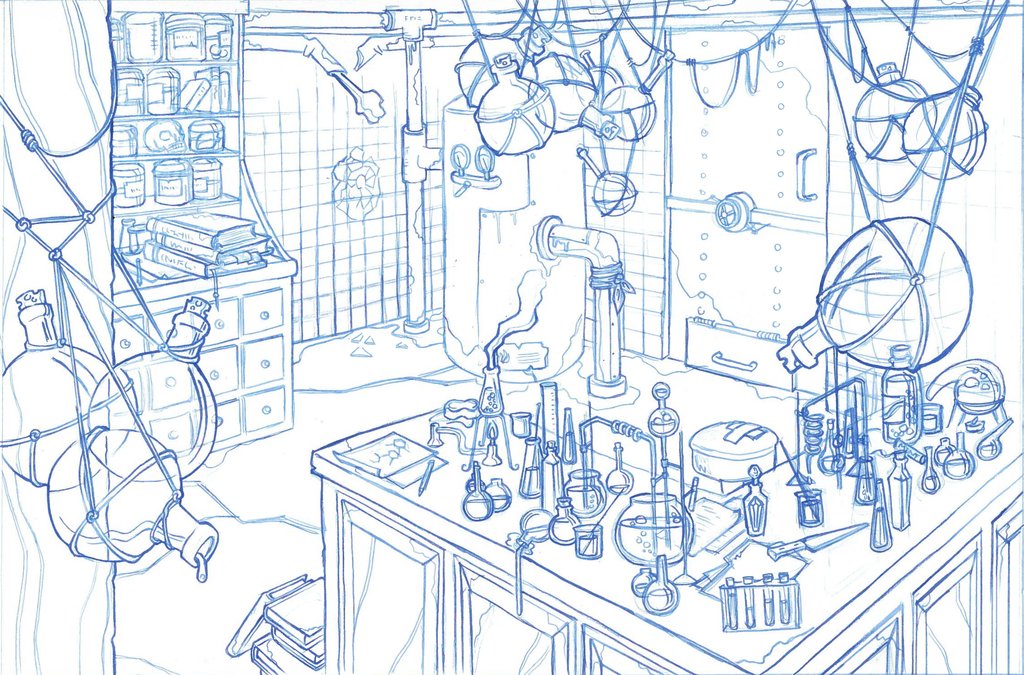
Tracker

Orderly views of your chaos



Tracker 1

Orderly views of your chaos 1

Introduction 3

Environment 3

Home Page Quick Entry Forms 5

Register ID 5

Find ID 5

Transfer 5

Registration 5

Tags 6

Project entry and verification 7

Initial Contents 7

Registering stacks 8

Component display and update 9

Rack verification 10

Transfers 12

Cherry Picking 13

Graphs 14

EZPASS Creation 16

Walk Up Sequencing File Creation 18

Search 18

A very brief technical overview 20

# Introduction

Tracker is used to track lab components: plates, tube and racks. Components are tracked via two steps: registration and transfers. First individual components are registered into Tracker. Next, transfers are done between registered components.

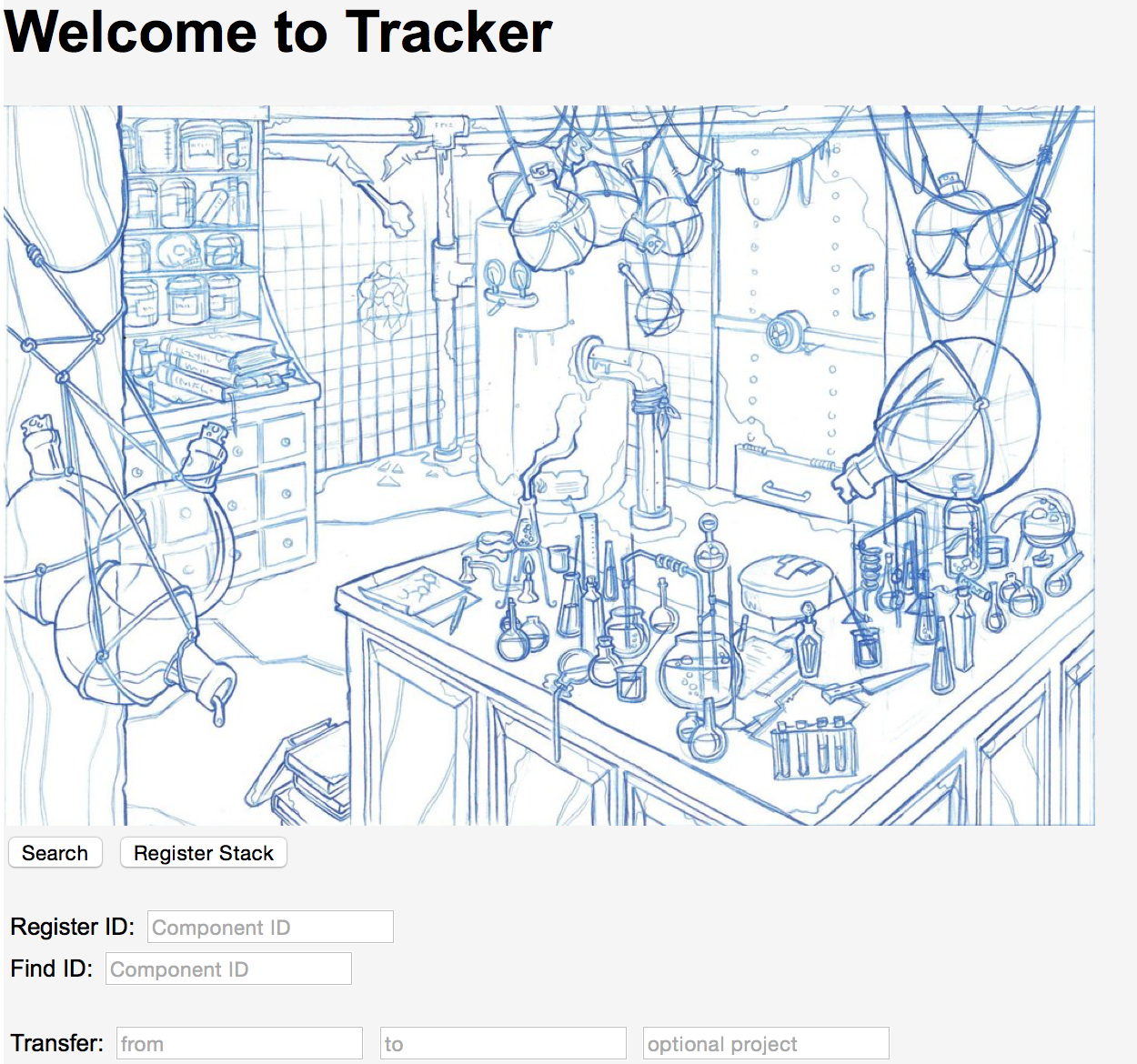
Using the information input during registrations and transfers, tracker provides verification and reports of component contents. In addition, lab protocols can be tracked and viewed later to see the steps and components that went into making a final product.

This document is a users guide to Tracker.

# Environment

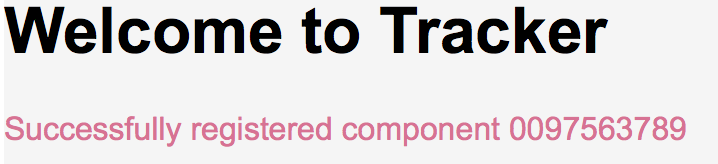
Tracker runs as a web application, available from any web browser that can connect to Broad’s internal network. To access Tracker enter the url [http://btllims.broadinstitute.org:9000](http://btllims.broadinstitute.org:9000/) in the browser’s navigation bar.

Tracker’s home page has a number of “quick entry” forms at the bottom of the page to start the most common operations. In addition component searches and registration of plate stacks can be done. The home page is shown below:



Home Page

Status messages are displayed at the top of the page in red when operations complete or additional information must be provided. For example:



Message Display

The home page can be returned to from any other page by simply clicking on the *Home* button located at the bottom left corner of the page.

## Home Page Quick Entry Forms

### Register ID

Before any operations can be done with a component it must be registered with a unique ID supplied by the user. Normally this ID is the barcode attached to the component, which can be easily scanned using a barcode scanner. To register a component simply input its ID in the *Component ID* field following the *Register ID:* label and hit return to be directed to the forms used to complete the registration process. Additional information about the registration process is found in the section “Registration”.

