%%

* 1. Viewing and Editing a File's Metadata

Clicking on any filename in the Filename column of the Dashboard or Explore Data tabs will display the metadata for that file in a screen similar to the following:



%%% All files have Basic Information. TOA5 files also have **Information From The File** and **Columns** sections.

%%% I think we need a Metadata section which describes the Basic Information and is referenced from here and elsewhere in the doc.

The fields displayed in the metadata view are dependent on the type of file selected. The following list is not complete, but explains some of the less obvious fields.

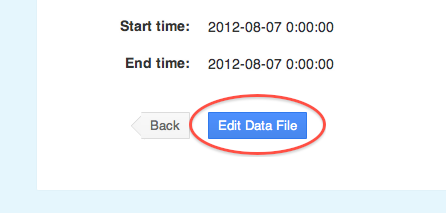
|  |  |
| --- | --- |
| **Name** | The name of the file as it is stored in the HIEv system. |
| **Type** | The Type of the file is a single value that describes the data contained within the file. This value is chosen from a constrained list of possibilities defined by the system administrator. The file's **Type** is generally aimed at tracking data through its various stages of processing. |
| **File Format** | If the file format is known, it is shown here. Possible values include:  TOA5 The file was inspected on upload and discovered to be TOA5 format. TOA5 files are processed differently on file upload. See %%%  BAGIT The file is a Package which is formatted as a BAGIT ZIP file. See Appendix %%% for more information on BAGIT files.  UNKNOWN The file format is not known to the HIEv system. |
| **Description** | The human-entered description of the file. %%% Is this true for TOA5 files? |
| **Tags** | The file Tags are a set of flags that have been given to the file from a constrained list of possibilities defined by the system administrator. |
| **Experiment** | The Experiment field indicates which experiment produced the file. Each file must be associated with exactly one experiment. Any user with the appropriate %%% What’s appropriate? permissions can create experiments. |
| **File ID** | File IDs are unique integers which are assigned and used internally by the HIEv system to identify files. File IDs cannot be changed by the user. In general, they will not change, but in the case of TOA5 files, may change after uploading further TOA5 data. |
| **ID** | This ID is a character string which is used outside HIEv to identify this Package and is stored in HIEv as a reference. It is set and can be changed by the user on the Edit Metadata screen. No two files in the HIEv system can have the same non-null ID. See section for more information about the use of this ID. |
| **Date added Added by** | The Date added field indicates the date that the file was uploaded to the HIEv system and the Added by field indicates the user that uploaded the file. For Packages, it’s the date on which it was Packaged and user who packaged it. |
| **Start time End time** | The Start time and End time fields indicated the range of the data contained within the file. For non-TOA5 these dates must be manually entered by the uploader. For TOA5 file this information is automatically extracted from the file itself along with the following extra pieces of information: %%% Really? |
| **Sample interval field** | The frequency of samples in the data file if relevant . |
| **Datalogger model** | The model of data logger used to generate a TOA5 file. |

%%% There are other fields, such as Facility, Size, Added by. Are Start time and End time still displayed? Aren’t the fields displayed dependent on the type of the file? There is also Information From The File and Column information. Should we bother to carefully define only such a small selection of the possible displayed data? Where are the Column Mapping contents and Units and Measurement Type abbreviations explained or defined?

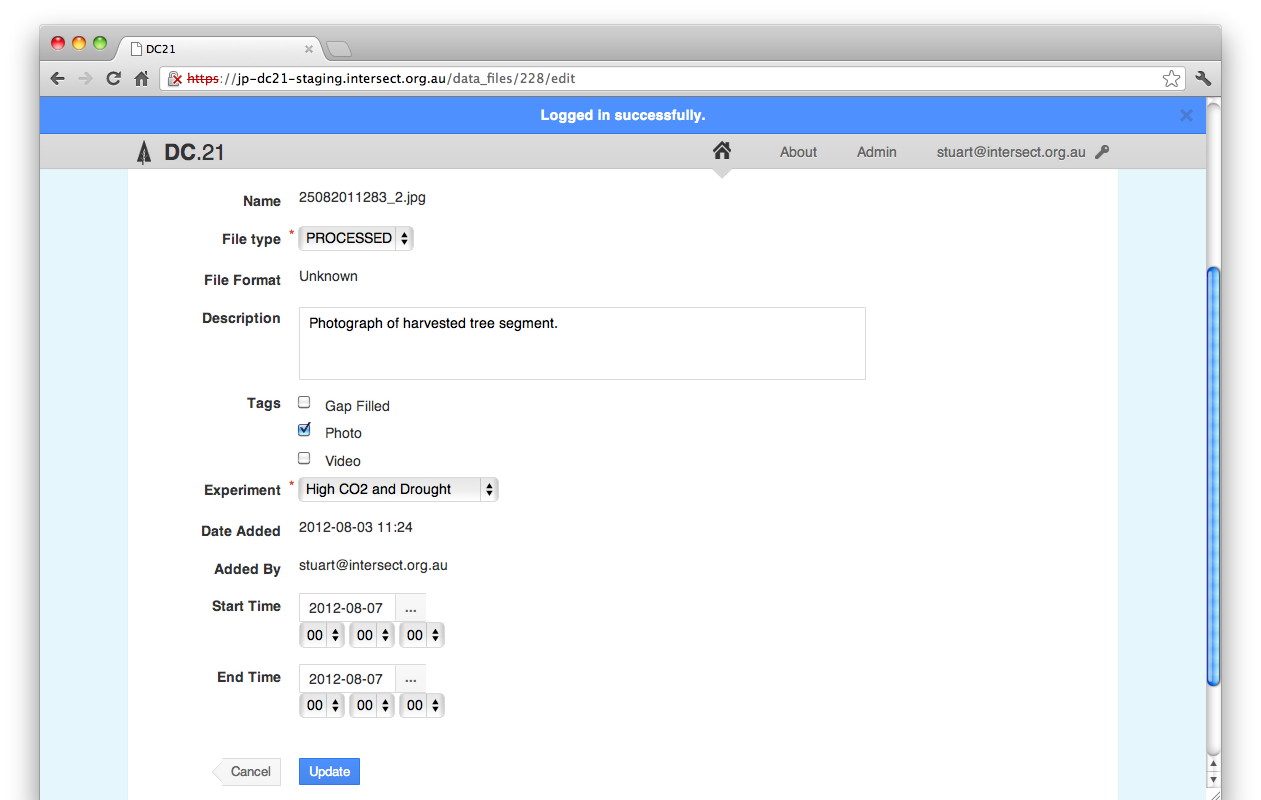
%%% Also Add to Cart from this screen, but Cart functionality may not be defined until later in the doc, so forward reference.

There is a button which says Add to Cart. Click this button to add this file to your Cart. If the file is already in your Cart, then the button will change to say Remove from Cart, and you can use it to remove the file from your Cart.

At the bottom of the screen is a button to edit the file's metadata:



This button will take you to a form that allows you to modify the file's metadata:



Once you have finished editing the metadata, click the "Update" button to save your changes.

%%% DC21-500 – Changes to File id and ID. What are these things? Who to ask?

Background: file ids are thrown away when a T0A5 file is updated nightly - a new file\_id is assigned. This means that file URLs are not permanent. This impacts download and also has implications for packaging.

I spoke with Remko Duursma and Gerry Devine (new data manager). UWS have agreed to regard the file id as a transient handle to download the file after an API search. If the download fails because the file has since been replaced, they will redo the search. The applciation should instead use a separate identifier as the permanent reference.

Changes to be implemented in sprint 8, plus some business rules and notes follow.

1. On HIEV file Data/View screen, there should be two additional fields displayed: File Id and ID

File Id: this is te file's internal file identifier

ID: this is a new text field, up to 1000 characters in length. If the required width to display the value is greater than the width of the screen, the field should be truncated (with ... showing at the end).

It should be possible to hover over the fields to see their full value.

These fields should be displayed after the Size field.

2. NOTE: In general, the ID field will apply to HIEv Package files, but as HIEV Package are treated just like any other file the functnioality will apply to all files.

3. ID edit rules

If the user presses Edit Metadata, assuming they have the right permission to edit the metadata, the ID field should be available to be changed.

Field Validation : no special validation should aply to ithe ID field, except for any "special" characters which are known to cause problems with the UI.

BUSINESS RULE: no two files can be given the same ID value within the Dc21 app. This should be repoted when the user attempts to Save any metadata changes. Error message: This ID is already is use by file <filename>..."

4. BUSINESS RULE: It should not be possible to change the contents of, replace the contents of or delete a HIEv file which has an ID, ie not by the UI or upload func. So, if a user wants to delete a file which has an ID, they must explicitly remove the ID value first and then delete it, ie: 2 steps.

6. BUSINESS RULE: The implication of 4, above, for TOA5 files which are uploaded nightly is that if the upload will replace a pre-existing TOA5 file with an ID, the upload should be considered "unsafe", ie: same processing as for T0A5 files considered unsafe, ie: file type set to ERROR. Also, if the file which was to be replaced has the same name as the upload file, the new file should be created with +1 seqno in its name , ie: name \_+1. This will be a flag to the data manager to resolve the issue if need be.

7. HIE browse list: the ID and file id fields should be a avaialble search criteria.

8. Search API  
a) ID should be a search criteria.  
b) The API search should return the ID and file id for a particular file in JSON.

=> So, if a researcher wants to obtain a file by ID, then the steps are to do an API search via the ID field and then download the file via its file-id.

9. When creating a package, as the file id is a transient handle, the anchor back to the HIEv in the README.HTML should be removed as it is likely to go out of date. THe README.HTML should include the ID of contained files if it has been added.

Scenario:  
---------

1. User publishes a package in Dc21 -> RIF-CS available  
2. RDC harvests the RIS-CS -> work item enters RDC workflow  
...  
3. RDC calls UWS handle server to create Collection Record (and other) handle(s).  
4. RDC sends e-mail acknowledgement to HIE that the package with details ... which was published by dc21 on ... has been recorded in the RDC and has handle ...