### Find ID

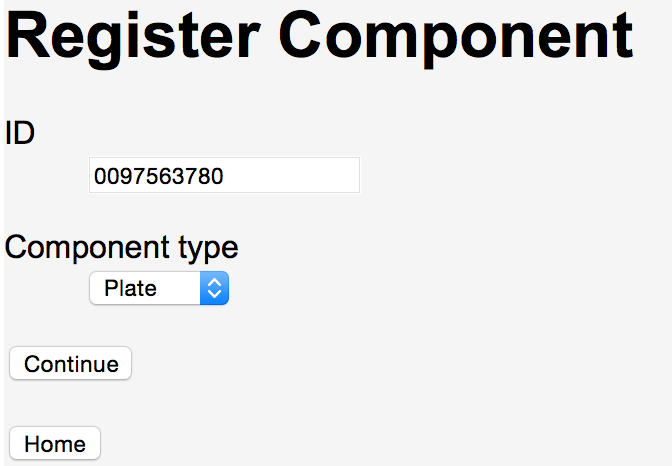
Once a component is registered it can be edit, have its contents transferred to another component, or have verification or reports of its contents done. To go to the page that displays details about an individual component and contains the preceding operations, input the component’s ID in the *Component ID* field following the *Find ID:* label and hit return. Additional information about a components display page is found in the section “Component display and update”.

### Transfer

The *Transfer:* quick entry form is used to transfer contents between registered components. The source component ID is entered in the *from* field, the target ID in the *to* field and optionally an associated project is entered into the *optional project* field. Upon hitting the return key, if any additional information pertaining to the transfer such as a component quadrant is needed, a new page appears to query for the additional data. Additional information about transfers is found in the section “Transfers”.

# Registration

After entering an ID in the *Register ID* quick entry form a page appears to enter the component type. Based on the component type a final registration page appears to enter the type specific attributes for component. Following are examples of the two pages:



Register Component Type

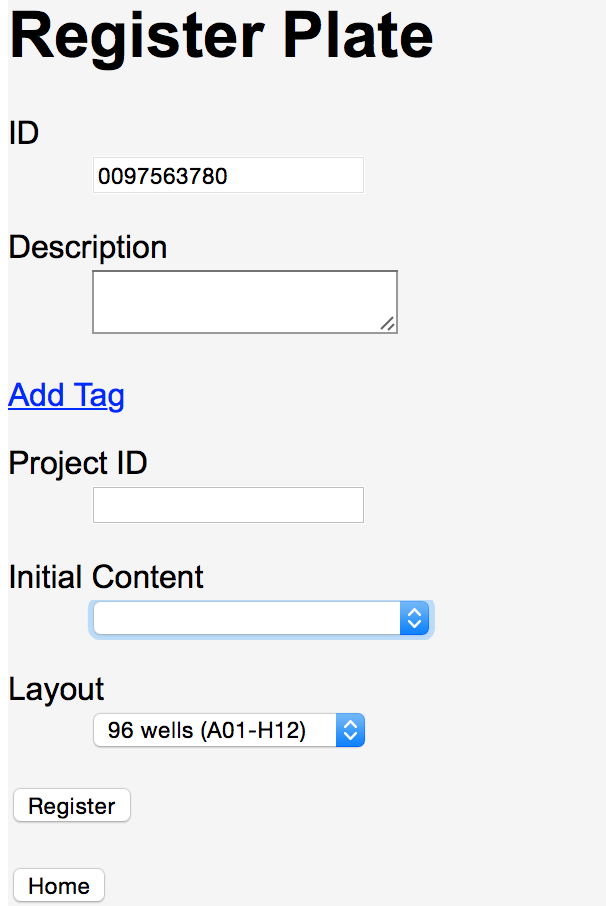


Plate Registration

*Description* can be set to any value. Searches can find any text contained within a description. For example, if one component has the description “The best component” and another “The worst component” searching for “component” will find both the preceding entries.

*Layout* must be specified for plates and racks. Two layouts are available: 96 and 384.

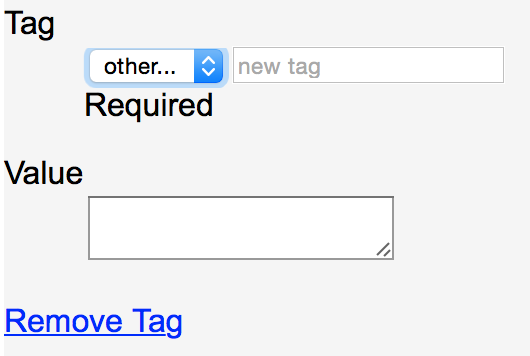
## Tags

Sometimes there is a need to add additional attributes to supplement the fields already in place for a component type. These attributes may simply be a tag name, such as “Destroyed”. Other attributes may have both a name and a value, for example “Reason for destruction”: “Dropped on floor”. To add an attribute to a component, choose the Add Tag option. That will display a place to put the tag name and an optional associated value. For example:



Add Tag form

The Tag field is a drop down list with all previous tag names used and a final choice of other…. To avoid duplicate tag names (e.g., grams, gm, weight (gm), …) it is best to choose one of the names already in the tag list. However, if none of the previously used names is appropriate then the other… choice can be used to enter a new tag as shown below:



Add Tag other… form

Any number of tags can be added to a component. To remove a tag, select the *Remove Tag* option displayed below the tag name and value.

Tags can be used to search for components. See the “Search” section for more details.

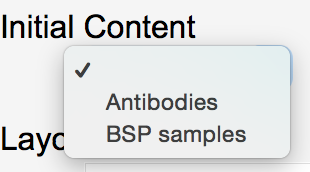
## Project entry and verification

*Project ID* is used to verify that the component is part of a specified Jira issue. A *Project ID* set for a plate or rack must be the related Jira ticket ID (e.g., SSF-780) that the component is being used for. If *Project ID* is not set correctly the registration is aborted and an error message is displayed containing the actual Jira ticket ID, if any, the component is associated with.

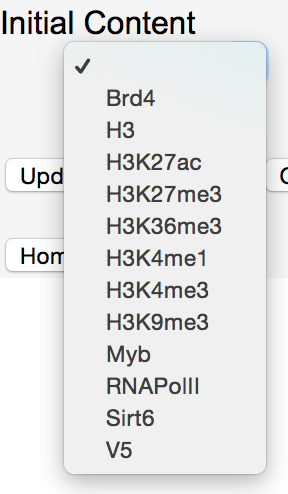
Within Jira a rack is associated with a ticket when a BSP sample sheet is attached to the Jira ticket. A plate’s association is done when the Jira ticket's “Plate Barcode” field is set to the plate ID.

## Initial Contents

*Initial Content* is used to set the initial contents, if any, of the component. *Initial Content* is often used for components that will be the source of transfers. Contents for components without initial contents are determined by the initial contents of source components. Currently there are three types of initial contents that can be set: MIDs (Molecular IDs or barcodes) for a plate, an antibody for a tube, and a tube type for a rack. Following is a display of the drop down lists used for *Initial Content*.



Rack Tube Type Selection



Tube Antibody Selection

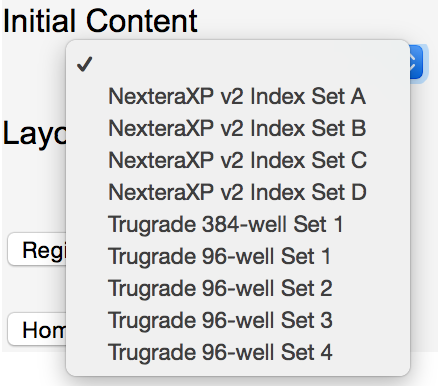
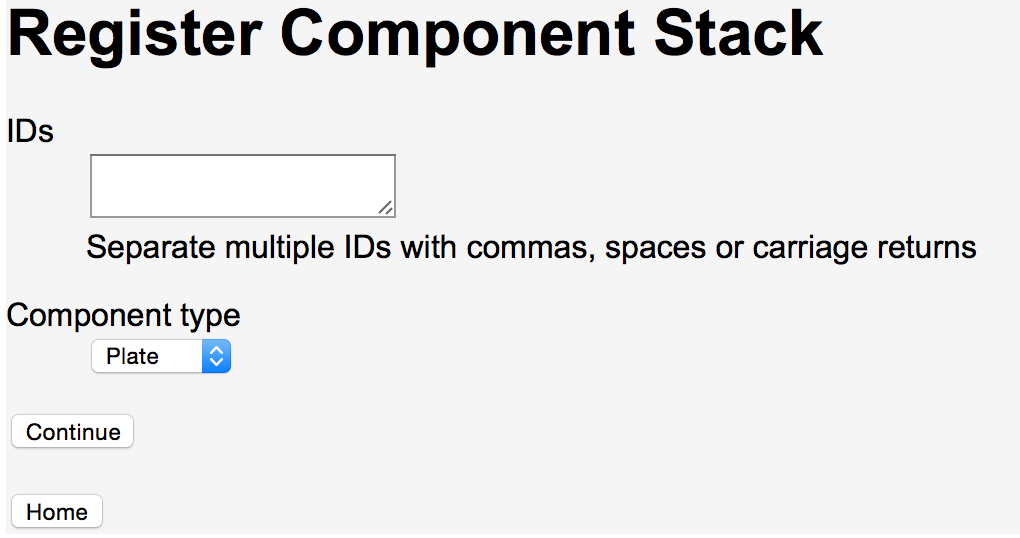