5. Technical ops reads emails and updates Dc21 package with RDC Collection Record Handle.

Notes:

Step 4 has not been implemented and a request would need to be made of the Library to do this, and it would need to be fitted into their work plan.

Step 5 is a manual step, which could be automated, but probably not the bandwidth to do it in sprint 8.

This doesn't fix the versioning issue we've spoken about, but it is a first step.

Acceptance Criteria

TOA5 files are not replaced if they have their ID field set.

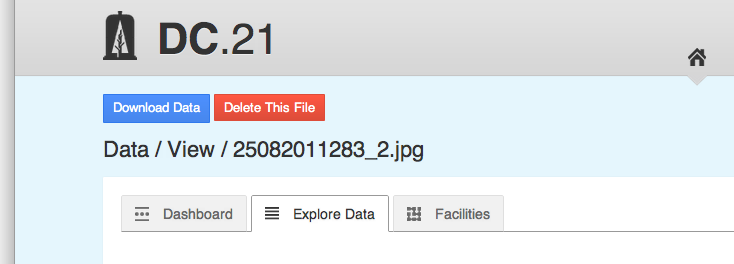
Packages can be deleted regardless of whether they are published, but not if they have an ID field set

It should be possible for a user to ermove the ID field from a file or package.

* 1. Deleting a file

In order to delete a file from the HIEv database, first view the metadata for that file by clicking on the file name in either the Dashboard or Explore Data views. If you have permission to delete that file, a red Delete This File button will appear at the top of the Metadata View screen.

Note You will only have permission to delete this file if you are logged in with the user credentials which were used when the file was uploaded, or you have Administration permission.



When you click on the Delete This File button, you will be asked to confirm that you do wish to delete the file.

Deleting files removes the file completely from the HIEv system. It is irreversible.

%%% Files can only be deleted one at a time by using the "Delete This File" button at the top of the page displaying the file's metadata. Users can only delete their own files. Administrators can delete all files.

1. Publishing Your Data

Sprint 7

 When a package is created, a RIF-CS collection record is generated but kept internally to the app.

 A user can delete a file of type Package as long as it hasn't been published.

 When the package is published, the RIF-CS is made available to OAI-PMH harvesters.

 Once a Package has been published, the Delete This File and Edit Metadata options are no longer available to the user on the Data / View form.

Sprint 8 – DC21-450

\* On the Data / View form, if a user presses the Publish button for a Package, the warning message should be changed to simply say "Do you really want to publish this Package?" without the additional implications.

\* Once a package is published, the "Delete This" and "Edit metadata" buttons should remain available and functional.

If any of the metadata is changed, the user should be warned via a dialog box that the changes will not affect the unerlying RIF-CS. Message: "These changes will not affect the underlying RIF-CS. To modify the metadata in the RIF-CS, you will need to create a new package."

\* If the published package is deleted, any RIF-CS records associated with the package should no longer be available to OAI-PMH harvesters. These OAI-PMH records should be moved to an archive folder on the HIEv server so that they can be recovered along with the package zip file if ever needed. Please discuss with Georgina the best place for these files.

Acceptance criteria:

- A simpler warning message on publish

- Info message if Package metadata is changed.

- User can still delete package files after publish and change their metadata

- RIF-CS records no longer avaialable for harvest after a published package file is deleted.

- RIF-CS and published package file is stored on a server archive if the package is deleted.

%%% DC21-305 – Sprint 8

See attached README.HTML file with RDFA lite attributes added to it.  
There are some limited implementation notes recorded as comments in the file.

The current haml template should be modified to adhere to this new HTML design. Please check with Peter B if any of it is unclear.

To view the attributes, use: www.rdfa.play/info.

Acceptance Criteria:

* the RDFA lite attributes appear correctly in the README.HTML file
* the attributes tree displays as per the sample README when run through www.rdfa.play/info.

%%% Add comments about harvesting, OAI-PMH, etc...

Once an experiment is complete, the data collected can be published to the [Australian Research Data Commons](http://www.ands.org.au/about/approach.html#ardc).

Before publishing, the data files to be published must be combined together into a single Package file. Package files are ZIP files which use the Bagit format, which is described in Appendix %%%.

Once created, Package files are shown in the file list which is displayed on the Explore Data tab.

Exactly which files are included in a Package intended for publishing is largely dependent on what is meaningful for the data and research discipline in question. It is entirely valid to have the same data appear in multiple Package files if that will aid discovery and reuse for other researchers. For example, a large set of data could be divided into two smaller, but overlapping, sets of data that represent different lenses (research problems) that the data could be seen through.

Once a Package has been published the metadata describing the Package will be made available for harvesting by the UWS Metadata Store. After this has occurred and the Package has been approved, it will become discoverable in [Research Data Australia](http://researchdata.ands.org.au/).

* 1. Creating a Package

Creating a Package creates two related components:

|  |  |
| --- | --- |
| ZIP Bagit File | This ZIP file contains a snapshot of all the data for this Package. This includes copies of the data files, a Readme.HTML file which contains a copy of the all of the data files’ metadata. |
| Matching RIF-CS File | This file contains a copy of the Package metadata which is entered at the time the Package is created. |

To create a Package containing one or more files:

* Add the required file of files to your Cart, ensuring that the Cart contains only those files you wish to include in your Package. See section %%% for instructions on using the HIEv Cart.
* Click on the Cart status box to open the Cart dropdown menu.
* Select Package from the dropdown menu. Alternatively, selected Edit Cart from the dropdown menu and then click on the Package button on that screen, as shown below. If this method is used, the Cart can be reviewed prior to Packaging.
* The New Package screen will be displayed. Enter the metadata associated with your Package. See below for details.
* Click on Save to cause your Package file to be created and saved. The Package file can now be viewed in the Explore Data tab. If you click on Back, you will be returned to the Explore Data tab and the Package will not be created.

%%% New Package screen dump.

The New Package screen shows the following fields:

|  |  |
| --- | --- |
| Filename | The name you require for the Package file. |
| Experiment | Select the experiment to which this Package relates from the dropdown list. |
| Description | Enter a description which is appropriate for Publishing on the Australian Data Commons. %%% |
| Tags | Check any tags which are relevant to this Package. |
| Start Time End Time | Enter the earliest and latest times of the observations included in this Package. |

Note It is important to check this metadata closely before Clicking %%% button name. This metadata is copied into the RIF-CS file, which cannot be edited. If it is wrong, the Package must be deleted and re-created.

It is possible to create a Package which contains other Packages. There may be circumstances when there is a requirement to Publish a Package which includes of one or more other Packages.

* 1. Editing a Package’s metadata

Note Editing most fields of a Package’s metadata is not recommended. Please see section for information about what metadata can be meaningfully modified.

To edit a Package’s metadata:

* Navigate to the Package file using the Explore Data tab.
* Click on the Package file’s filename to open the metadata edit screen.

%%% Edit Package Metadata screen dump

* Edit the metadata in the same manner as the metadata of any file is edited. Refer to the Field table in the previous section Creating a Package and also in section 4.4Viewing and Editing a File's Metadatato see the meaning of the fields. Not all metadata fields can be modified.
  1. Publishing a Package

When a Package has been created and its metadata is correct, it can be Published.

To Publish a Package:

* View the Package’s metadata by clicking on its filename on either the Dashboard or Explore Data tab. Review its metadata to ensure you have selected the correct Package file and that it is ready to Publish.

%%% Metadata view screen dump with Publish button highlighted.

* Click on the Publish action button at the top of the metadata screen. This Publish button will not appear if the Package has already been Published.
* A dialog box is shown for you to indicate that you are sure you wish to proceed. Click on OK. The Package metadata will be copied for harvesting by the %%%. If you click on Cancel button, you will returned to the Package metadata screen.

%%% Describe action – Places RIF-CS file into harvestable location.

* 1. Managing Published Packages

%%% Delete and edit are/will be available from Sprint 8 – explain how they interact with the published data.

%%% Deleting before or after harvest is different – and you can’t know when harvesting happens!

%%% Editing is rearranging Titanic’s deck chairs.

%%% Setting the ID is a good thing, once you know the external ID created by harvesting.

It is possible to edit the metadata or delete any Package file that you have created. If you have administration privileges, you can delete any Package file. %%% Except those with IDs set. You need to remove the ID first.

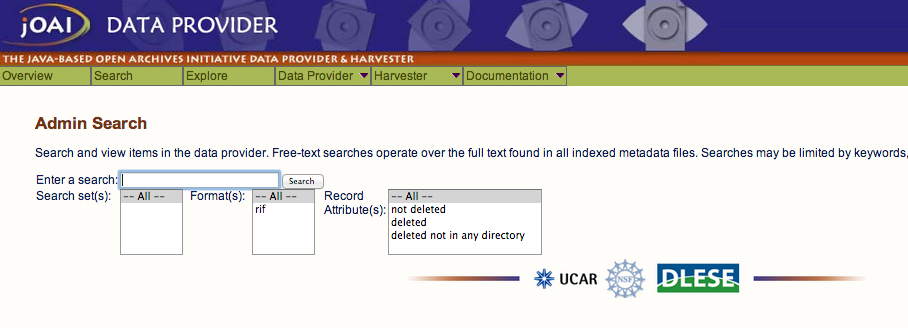
If you edit the metadata of a Published Package, the Published version of that data is not changed. Only the metadata stored within the HIEv system is modified. In order to modify the published metadata, you must create a new Package with correct metadata and Publish that new Package.

If you delete a Published Package, you are deleting the data from within the HIEv system. However, that data is %%%.

* 1. Viewing Published data

%%% PeterB suggests it may be better to remove this section, although there’s presently no other way. There is a chance of a RIF-CS view option in a later sprint. All HIEv applications must use jOAI if they implement Publishing.

The descriptions of published Packages can be viewed by going to the jOAI web interface at **http://<***your.HIEv.sever>***/oai/admin/query.do** and performing a search:



Clicking the search button with the search field blank will show all published Packages:



1. Downloading files

HIEv allows you to download any data file, or multiple data files, to your local computer.

If you download a single file, it will be saved on your computer in its usual format. If you simultaneously download more than one file, the files will be combined into a ZIP file and that ZIP file will be downloaded to your computer. (This ZIP file is not a Bagit format ZIP file.)