Plate MID Selection

# Registering stacks

Registering stacks of components is similar to registering a single component except that a list of IDs, instead of a single ID, is registered. All the components in the stack must be the same type and the attributes for the components, except for ID, are entered once and applied to all the components in the stack. To register a stack of components press the *Register Stack* button to display the following page:



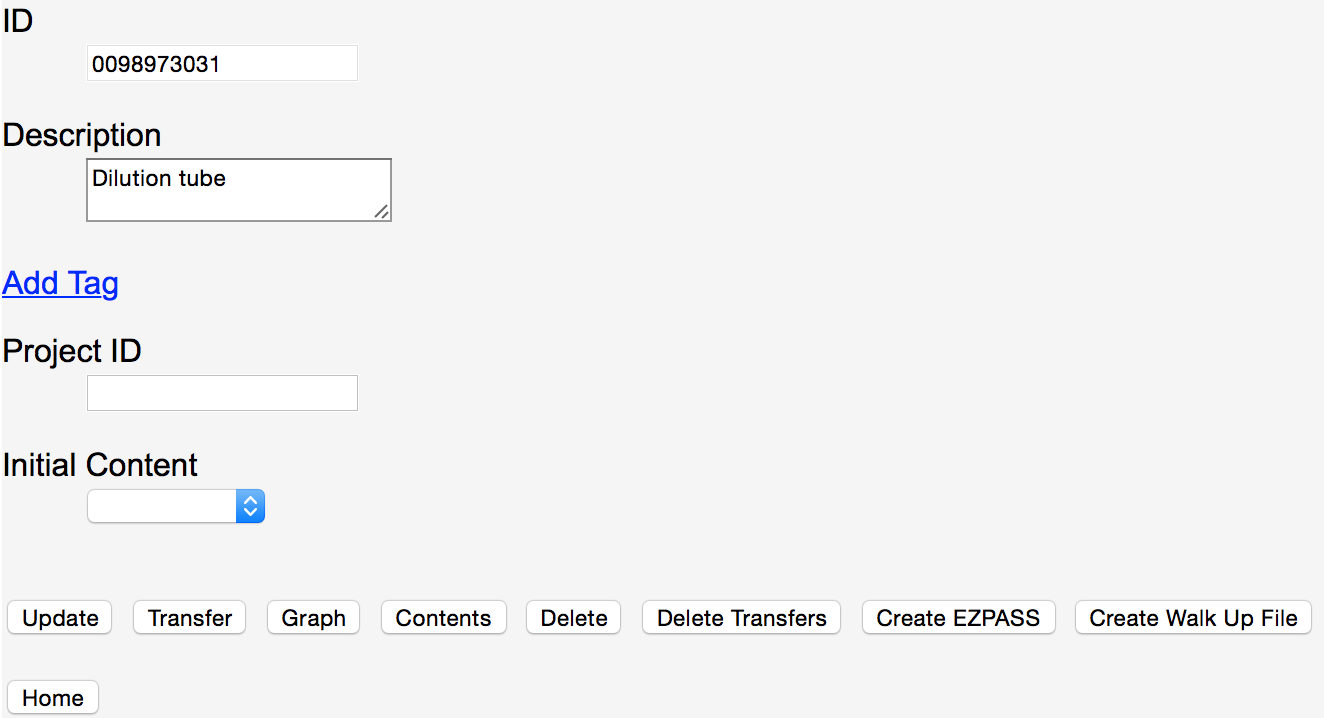
Register Stack IDs

As noted in the display, commas, spaces or carriage returns can separate IDs. If using a barcode scanner to scan in a stack be sure that the scanner inserts one of the required characters between IDs.

The final stack registration page is component type specific and identical to the pages used for registering a single ID entered in the home page’s *Register ID* quick entry form.

# Component display and update

When a component is selected, either via the *Find ID* quick entry form or from *Search* results, details about the component are displayed. Following is the display for tube 0098973031:



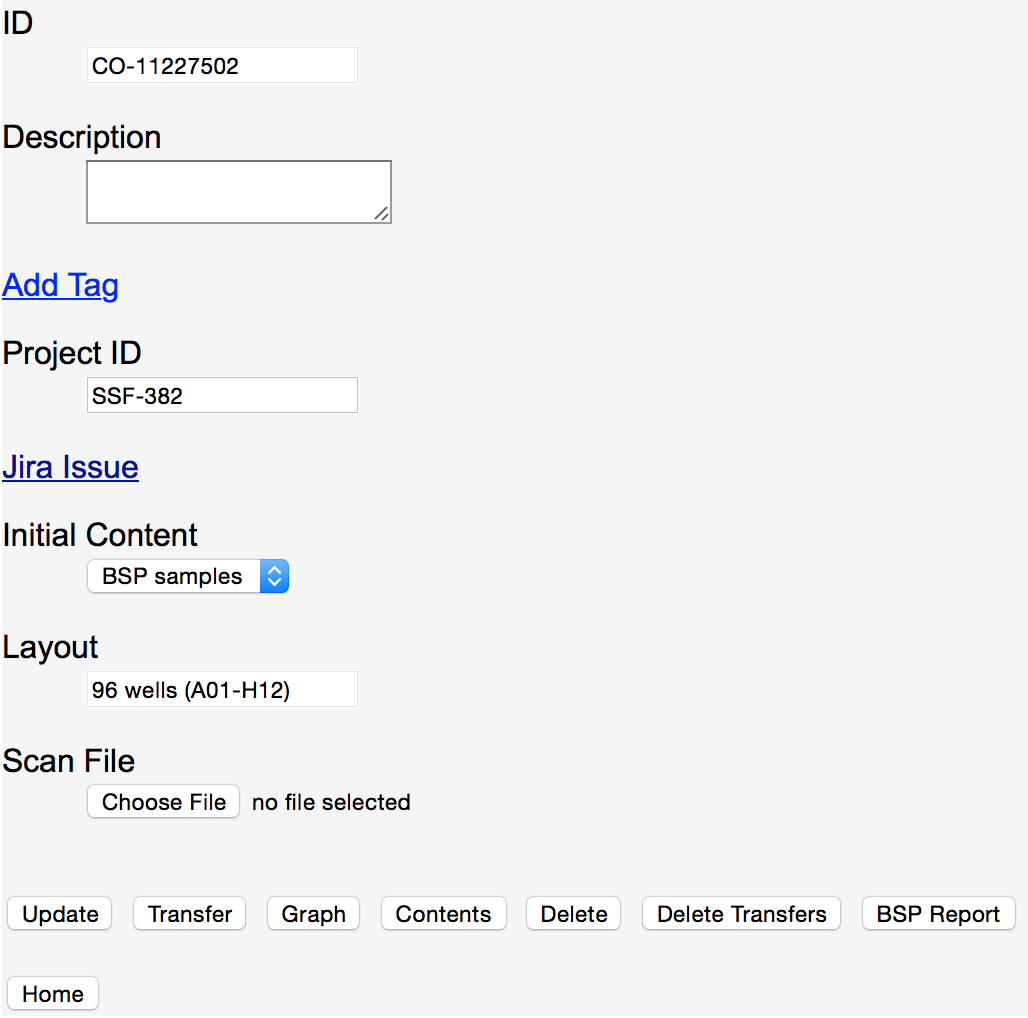
Component Display

In addition to the component details displayed, there are a number of operations that can be done from this page:

* Fields can be modified and then updated using the *Update* button. The ID field cannot be modified.
* Transfers, with the displayed component as the source, are recorded using the *Transfer* button.
* A graph of all transfers, direct or indirect, to and from the component is displayed using the *Graph* button.
* Contents of the component are displayed using the *Contents* button. Contents include all initial contents set or transferred into the component as well as sample names.
* The component and all its associated direct transfers are deleted via the *Delete* button.
* Transfers from the component to another component specified in a follow up page are deleted using the *Delete Transfers* button. This operation only deletes direct transfers and does not delete the component itself.
* An EZPASS is created using the *Create EZPASS* button. The EZPASS includes information deduced from transfers, such as molecular barcodes and sample information. EZPASSes can only be created for tubes. EZPASS and walk up sequencing sheet creation are only available for tubes.
* A walk up sequencing sheet is created using the *Create Walk Up File* button. The walk up sequencing sheet includes molecular barcodes and sample names.
* Verification of a component’s sample content is done via the *BSP Report* button. BSP reports are only available for racks.