To download one or more files:

* Add those files to your Cart, ensuring that the Cart contains only those files you wish to download. See section %%%.
* Click on the Cart status box to open the Cart dropdown menu.
* Select Download from the dropdown menu. Alternatively, select Edit Cart from the dropdown menu and then click on the Package button on that screen, as shown below. If this method is used, the Cart can be reviewed prior to Packaging.
* A file dialog box will open. Navigate to the sub-directory into which you wish to save the downloaded data and select the name you wish to use for the downloaded data file. The file dialog will be for the one data file if only one file is in your Cart, or it will be for one ZIP file if multiple files are in your Cart.

When downloading data files, only the data files themselves are downloaded. Metadata is not downloaded.

If you download a Packaged ZIP file, you can access the metadata for each of the files, which is included in that Packaged ZIP file. See Chapter %%% for instructions on creating a Packaged ZIP file and Appendix %%% for details of the Bagit format, which is used for Packaged ZIP files.

1. Facilities and Experiments

All data files uploaded to the HIEv system must be associated with an Experiment. In turn, all Experiments are associated with a Facility.

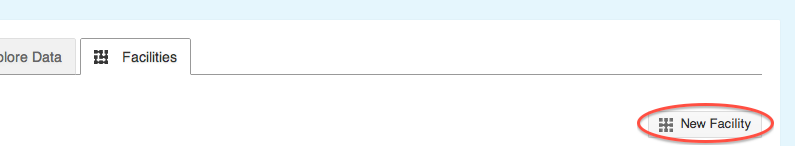
Therefore entries for Facilities and Experiments must be created before the associated data files are uploaded.

* 1. Creating and Editing Facilities

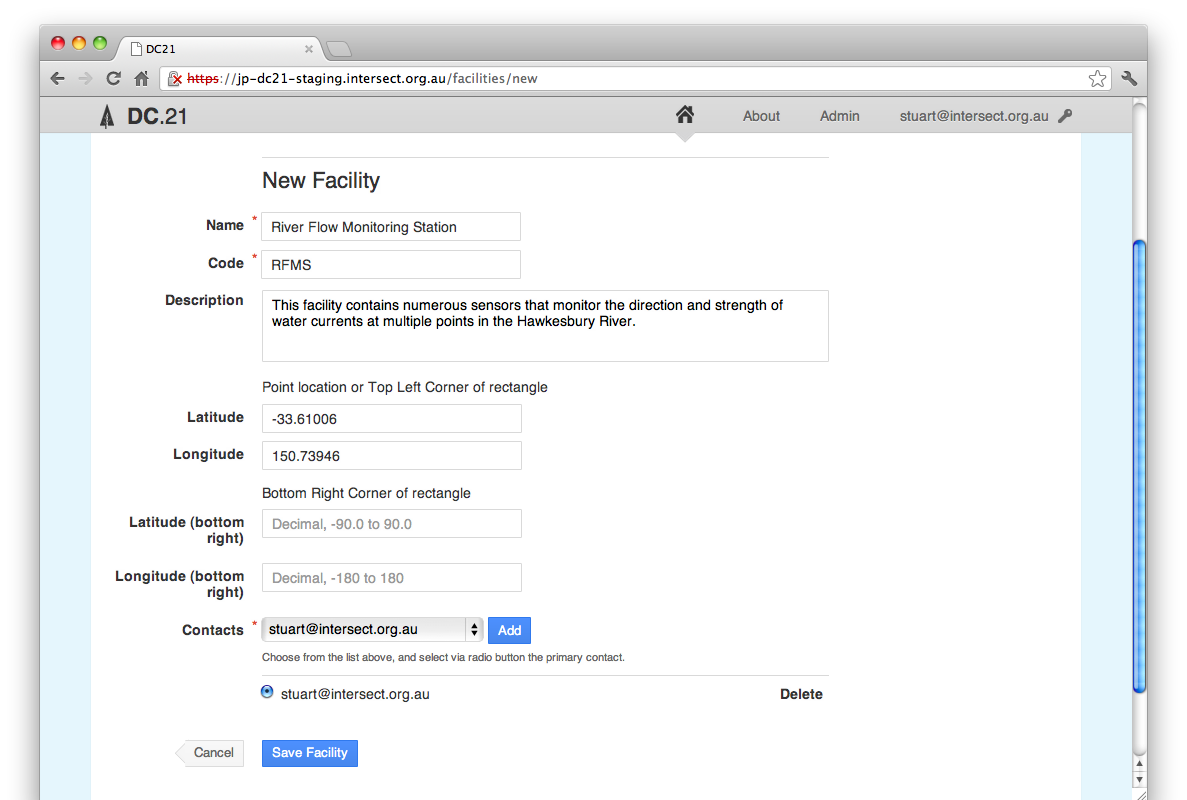
A Facility represents any instrument, or discrete set of instruments that are used in concert. This could be a multi-million dollar fixed facility with hundreds of instruments and sensors or a single piece of portable equipment that is taken out into the field.

Note Take care. Once created, a Facility entry cannot be deleted. This prevents data files which reference this Facility from becoming invalid.

Facilities are added by clicking the "New Facility" button at the top-right of the Facilities tab:



This will display a form that allows you to add details about the facility you would like to create:



The **Name** for the facility is a short, plain-English title that will be used in the application interface to refer to the facility.

The **Code** for the facility is a short unique string.

The **Description** of the facility should be as comprehensive as possible describing details that would help a researcher both discover the facility when searching and assist the researcher in being able to interpret the data that is produced by the facility. These details would include things such as:

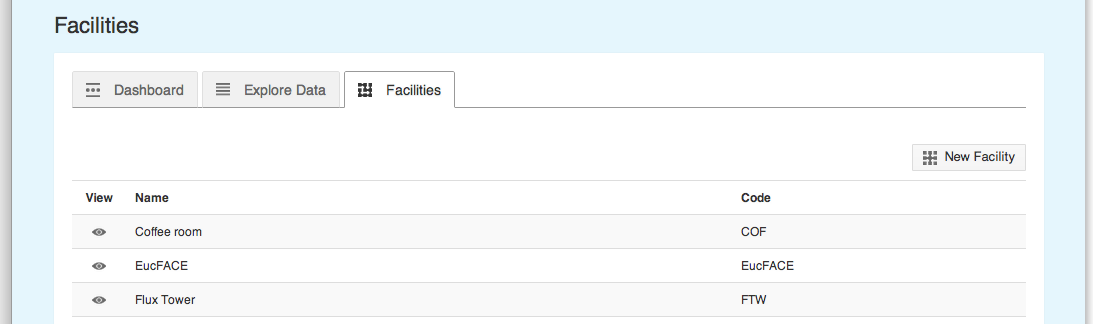
* The purpose of the facility
* Types of sensors installed at the facility
* Location of the sensors within the facility

The **Latitude** and **Longitude** for the facility are expressed in Decimal Degrees (<http://en.wikipedia.org/wiki/Decimal_degrees>) and can be taken directly from Google Maps. If a single set of co-ordinates are given that is considered to be the central point for the facility. If two sets of co-ordinates are given they are considered to be a rectangle that bounds the facility.

The **Contacts** for the facility must be selected from the users registered within the HIEv system. There must be at least one primary contact for each facility.

%%% The Contacts Add button needs explanation, and also the scope of the Contacts List.

Once facilities have been created they will appear on the Facilities tab:

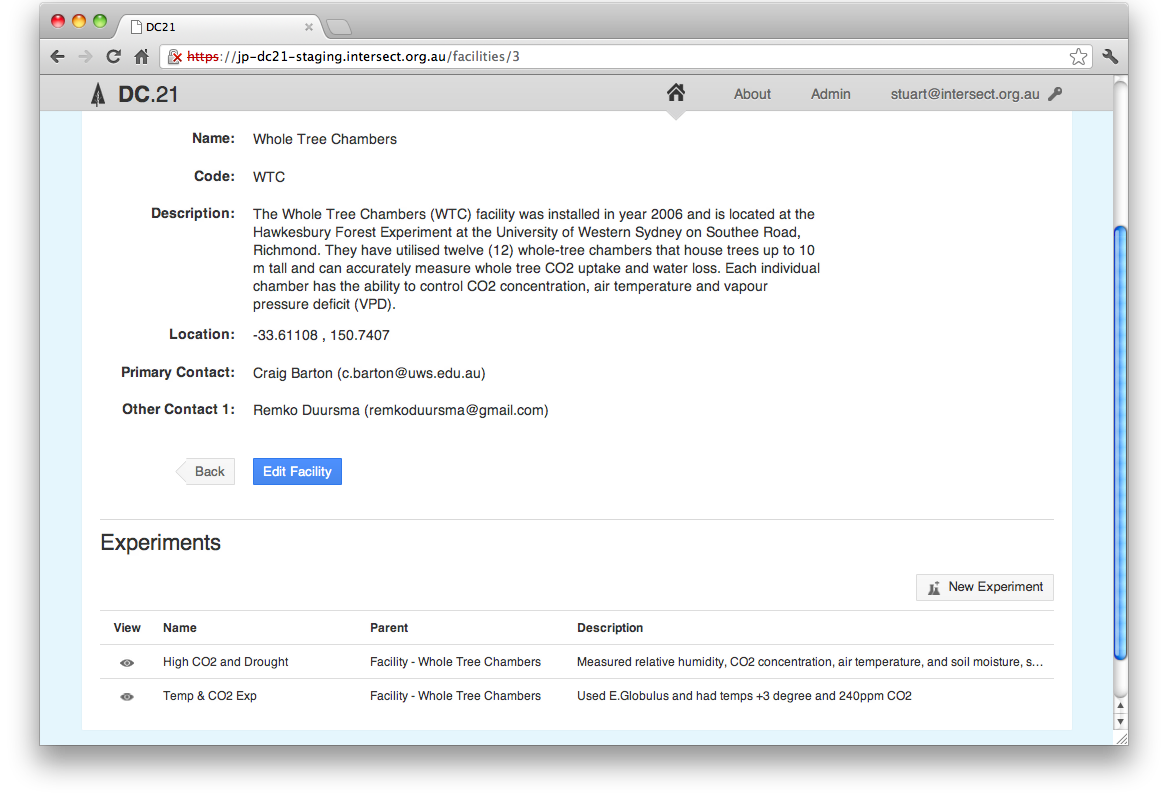


* 1. Creating and Editing Experiments

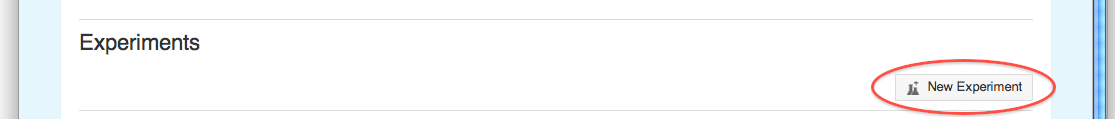
Before data can be uploaded into a HIEv system at least one Facility and Experiment must be defined. To define an Experiment, start by selecting the Facility where it will be running.