## Rack verification

Following is the display for a rack:



Rack Display Page

After data is entered for a rack component verification of reported contents vs. expected contents can happen. Expected contents are determined by data entered in the Jira ticket associated with the component. Note that the Jira ticket can be displayed by selecting the *Jira Issue* option that appears once a *Project ID* has been set for the component.

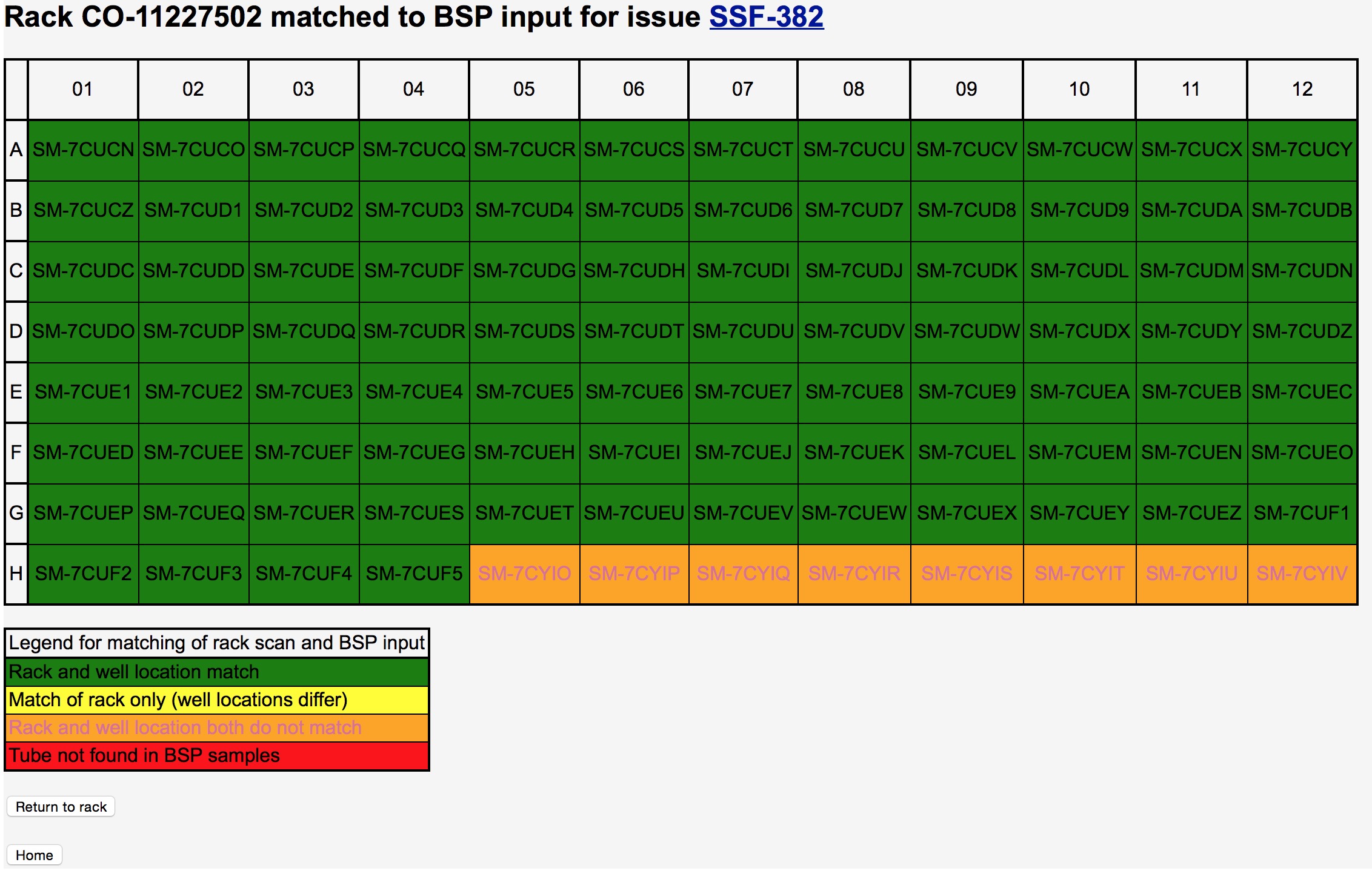
To verify the contents of a rack the following must have taken place:

* In the associated Jira ticket the BSP sample sheet must be attached.
* A scan of the 2D barcodes of the tubes in the rack must be done and associated with the component using the *Scan File* option in the rack display page.

The rack scan file is a spreadsheet or csv file formatted as follows (DATETIME column is optional):

|  |  |  |  |
| --- | --- | --- | --- |
| DATETIME | RACK | TUBE | BARCODE |
| 12/22/14 14:21 | CO-11227502 | A01 | 177159455 |
| 12/22/14 14:21 | CO-11227502 | B01 | 177159432 |
| 12/22/14 14:21 | CO-11227502 | C01 | 177159431 |
| 12/22/14 14:21 | CO-11227502 | D01 | 177159408 |
| … | … | … | … |

After the BSP and rack scan spreadsheets have been set, selecting the *BSP Report* button verifies the contents of the rack. Following is an example of verification results:



BSP Report

The page shows how the rack scan results compare with the reported contents from the BSP sample sheet. The match between the rack scan and the sample sheet is done via the tubes’ barcodes.

As the legend in the lower left of the screen indicates green locations are an exact match. For mismatches, the severity of the mismatch is indicated by colors (yellow to red) explained in the legend. The label (e.g., SM-7CUCN) in a box is the BSP sample ID for the tube found in that location. The locations used are from the scan entered for the rack scan. For example the tube in location H05, sample SM-7CYIO, was found at location H05 in the rack scan but found in a different rack in Jira’s BSP data.

If the BSP data included information about antibodies to associate with the samples the severity of the mismatch is increased if the sample location changed to a place with an antibody association different than the original location.

# Transfers

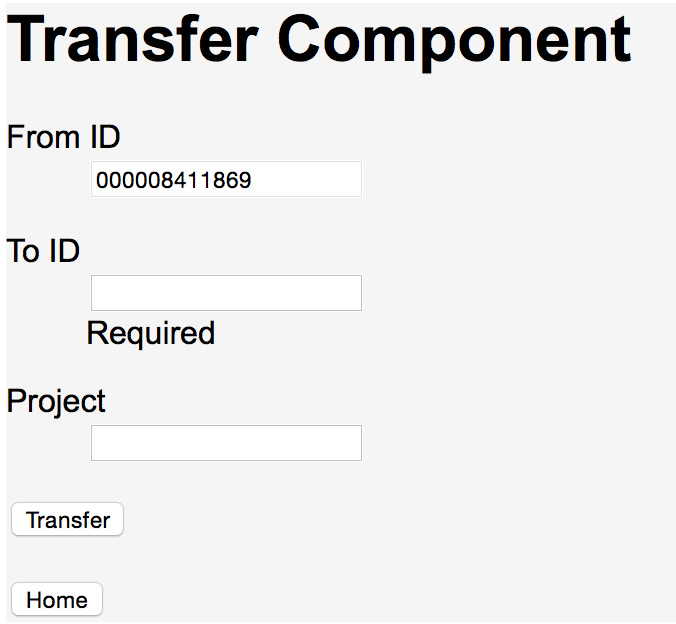
Transfers are used to track when the contents of one component are moved to another component. Components must be registered before they can be part of a transfer. The source and destination of transfers are identified by component IDs. Valid transfers are:

- Rack to Rack, Plate or Tube

- Plate to Plate or Tube

- Tube to Tube

Following is example of the transfer page initiated for a tube 000008411869:

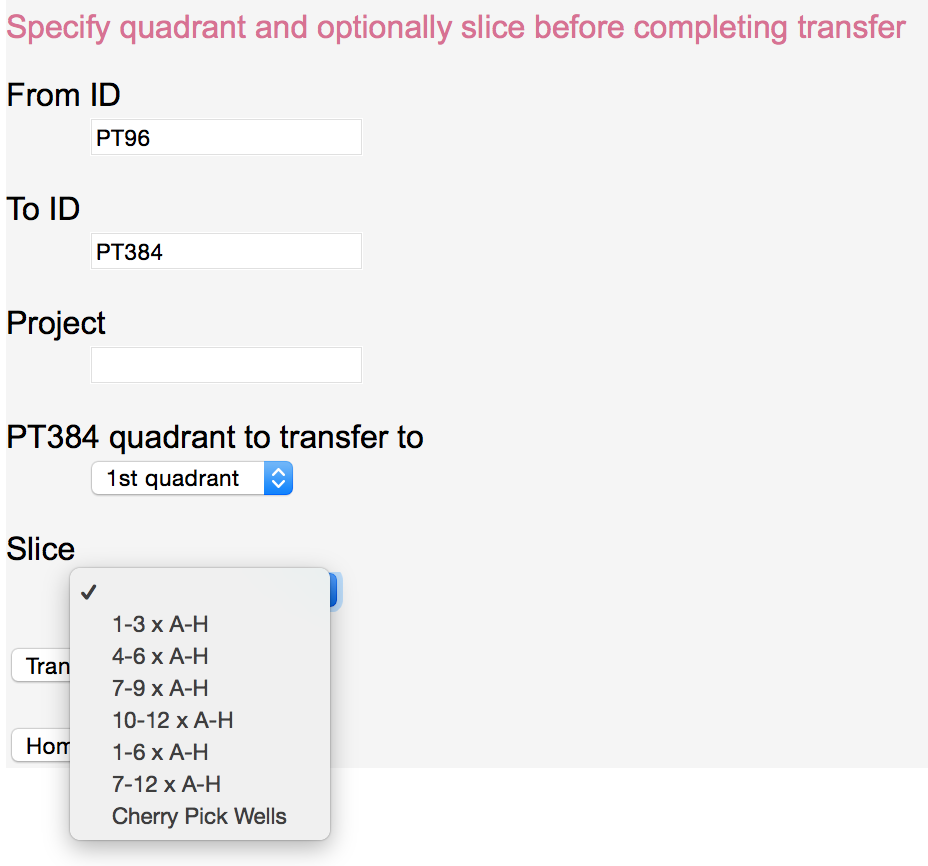


Initial Transfer Page

In addition to the *from ID* and *to ID* an optional *Project* can be specified. If project is specified a check is done that the project is set for either a component transferred into the source, directly or indirectly, or the transfer source itself. If the specified project is not found then an error is reported and the transfer is aborted.

Transfers of plates and racks need not be for the entire component. When transferring between different size components, for example a 96-well and 384-well plate, a quadrant must be specified. In addition, for all transfers where the source is a rack or plate, the transfer can be limited to 24 or 48-well slices or cherry picked wells.

Following is the Transfer page, from a 96-well plate to a 384-well plate, requesting quadrant and optional slice information. Slice choices are displayed in the drop down list as *rows x columns* and *Cherry Pick Wells* as a final choice.

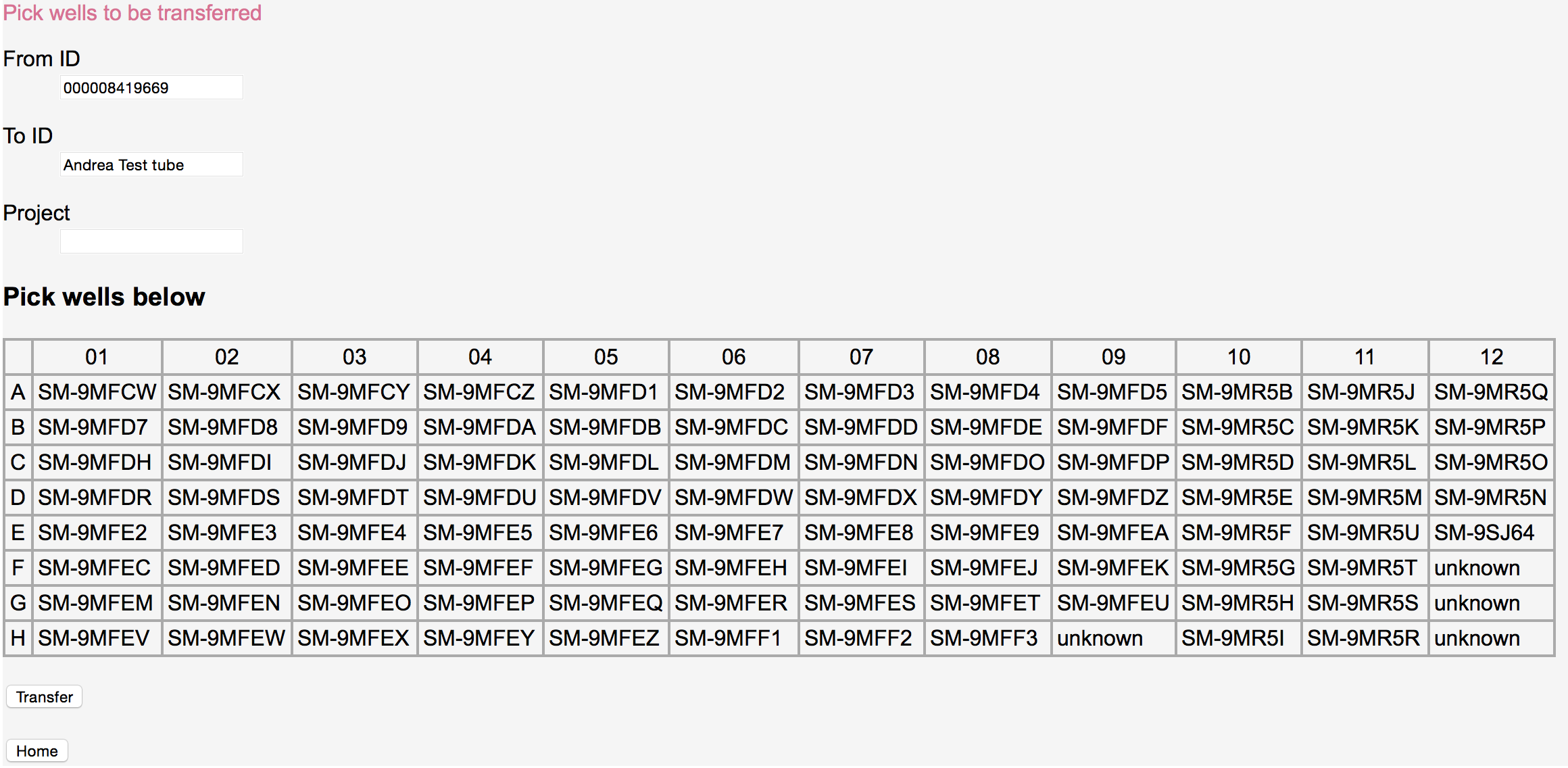


Transfer Page to Specify Transfer Quadrant and Slice

When both a quadrant and slice are selected the slice is taken from 96-well coordinates. For example, in the page above, if slice *10-12 x A-H* is chosen from the *1st quadrant* then the 24 wells covered by the slice 10-12 x A-H in the 96-well source plate are transferred. Similarly, if the transfer was from a 384-well plate to a 96-well plate or between 384-well components, the wells transferred into are 10-12 x A-H in the 96-well target plate or a virtual 96-well component between the 384-well components.

## Cherry Picking

To pick individual wells the last slice choice, *Cherry Pick Wells*, must be selected. When a request to cherry pick wells is submitted an additional page appears to do the selection, as shown below:



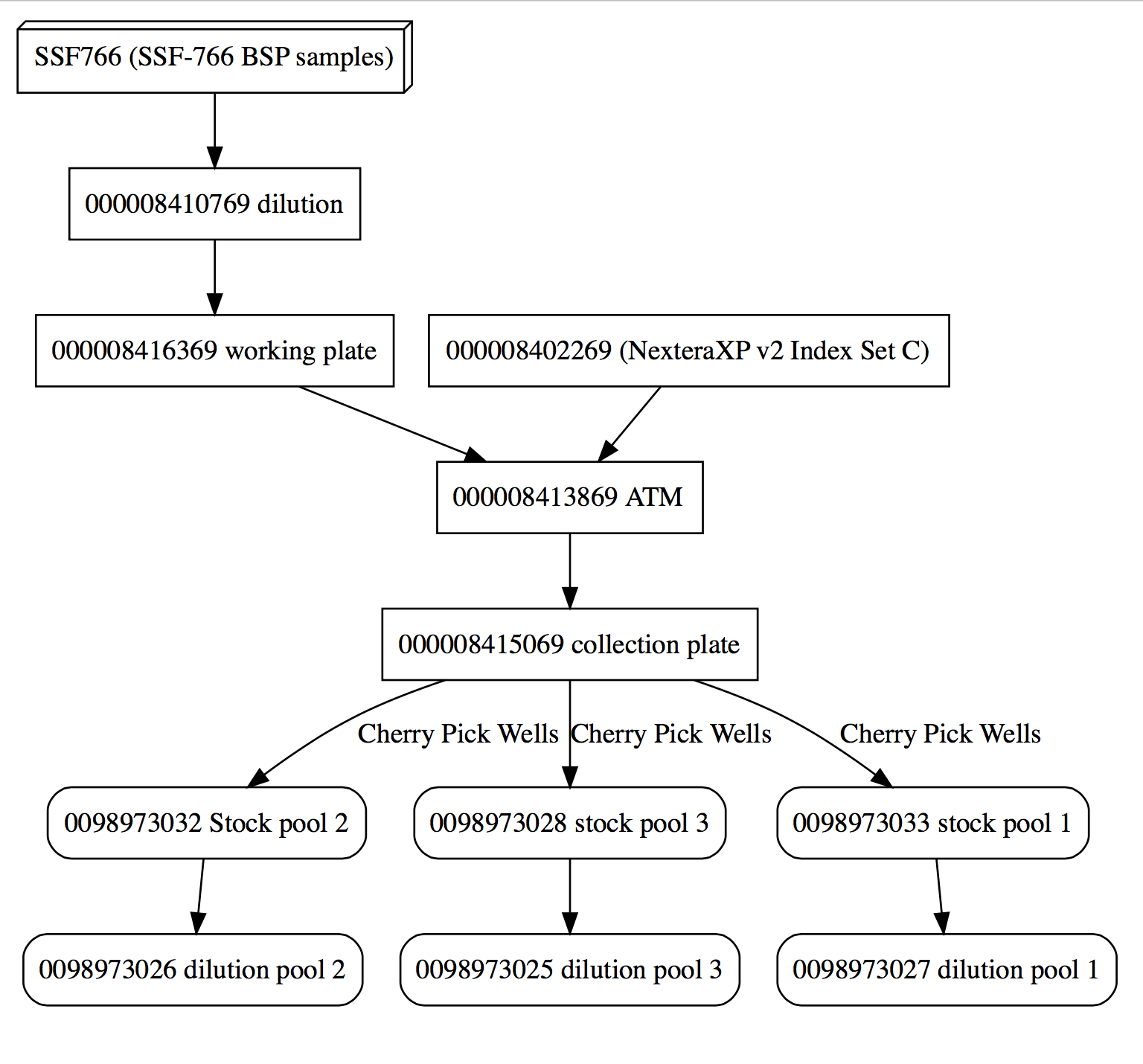
Transfer Page to Cherry Pick Wells

When a sample ID is known for a well’s contents it is displayed, otherwise “unknown” is set in the well’s location. To select wells the following UI choices are available:

* Mouse click to select an individual well and unselect any other wells previously selected.
* Mouse click while holding down the control key to select or unselect an individual well while leaving the selection state of other wells unchanged.
* Mouse click while holding down the shift key to select the range of wells between the last selected well and the one now being selected and unselect any other wells previously selected.
* Mouse click while holding down both the shift and control key to select a range of wells while leaving the selection state of other wells unchanged.

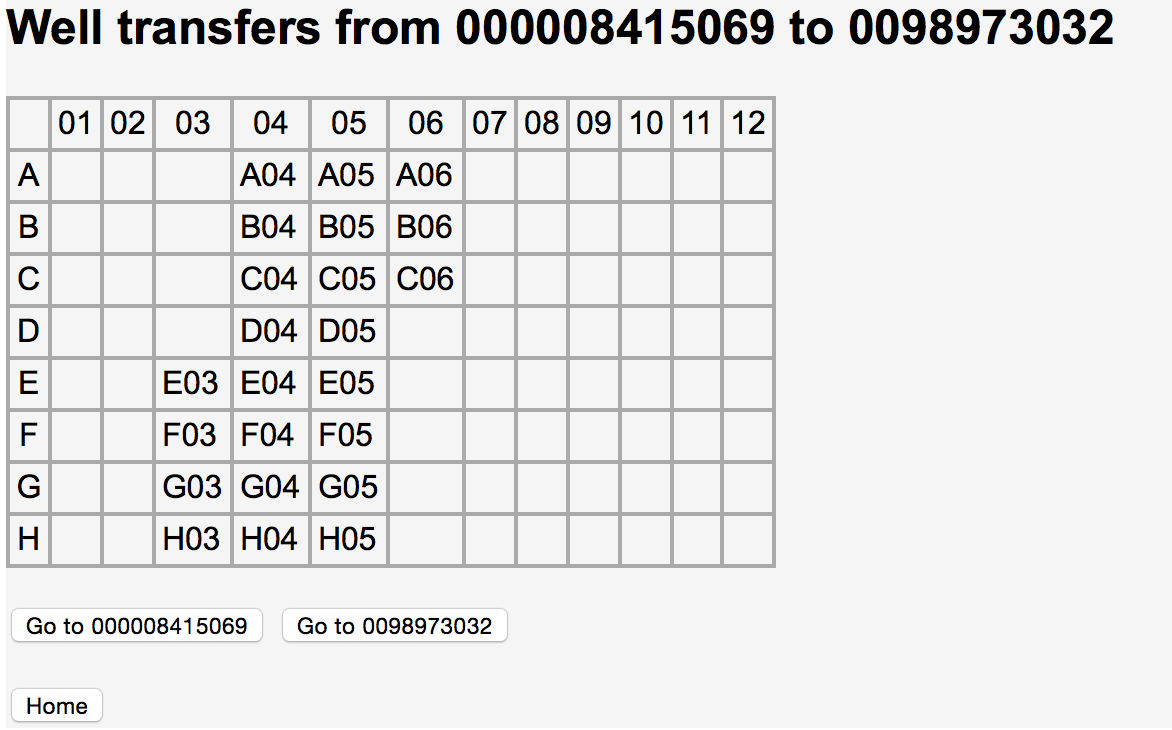
## Graphs

Tracker keeps track of transfers using directed graph. Components are the nodes of the graphs and transfers are the edges. Using these graphs a number of operations can be done. The most basic operation is to display the graph. When a display is requested, using the *Graph* button, a pop-up window displays the graph for the component. Other components transferred into or out of the component, directly via a single transfer or indirectly via multiple transfers, are shown in the display. Between the components are arrowed lines showing the directions of the transfer. Within each component’s rectangle is the component’s ID, description (up to 23 characters), project and initial contents. Following is a graph for the collection plate 000008415069. The project (SSF-766) associated with the BSP samples input rack SSF766 is displayed along with descriptions for other components and the initial contents (NexteraXP v2 Index Set C) of 000008402269. Racks are displayed as 3-d rectangles, plates as plain rectangles and tubes as rectangles with rounded corners.



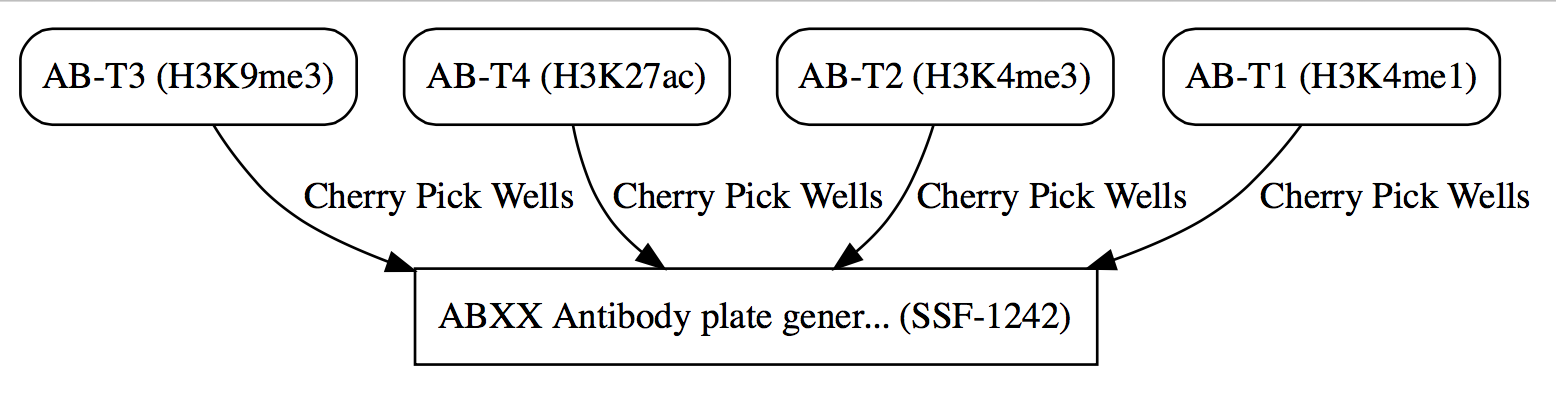
Component Graph

Clicking on a component’s label navigates to the associated component’s display. Clicking a label on the line representing a transfer normally displays a map of the source wells, with a destination well location set for each source well being transferred. Following is the display when the “Cherry Pick Wells” label for the line between 000008415069 and 0098973032 is selected:



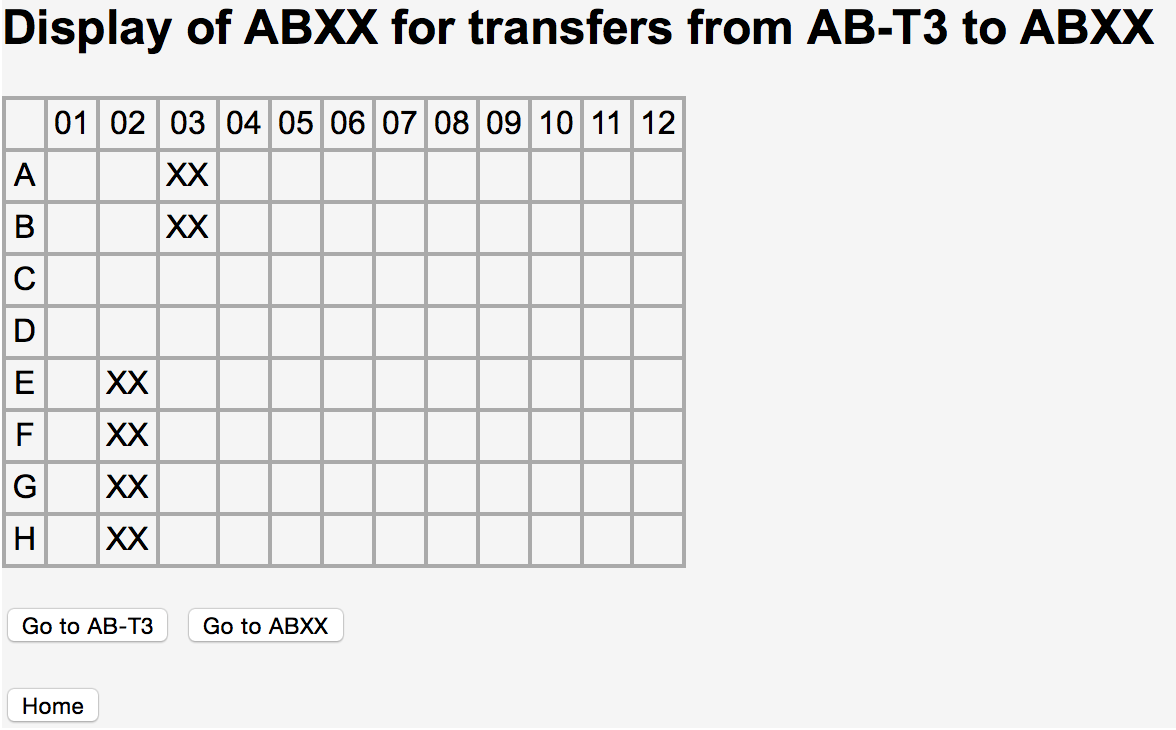
Well Transfers

Note, unlike the previous example, when transfers take place between different size components the source and destination wells may not be the same. Also, if the transfer is between a tube and a plate, the destination plate is shown with the wells marked to indicate where the tube contents is transferred. For example, if the plate ABXX has antibody tubes transferred into its wells, as show below:



Tube to Plate Graph

Then the transfers from a tube to the plate are displayed as shown in the following example, with each destination well in the plate marked by *XX*:



Tube to Plate Transfer

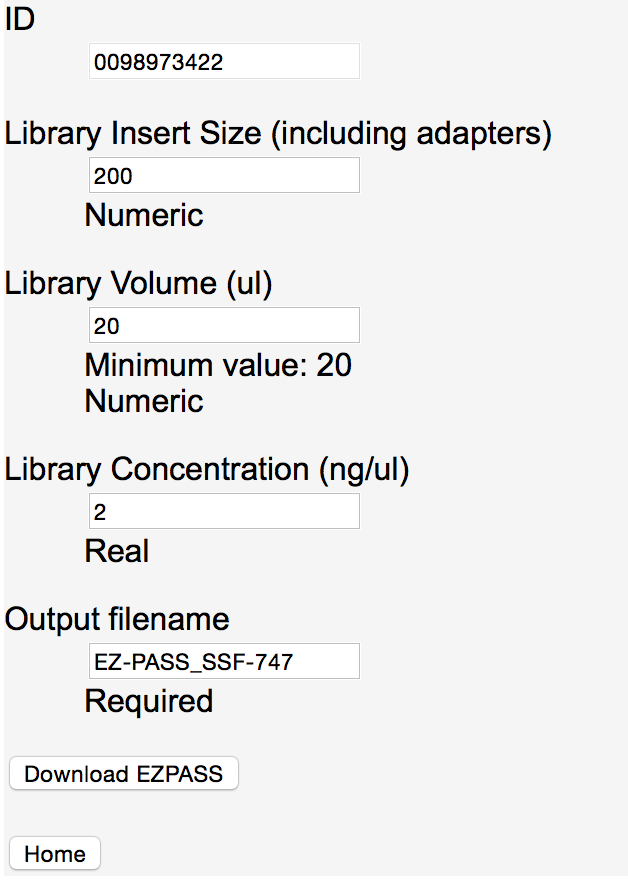
Finally, when there is no label on a graph’s transfer line that infers that the source and destination are the same size and that the entire source was transferred.

# Contents

## 

## EZPASS Creation

When an EZPASS is requested for a tube a new page appears to request additional attributes needed to complete the EZPASS. Following is an example of the “Create EZPASS” page:



Create EZPASS

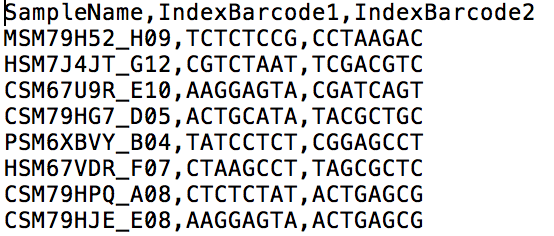
After all the requested attributes are filled in, the *Download EZPASS* button is selected to generate the EZPASS. EZPASS creation is completed by doing a traversal of the directed graph stored for transfers. The final contents of a component are calculated based on the contents of all the components leading into the target component. In particular, MIDs (molecular IDs or barcodes) are attached to samples based on well locations of the MIDs and samples. For example, in the preceding graph for tube 0098973031, the sample contents originating in the rack CO-14655311 are combined with the “NexteraXP v2 Index Set A” MIDs to create MID tagged samples in the ATM plate and beyond. Rules for combining MIDs and samples are:

* Once a sample has MIDs attached no additional MIDs can be attached to the sample.
* If more than one plate of MIDs is transferred into a plate with samples, if there are overlapping wells from the multiple transfers, then the MIDs from the first transfer recorded in Tracker is used.

When an EZPASS is generated it is downloaded as a spreadsheet file named by default either *EZ-PASS\_ projectName.xlsx* or *EZ-PASS\_componentID*.xlsx, for example EZ-PASS\_SSF-747.xlsx. If a project is found for the target component or any of the components transferred into the target then that project name is used. Otherwise the filename uses the target component id. The filename can be overridden by specifying an *Output filename*.