Note Take care. Once created, an Experiment entry cannot be deleted. This prevents data files which reference this Experiment from becoming invalid.

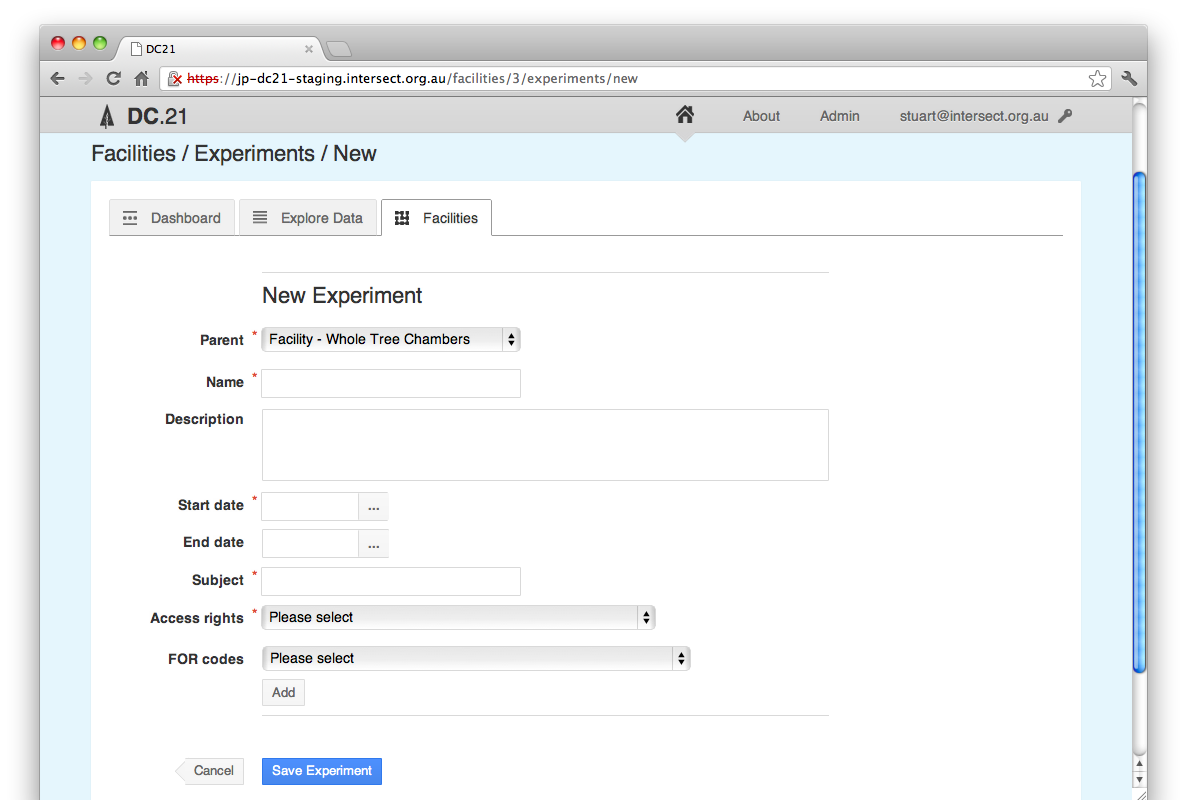
This is done from the Facilities tab by clicking the "Eye" icon in the left-hand View column beside the desired facility. This will display the information page for the selected facility:



When viewing a facility, any experiments defined for that facility are listed below it. At the top-right of the experiment list is a "New Experiment" button.



Clicking this button will display a form prompting you for all the information needed to create a new experiment under the current facility.



The **Parent** for an experiment is either the current facility, or another experiment running at that facility. If an experiment is selected, the new experiment is considered a sub-experiment of the one selected.

The **Name** for the experiment should be short, but descriptive enough to uniquely identify the experiment, including distinguishing an experiment from those that are likely to come in the future.

The **Description** for the experiment should describe the purpose of the experiment and the techniques employed. Particular focus should be given to aspects of the experiment that produce data that is stored in this system.

The **Start date** for the experiment is the date that experiment was first considered to be active.

The **End date** for the experiment is the date that the experiment concluded. This field should be left blank for experiments that are currently active.

The **Subject** for the experiment is a short phrase describing the experiment's main research area. The Subject is primary recorded to support publication to [ANDS](http://www.ands.org.au/guides/cpguide/cpgsubject.html) and in their own words, “A subject is a term, keyword, classification code or phrase representing the primary topic or topics covered by a registry object.”

The **Access rights** drop down list box provides a selection of licenses to release the data from this experiment under. It is preferred in Australia that data is released under a [Creative Commons](http://creativecommons.org.au/learn-more/licences) license.

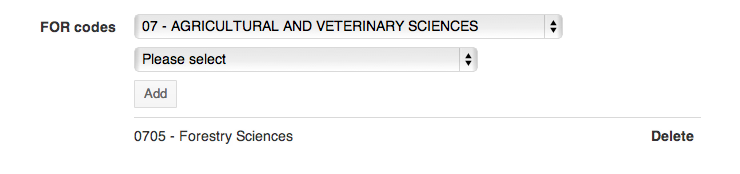
Each experiment can have one or more **FOR codes**. The Fields of Research is a hierarchical classification with three levels, namely Divisions (two digits), Groups (four digits) and Fields (six digits). A unique number identifies each level.

Each Division is based on a broad discipline. Groups within each Division share the same broad methodology, techniques and/or perspective as others in the Division. Each Group is a collection of related Fields of research. Groups and Fields of research are categorised to the Divisions sharing the same methodology rather than the Division they support.

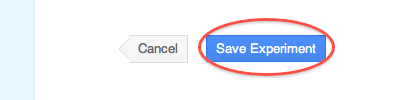
Codes are selected two digits at a time using the following interface. Codes can be specified to two, four or six digits but selecting options from the drop-down list boxes and clicking "Add".

Once an FOR code has been added it will appear below the list boxes and more FOR codes can be added.

FOR codes that have been added can be deleted by clicking the "Delete" button to the right of the code.



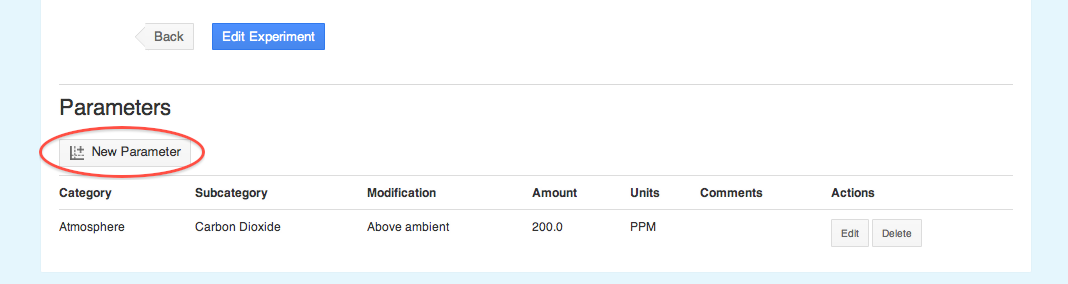
When you have finished adding all the required FOR codes, click "Save Experiment" at the bottom of the page:



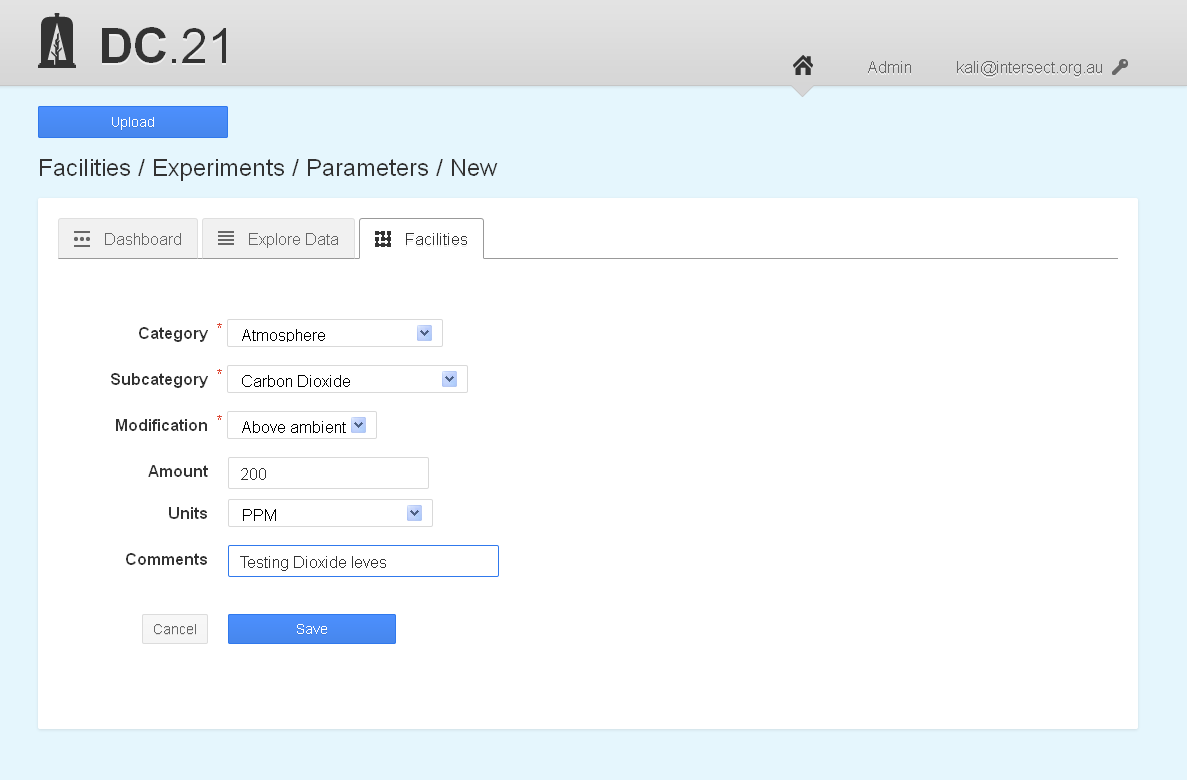
To abort creating the experiment, click "Cancel".

* 1. Setting Up Experiment Parameters

Experiments can optionally have one or more experiment parameters. These parameters provide a structured way to describe experimental treatments such as raising the CO2 within the tree chambers. Parameters are added by clicking the "New Parameter" button directly below the experiment:



This button will display the form below:



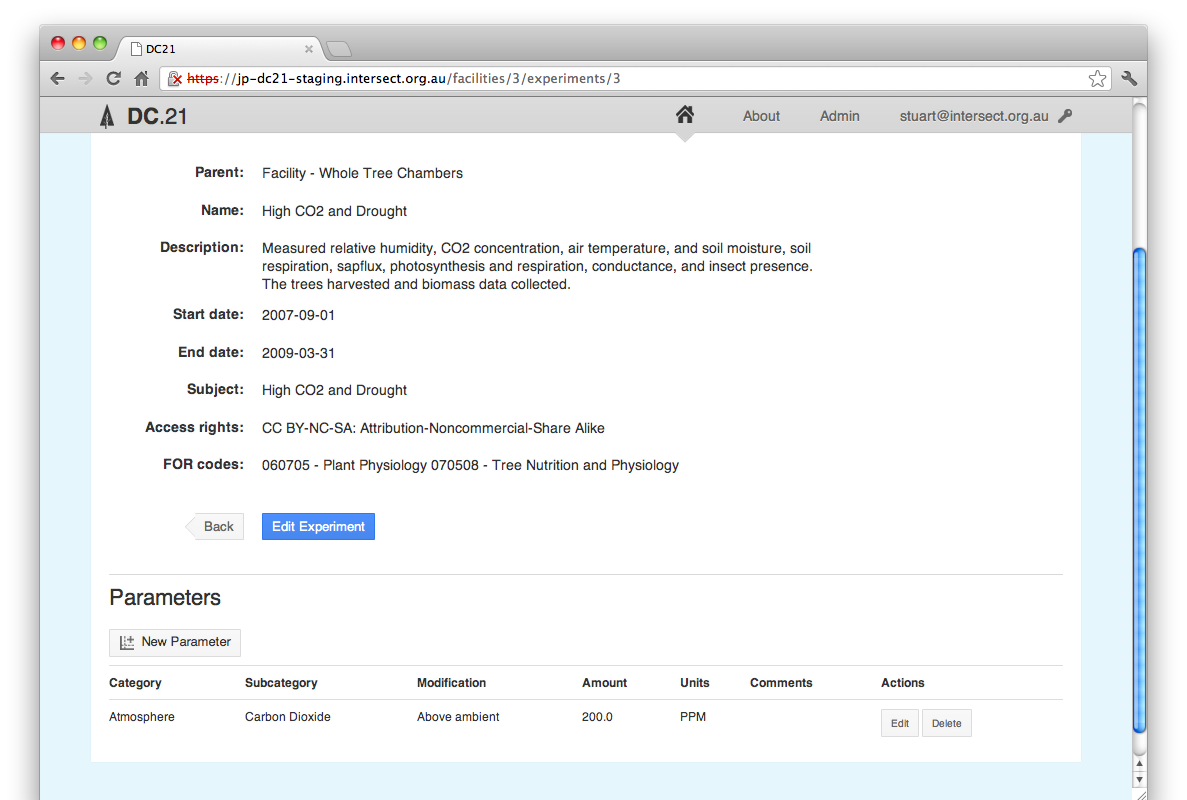
The first three fields are mandatory. The **Category** and **Subcategory** fields allow you to specify the medium that is being modified. The system administrator configures the values available in these dropdown list boxes.

The **Modification** indicates the general way in which the medium has been modified. The optional **Amount** and **Units** fields allow more specific information to be recorded about modification.

The **Comments** field can be used to record any unstructured, plain-text information you would like to record about the treatment

To finish, click the blue **Save** button at the bottom of the form.

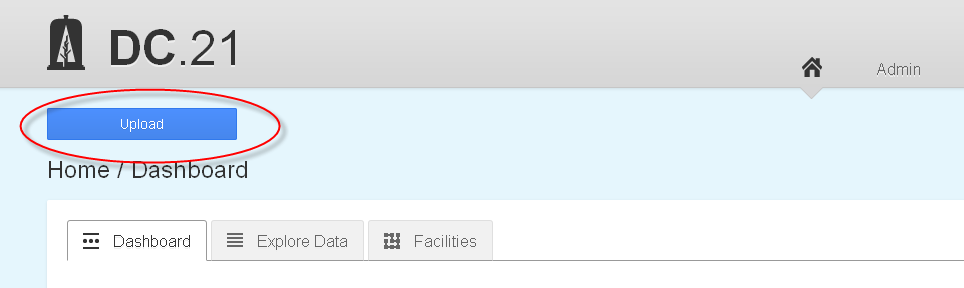
Once an experiment parameter has been created it will appear below the main description of the experiment:



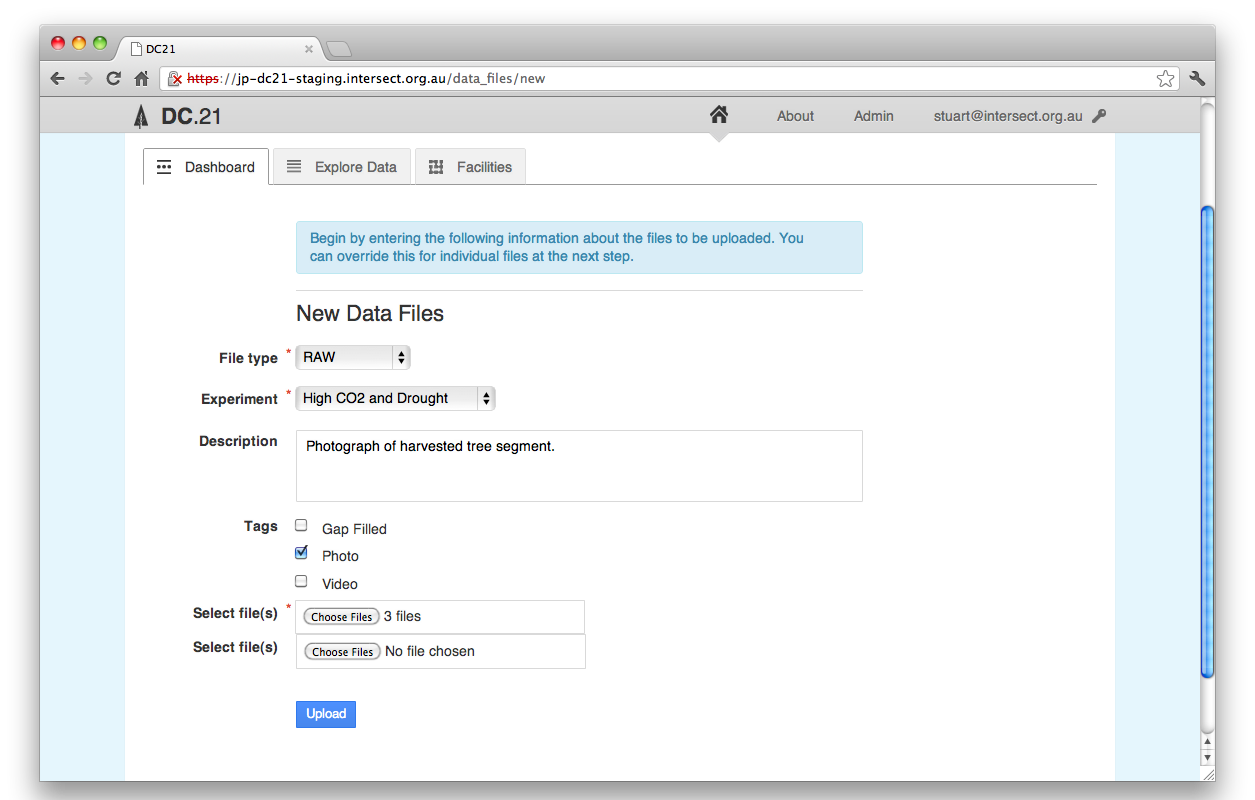
An experiment can have multiple parameters or none at all. Existing experiment parameters can be edited or deleted using the appropriate button to the right of the parameter in the **Actions** column.

1. Uploading Data files

New files are added to the system using the blue **Upload** button at the top left of the screen:



Clicking this button will take you to the "New Data Files" form:



On this form you will enter all the metadata for the new file or files and also select the files to be uploaded.

The **File Type** is chosen from a set of fundamental types of data that has been defined by the system administrator. These are aimed at helping track data through its various stages of processing.

Note Files that are uploaded with a File Type of **RAW** and are also detected as containing valid TOA5 header information are treated as a special case. See the end of this section for more information.