## Walk Up Sequencing File Creation

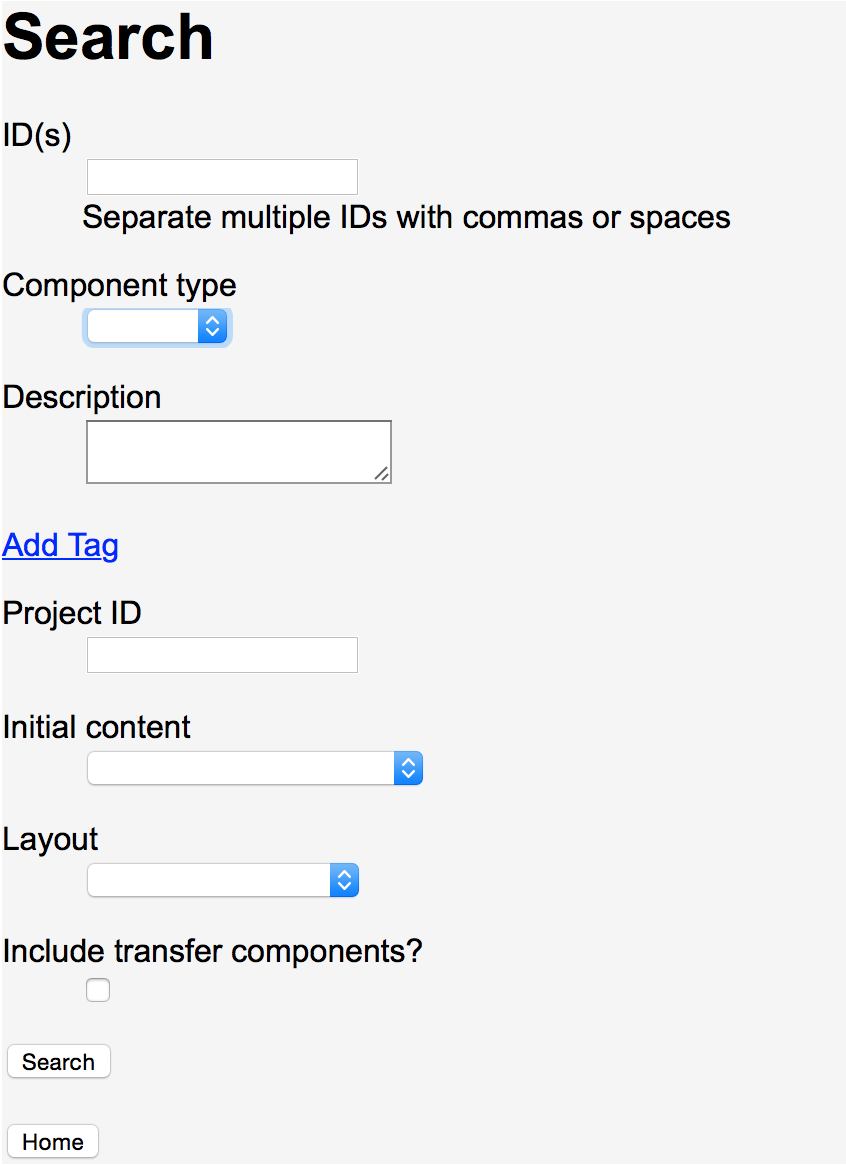
As an alternative to an EZPASS, a walk up sequencing file can be created for a tube by selecting the *Create Walk Up File* button. Similar to an EZPASS, a walk up sequencing sheet is created by looking at the contents, in particular MIDs and samples, which have been transferred into a tube. A walk up sequencing sheet is a “csv” (comma separated value) file with three columns: SampleName, IndexBarcode1 and IndexBarcode2. SampleName is based on the collaborator sample name; IndexBarcode1 contains the P5 barcode sequence; IndexBarcode2 the P7 barcode sequence. The SampleName is created by first replacing any non-alphanumeric characters in the collaborator sample name, except underscore (“\_”) and dash (“-“), with an underscore. Following the collaborator sample name, if an antibody was to be matched with the sample, an underscore, followed by the antibody name, is appended to the sample name. Finally to complete the sample name, and guarantee uniqueness, an underscore followed by the sample’s original position in the input sample rack is placed at the end of the name. Following is an example of the start of a walk up sequencing sheet for samples not containing antibodies.



Walk Up Sequencing Sheet

# Search

Searching for one or more components based on multiple criteria is done using the search page displayed when the *Search* button on the home page is selected. A blank search page looks as follows:



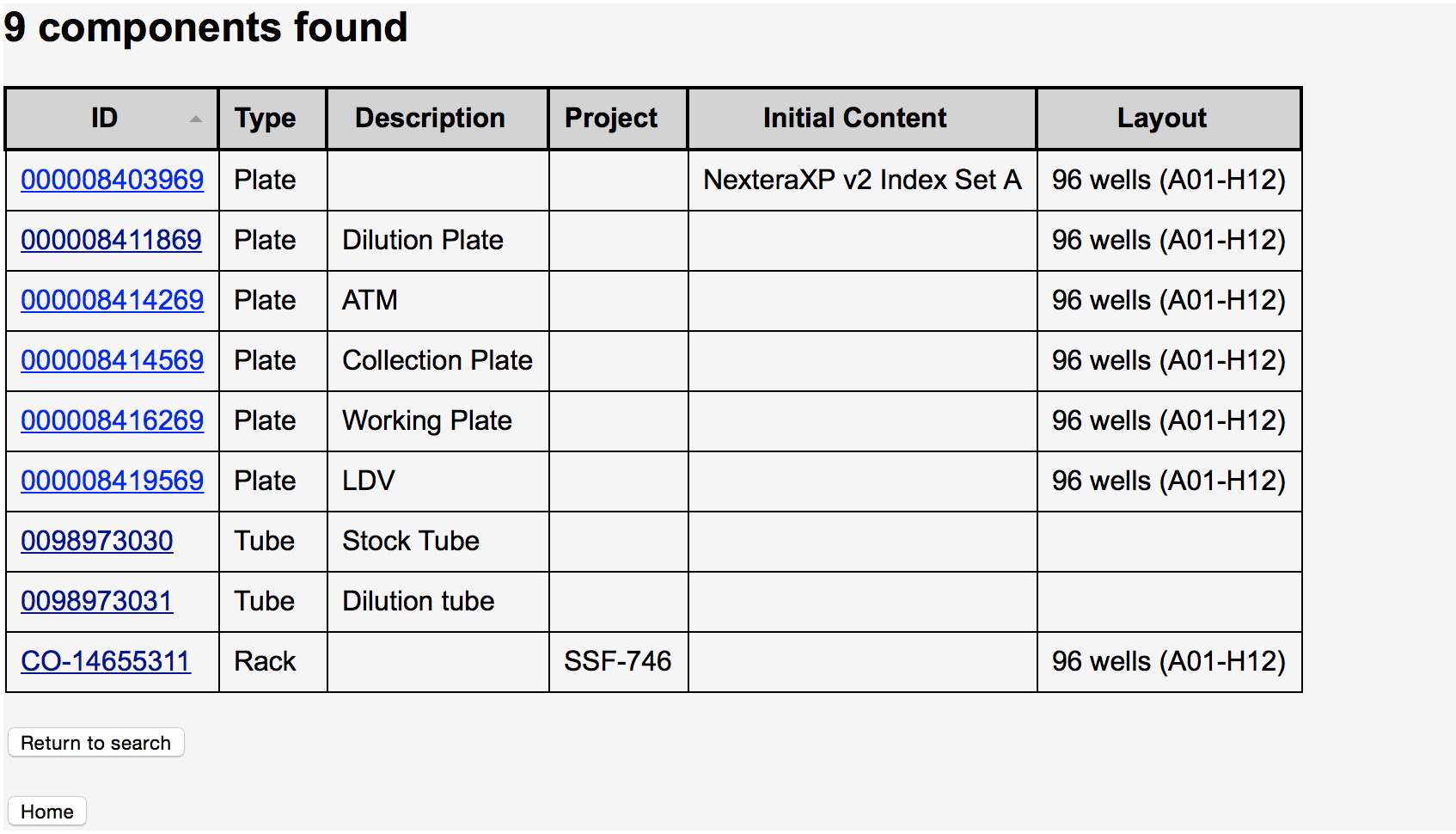
Search

Each criteria specified, with the exception of *Include transfer components?*, narrows the selection criteria. If no criteria is specified all registered components are found. Otherwise the search results are limited to match all the criteria specified. Attribute matching rules are:

* *ID(s)*, *Component type*, *Project ID*, *Initial content* and *Layout* must be exact matches.
* *Description* matches if the specified text is contained anywhere within a component’s description. Description matching is case insensitive.
* Tags, added via the *Add Tag* selection, must specify a *Tag* name to match. If a *Value* is specified with the name, matching components must have at least one tag with the specified *Tag* name and a value containing the specified *Value*. *Value* matching is case insensitive. If a *Value* is not specified then all components with the specified *Tag* name are considered a match, regardless of the tags’ associated values.

When *Include transfer components?* is checked, in addition to the components found based on the other search criteria, any components that are part of transfers involving the components found are also included in the search results. Direct and indirect transfers are included, similar to the results of a component’s graph display.

Results of a search are displayed in a table as follows:



Results of Search for ID 0098973031 with Include of Transfer Components

The resultant table is initially sorted by ID but can be sorted by another field by selecting the wanted table heading. To go to any of the found components’ display page simply click on the component ID.

# A very brief technical overview

At a high level Tracker is very simple. It consists of components, initial contents and a DAG (directed acyclical graph). Components have a universal id (e.g., a barcode) and a type (e.g., plate, tube or rack). Optionally a component can have initial contents (e.g., an antibody, a molecular barcode set or a sample set).

When one component is transferred to another the transfer is recorded in the DAG, which contains nodes that are components and edges that have source and destination component IDs. Having a source and destination makes the graph directed. The graph is maintained to be acyclical to insure a component is never transferred to itself, directly or indirectly.

To find the contents of a component, for example when making an EZPASS, walk-up sequencing pass or simply displaying the contents, the DAG is traversed. Specifically, the predecessors of the component are recursively examined to find all the initial contents that have been transferred, directly or indirectly, into the target component. The collection of all the initial contents becomes the contents of the target component.