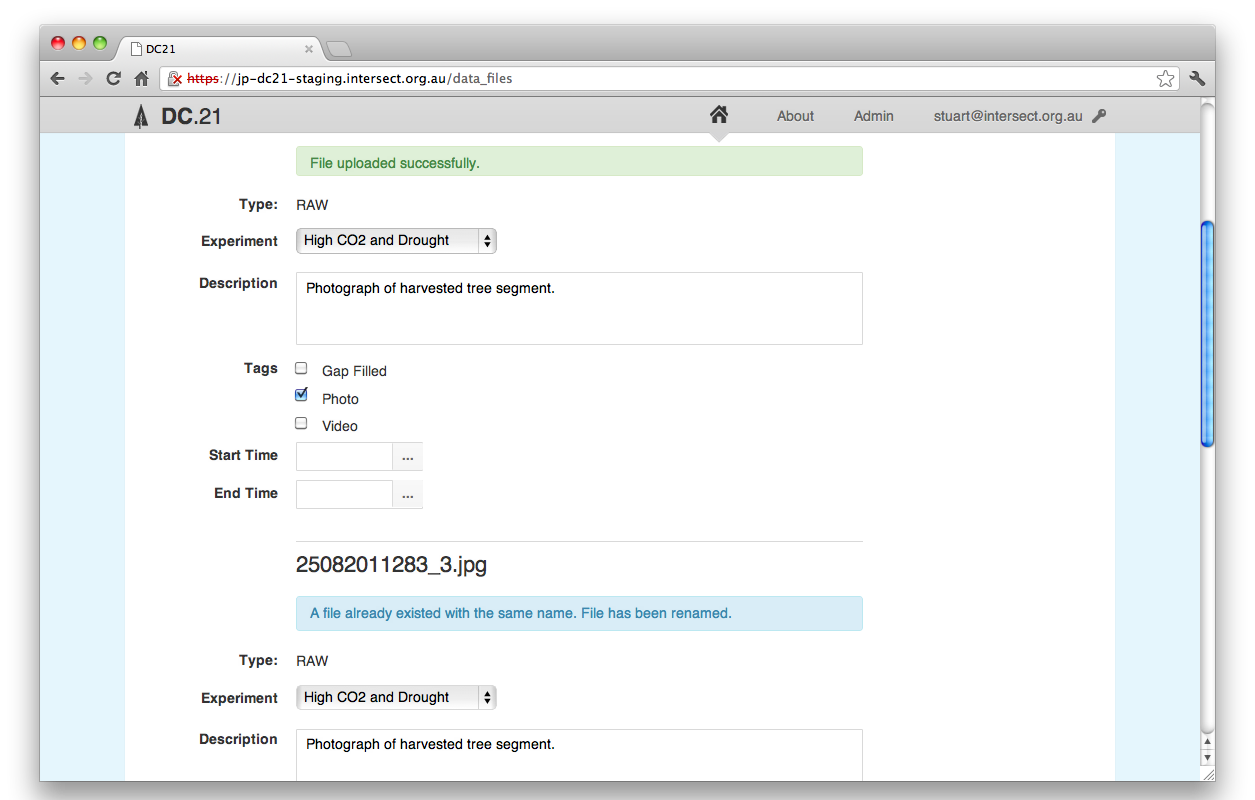
The **Experiment** for the file indicates which experiment produced the file. This is the primary mechanism by which files are grouped and associated with each other.

The **Description** for the file should contain enough information for others to understand the data within it. This will vary widely depending on the type of file but would typically contain information on the variables collected. Note that TOA5 files will have this variable information extracted automatically. %%% Is data entered here for TOA5 files ignored?

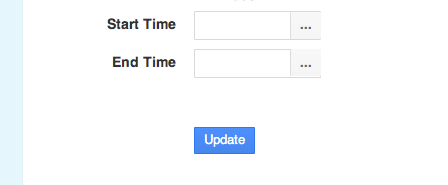
The **Tags** for the file are also chosen from a set defined by the system administrator. Whereas a file must have only one File Type, it may have any number of the available tags.

Once all the metadata has been entered and the files have been selected, click the blue Upload button at the bottom of the form.

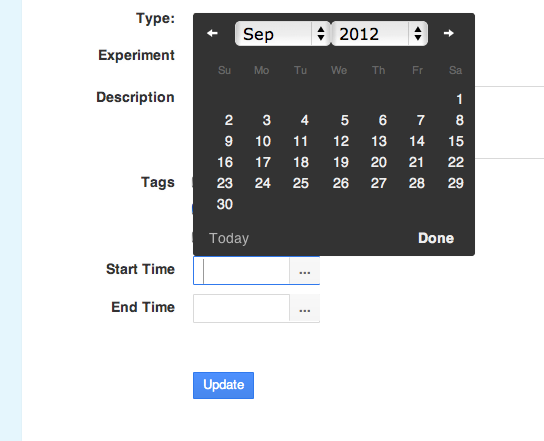
After the file or files have successfully uploaded, the supplied metadata will be applied to all uploaded files (%%% except descriptions to TOA5 files?) and you will be presented with a screen on which you can edit all of the uploaded files’ metadata individually. This is useful, for example, when you wish to give ten files the same description but add an extra tag to one of the files. %%% Does it permit you to change the descriptions on TOA5 files?



If the start and end dates for the data cannot be automatically extracted for the file (such as with TOA5 files %%% Does this mean it can or can’t?), the above screen presets you with the opportunity to enter this information manually:



Dates can be typed into the **Start Time** and **End Time** fields in *YYYY-MM-DD* format or by clicking on the ellipsis to the right and selecting a day from the calendar that is displayed:



If an uploaded file has the same filename as another file that already exists within the system, HIEv will automatically suffix a unique number onto the end of the original filename, just before the file extension.

* 1. Uploading RAW TOA5 data files

When a TOA5 CSV file is uploaded with the **Type** of RAW, it is considered to become part of the canonical stream of data for that data logger. As a result, there will only ever be a single file with a **Type** of RAW that contains any given sample from a TOA5 data logger.

This has the affect that:

1. If TOA5 file is uploaded with a **Type** of RAW, and the file being uploaded is a complete superset of another file (or files) that are also RAW TOA5 files from the same data logger, the subset files will be replaced with this new file, regardless of the file names.
2. If a TOA5 file is uploaded with a **Type** of RAW that only partially overlaps an existing file of RAW data from the same data logger, the file will be uploaded, but its **Type** changed to ERROR and the original file(s) left in place.
3. If a TOA5 file is uploaded with a **Type** of RAW that overlaps an existing file of RAW data, but does not pass a sample-by-sample comparison with the original file(s), the file will be uploaded, but its **Type** changed to ERROR and the original file(s) left in place.
   1. Automating the upload of data to HIEv

As well as the web interface, data can be uploaded to the HIEv system using an HTTP-based API. The upload of data into the system is facilitated through a Ruby script. Instructions and a download for using this script can be found at <https://github.com/IntersectAustralia/dc21/wiki/Setting-Up-Automated-Load-From-PC> %%% Check this URL

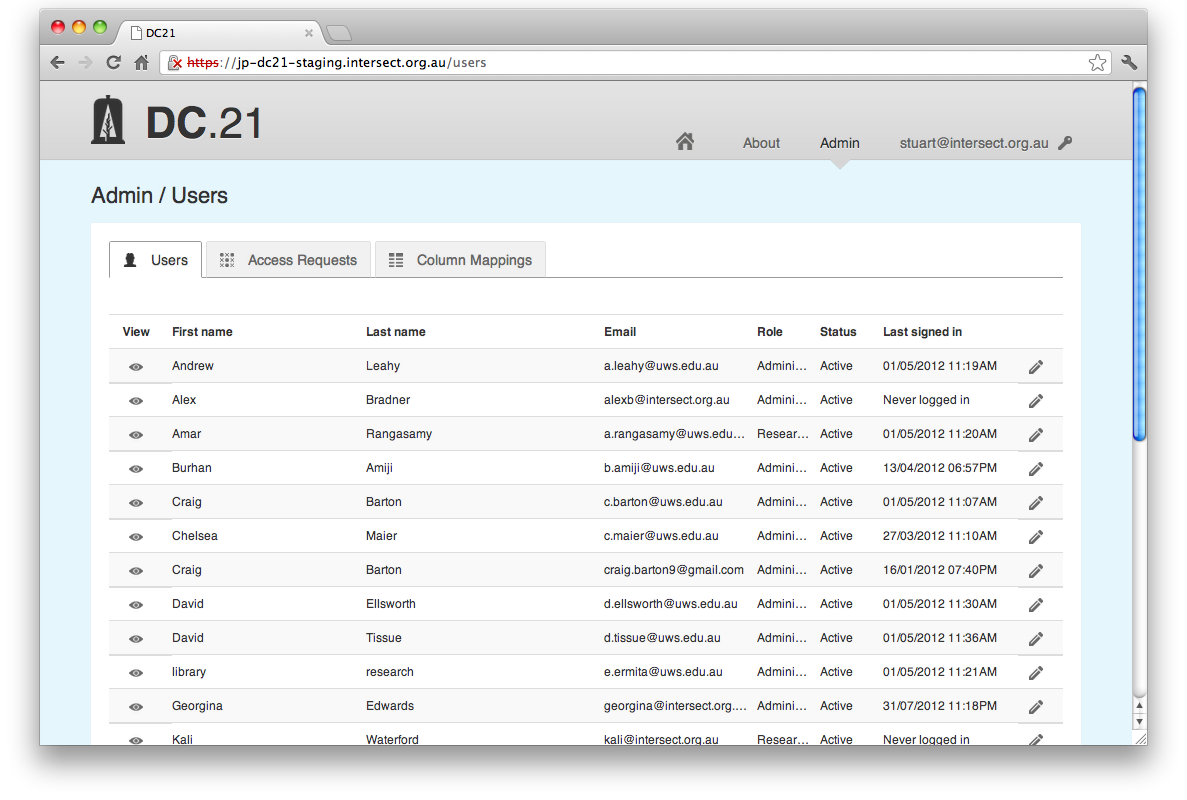
1. System Administration

When a user is created they are given a role within the HIEv system. This role dictates what permissions the user has within the system. The most powerful role a user can be given within the system is that of the System Administrator.

All System Administrators have access to the Admin section accessed from the **Admin** link at the top right of the screen (only visible for this role). The Admin section has three tabs: Users, Access Requests and Column Mappings.

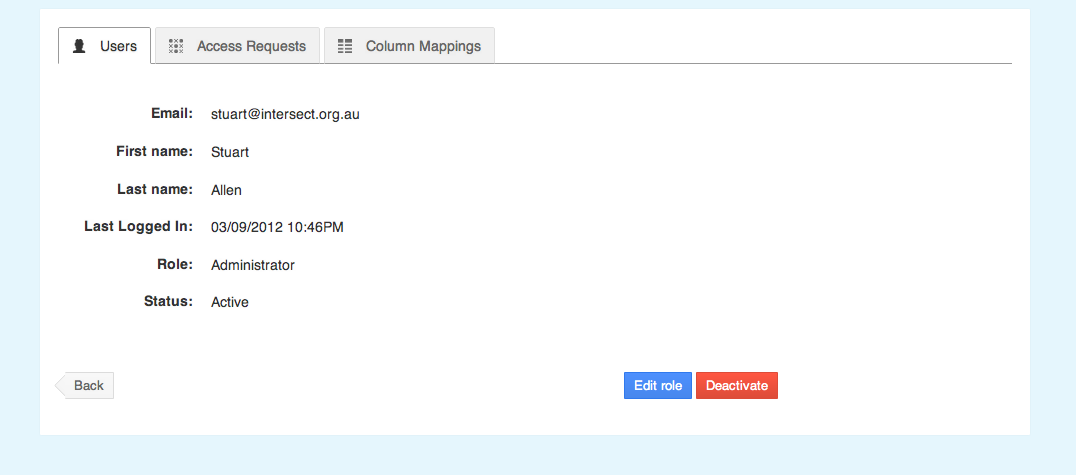
* 1. The Users Tab

The **Users** tab lists all the users that are registered within the system:



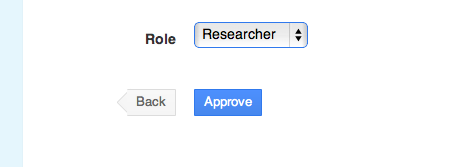
%%% The “Eye” is removed, but functionality remains essentially unchanged.

Clicking the "Eye" icon to the left of any user will display his or her details in full:



Clicking the **Deactivate** button will disable the account from being used to login to the system. No data uploaded by the user will be deleted.

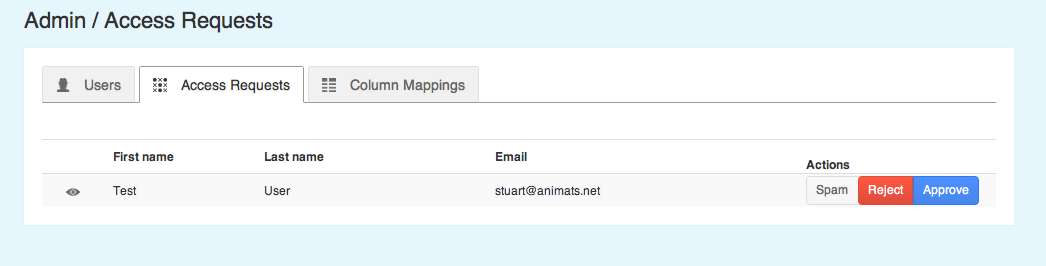
Clicking the **Edit role** button allows the administrator to change the role that has been assigned to the user being viewed:



Any given user can only have a single role.

* 1. The Access Requests tab

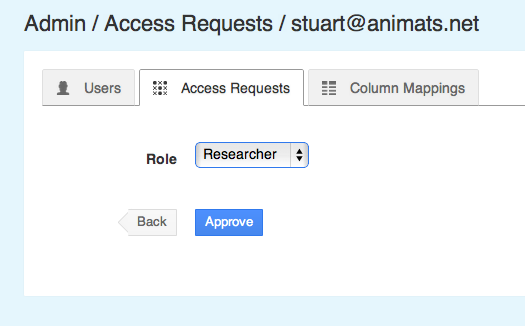
The **Access Requests** tab is where an administrator can approve or deny requests for a user account in the system:



Selecting **Spam** will simply delete the account request from the system.

Selecting **Reject** will inform the user that his or her request for an account has been rejected.

Selecting **Approve** with take you to a screen where you must select a role for the user in the system:

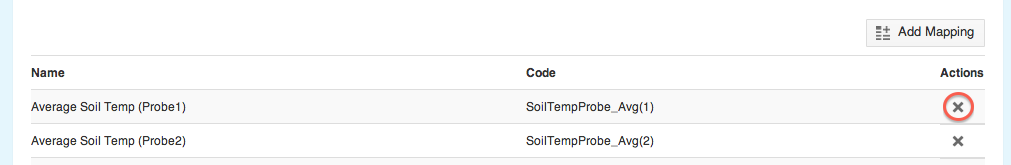


Once an account type has been selected and the account approved, the user will receive an email notifying them that they may now log into the system.

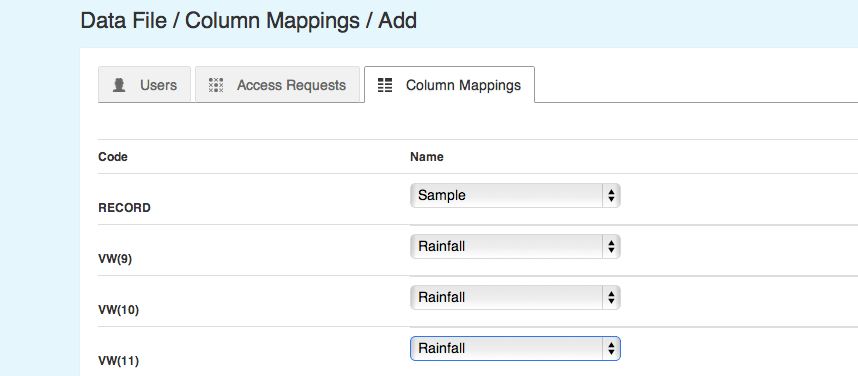
* 1. The Column Mappings tab

Column mappings are a way of defining a relationship between the column headers in TOA5 files (the "Code" part of the mapping) to a standard name from a defined ontology (the "Name" part of the mapping.) The **Column Mappings tab** allows the administrator to add and delete column mappings.

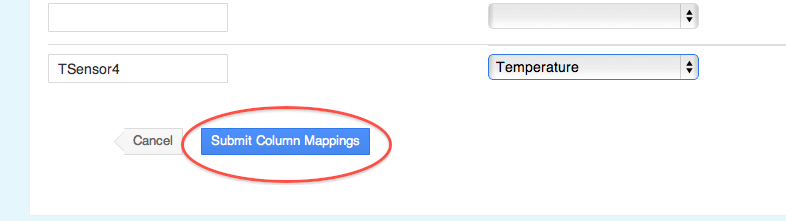
To delete an existing mapping, simply click the cross in the far right **Actions** row of the table for the mapping you wish to delete:



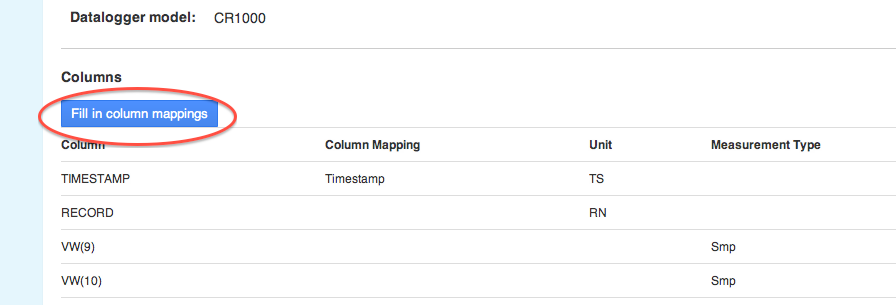
To add more mappings click the **Add Mappings** button at the top right of the tab. This will display a form where the mapping pairs can be defined. On the left of each row the code from the TOA5 column header can be entered and on the right the standard name to map to can be selected from a drop-down list box:



Once the mappings are defined click the **Submit Column Mappings** button:



Column mappings may also be defined using the **Fill In Column Mappings** button on the **View Metadata** page for any given TOA5 file. This is the preferred method as it avoids the need to manually type the column headers into the "Code" fields:



1. Modifying Tags, Column Mappings and Experiment Parameters

When HIEv is first installed, the available Tags, Column Mappings and Experiment Parameters are populated in the database from the configuration file: dc21app\_extra\_config.yml %%% Check file name. Once the system has been installed, the experiment parameters and tags can be modified by modifying the lookup tables that store these values. This can be done directly using a tool like PSQL. Another convenient way to do this is to use the Rails console. The instructions below use the Rails console to add rows to the tables.

If you wish to delete or modify existing rows in these tables, make sure you maintain referential integrity with existing records.

To begin, ssh to the server the HIEv system is running on. Once connected, the first step is to determine the Rails Environment the system is running as. This will generally be “production”, but you can check by looking for a RailsEnv line in the Apache Rails configuration (eg. <root>/etc/httpd/conf.d/rails\_dc21app.conf %%% Check this file name). This will look something like the following:

RailsEnv production

In this case, the Rails Environment is "production". Once the value of this setting has been determined for you system, navigate to the location of the application installation (e.g <root>/home/devel/dc21app/current %%% Check this file name) and enter the directory "dc21app/current" %%% Check this file name. From here you can start the Rails Console using the command:

RAILS\_ENV=<RailsEnv> bundle exec rails console

e.g.

RAILS\_ENV= production bundle exec rails console

This will give you a prompt similar to:

Loading production environment (Rails 3.1.1)

1.9.2p290 :001 >

From this prompt you can issue commands to add Tags and Experiment Parameters.

To add a Tag use the command:

Tag.create!(name: '<Tag name>')

eg.

Tag.create!(name: 'Analysed')

This will result in output similar to:

(0.3ms) BEGIN

(1.3ms) SELECT 1 FROM "tags" WHERE LOWER("tags"."name") = LOWER('Analysed') LIMIT 1

SQL (8.7ms) INSERT INTO "tags" ("created\_at", "name", "updated\_at") VALUES ($1, $2, $3) RETURNING "id" [["created\_at", Fri, 14 Sep 2012 10:55:24 EST +10:00], ["name", "Analysed"], ["updated\_at", Fri, 14 Sep 2012 10:55:24 EST +10:00]]

(0.6ms) COMMIT

=> #<Tag id: 6, name: "Analysed", created\_at: "2012-09-14 00:55:24", updated\_at: "2012-09-14 00:55:24">

To add a Modification or a Unit for an Experiment Parameter, use the commands:

ParameterModification.create!(name: 'Above average')

ParameterUnit.create!(name: 'PSI')

Parameter Categories and Sub Categories require an extra step to define the relationship between the two:

parameter\_category = ParameterCategory.create(name: 'Light')

parameter\_category.parameter\_sub\_categories <<

ParameterSubCategory.create(name: 'Brightness')

This will result in output similar to:

1.9.2p290 :001 > parameter\_category = ParameterCategory.create(name: 'Light')

(0.4ms) BEGIN

SQL (121.5ms) INSERT INTO "parameter\_categories" ("created\_at", "name", "updated\_at") VALUES ($1, $2, $3) RETURNING "id" [["created\_at", Fri, 14 Sep 2012 16:14:26 EST +10:00], ["name", "Light"], ["updated\_at", Fri, 14 Sep 2012 16:14:26 EST +10:00]]

(0.5ms) COMMIT

=> #<ParameterCategory id: 8, name: "Light", created\_at: "2012-09-14 06:14:26", updated\_at: "2012-09-14 06:14:26">

1.9.2p290 :002 > parameter\_category.parameter\_sub\_categories <<

1.9.2p290 :003 > ParameterSubCategory.create(name: 'Brightness')

(0.3ms) BEGIN

(0.3ms) ROLLBACK

(0.2ms) BEGIN

ParameterCategory Load (0.7ms) SELECT "parameter\_categories".\* FROM "parameter\_categories" WHERE "parameter\_categories"."id" = 8 LIMIT 1

SQL (1.2ms) INSERT INTO "parameter\_sub\_categories" ("created\_at", "name", "parameter\_category\_id", "updated\_at") VALUES ($1, $2, $3, $4) RETURNING "id" [["created\_at", Fri, 14 Sep 2012 16:14:27 EST +10:00], ["name", "Brightness"], ["parameter\_category\_id", 8], ["updated\_at", Fri, 14 Sep 2012 16:14:27 EST +10:00]]

(0.5ms) COMMIT

ParameterSubCategory Load (0.7ms) SELECT "parameter\_sub\_categories".\* FROM "parameter\_sub\_categories" WHERE "parameter\_sub\_categories"."parameter\_category\_id" = 8 ORDER BY "parameter\_sub\_categories"."name"

=> [#<ParameterSubCategory id: 27, name: "Brightness", parameter\_category\_id: 8, created\_at: "2012-09-14 06:14:27", updated\_at: "2012-09-14 06:14:27">]

To add a Column Mapping name use the command:

ColumnMapping.create!(code:'<Code>', name:'<Name>')

This will result in output similar to:

1.9.2p290 :001 > ColumnMapping.create!(code:'VOL', name:'Volume')

(0.1ms) BEGIN

(1.0ms) SELECT 1 FROM "column\_mappings" WHERE LOWER("column\_mappings"."code") = LOWER('VOL') LIMIT 1

SQL (8.5ms) INSERT INTO "column\_mappings" ("code", "created\_at", "name", "updated\_at") VALUES ($1, $2, $3, $4) RETURNING "id" [["code", "VOL"], ["created\_at", Wed, 31 Oct 2012 14:18:53 EST +11:00], ["name", "Volume"], ["updated\_at", Wed, 31 Oct 2012 14:18:53 EST +11:00]]

(0.9ms) COMMIT

=> #<ColumnMapping id: 6, code: "VOL", name: "Volume", created\_at: "2012-10-31 03:18:53", updated\_at: "2012-10-31 03:18:53">

1. Migrating data to a new system

To restore a **pg\_dump** you pass the file to psql with an empty database. If you have an existing database with the same name, you need to drop it first and recreate it.

The command to drop the database is **dropdb**. So you 'su' to the **postgres** user and run the command:

$ sudo su - postgres

$ dropdb <database name>

$ createdb <database name>

Once you have done that, you can exit the **postgres** user, and restore the database dump:

$ exit

$ psql -U <user> <database name> < sql\_dump.sql

To restore the data, you need to untar it into your root directory. It is likely that your permission system won't allow you to create a directory under root, so you should create it manually, and assign the right permissions to it:

$ sudo mkdir /data

$ sudo chown <user>:<group> /data

$ cd /

$ tar xvf <tar file>

1. Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Version No. | Revision Date | Summary of Changes | Revised by |
| V1.0 | 15/11/12 | Initial | Stuart Allen |
| V1.1 | 15/12/12 | After internal Intersect review | Stuart Allen |
|  |  |  |  |
|  |  |  |  |

1. The Bagit format

BagIt is currently defined in an Internet Engineering Task Force ([IETF](http://en.wikipedia.org/wiki/IETF)) internet draft.

Quoting from the preamble of the Bagit entry on Wikipedia:

BagIt is a hierarchical file packaging format designed to support disk-based storage and network transfer of arbitrary digital content. A "bag" consists of a "payload" (the arbitrary content) and "tags", which are metadata files intended to document the storage and transfer of the bag. A required tag file contains a manifest listing every file in the payload together with its corresponding checksum. The name, BagIt, is inspired by the "enclose and deposit" method,[[1]](http://en.wikipedia.org/wiki/BagIt" \l "cite_note-ENCDEP-1) sometimes referred to as "bag it and tag it".

Bags are ideal for digital content normally kept as a collection of files. They are also well-suited to the export, for archival purposes, of content normally kept in database structures that receiving parties are unlikely to support. Relying on cross-platform (Windows and Unix) filesystem naming conventions, a bag's payload may include any number of directories and sub-directories (folders and sub-folders). A bag can specify payload content indirectly via a "fetch.txt" file that lists URLs for content that can be fetched over the network to complete the bag; simple parallelization (e.g., running 10 instances of "wget") can exploit this feature to transfer large bags very quickly. Benefits of bags include

* Wide adoption in digital libraries (e.g., the Library of Congress).
* Easy to implement using ubiquitous and ordinary filesystem tools.
* Content that originates as files need only be copied to the payload directory.
* Compared to XML wrapping, content need not be encoded, saving time and storage space.
* Received content is ready-to-go in a familiar filesystem tree.
* Easy to implement fast network transfer by running ordinary transfer tools in parallel.

Further information about the Bagit hierarchical file packaging format can be found at various places on the Internet, including

Internet Engineering Task Force – <http://www.ietf.org>

Wikipedia – <http://en.wikipedia.org/wiki/BagIt>

Version 0.97 of the Bagit specification - <http://tools.ietf.org/html/draft-kunze-bagit-08>

1. RIF-CS

Quoting from the Global Registries website (<http://globalregistries.org/rifcs.html>):

The **Registry Interchange Format - Collections and Services** (RIF-CS) Schema was developed as a data interchange format for supporting the submission of metadata to a collections service registry. It is based on ISO2146 but only includes elements needed for a collection service registry and so is not a full binding to the standard.

A collection in the RIF-CS Schema context could be a repository, a registry, a collective work or an index/database. There are no hard and fast rules about what constitutes a collection and it is up to the data providers to consider what their collections are and what metadata should be provided. The RIF-CS schema also supports other registry object types, namely services, activities and parties. Any or all of these along with their relations to each other are able to be expressed in RIF-CS format.

The Australian National Data Service (ANDS – http:// <http://www.ands.org.au>) uses the RIF-CS standard for management of data in the Australian Research Data Commons. It provides a training resource for RIF-CS at <http://www.ands.org.au/training/rif-cs/index.html>.

ANDS uses the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH – see <http://www.openarchives.org/pmh/tools/tools.php>) to collect RIF-CS data.

1. Data File Upload Scenarios

If a TOA5 format CSV data file is uploaded to HIEv, the TOA5 format is automatically detected and the processing is as shown in the table below.

Table : Action on uploading files

| **IF...** | **Type selected is** | **File type is** | **File overlap is** | **File name is** | **THEN...** | **Resulting type will be** | **Resulting file name will be** | **Resulting messages to user** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | RAW | TOA5 | None | does not already exist |  | RAW | as per uploaded | success |
|  | RAW | TOA5 | None | already exists |  | RAW | suffixed - see (1) | filename change (3) |
|  | RAW | TOA5 | Safe | does not already exist |  | RAW | as per uploaded | safe replacement (2) |
|  | RAW | TOA5 | Safe | same as file being replaced |  | RAW | as per uploaded | safe replacement (2) |
|  | RAW | TOA5 | Safe | already exists (but is not the file being replaced) |  | RAW | suffixed - see (1) | safe replacement (2), filename change (3) |
|  | RAW | TOA5 | Unsafe | does not already exist |  | ERROR | as per uploaded | bad overlap (4) |
|  | RAW | TOA5 | Unsafe | already exists |  | ERROR | suffixed - see (1) | bad overlap (4), filename change (3) |
|  | RAW | Non-TOA5 | N/A - only for TOA5 | does not already exist |  | RAW | as per uploaded | success |
|  | RAW | Non-TOA5 | N/A - only for TOA5 | already exists |  | RAW | suffixed - see (1) | filename change (3) |
|  | Not RAW | TOA5 | N/A - we don't check unless RAW | does not already exist |  | as specified | as per uploaded | success |
|  | Not RAW | Non-TOA5 | N/A - only for TOA5 | already exists |  | as specified | suffixed - see (1) | filename change (3) |

(1) suffixed means appending \_1 (or the next available number) - e.g. blah.dat becomes blah\_1.dat (or blah\_2.dat if blah\_1.dat already exists)

(2) MESSAGE: The file replaced one or more other files with similar data. Replaced files: <filenames here>

(3) MESSAGE: A file already existed with the same name. File has been renamed.

(4) MESSAGE: File cannot safely replace existing files. File has been saved with type ERROR. Overlaps with <filenames here>