

# ePrints Author ID Project

The ePrints Author Project is a USQ initiative lead by the Division of Academic Information Services (DAIS) and the Division of Information Communication and Technology Services (DICTS) to provide a method for supporting the unique identification of authors within the institutional repository powered by ePrints.

ePrints is an open source application which USQ runs to provide its institutional repository. ePrints is available at <http://eprints.usq.edu.au> and provides the world with access to the cutting edge research that is being undertaken by the University.

A major limitation within ePrints is the inability to truly identify a given author and their works. ePrints has a creator identification field but does not provide any support or validation of the information provided. This field is associated with the name of the creators of the paper which follows the citation style. This situation is less than preferable for a number of reasons. The first problem is that the suggested use of the field is to provide an email address – this is an identifier that might change over the life time of the entries and might not be useful later on. The second issue arises when we consider that the cited name might change. For the integrity of the generated citations, the cited name should be as it is in the paper. As such it is hard to normalise this information as this might lead to technically incorrect data (even though it is logically correct as it is the same person).

USQ's ePrints Author ID project aims to solve the gap that exists within this space. Author ID provides the ability to define a new data set called “authors” which handles the unique identification of authors. This field is then linked back to the ePrint creators field for a citation line and enables the distinct identification of authors within the system regardless of the identities or names that they were cited under for a given paper.

The Author ID project also took this a step further and provided a new system called “author instance”. Author instances are designed to link an author with a paper in an identifiable point in time. The idea is that the author record contains the base information about a person that shouldn't change, such as their actual name (as opposed to their cited name), their email address and any identifiers external to ePrints that can help to identify the author. The author instance record then contains details like what department the author is in at that point in time, or even their institution. Then over the life time of an author various instances are created which permits their contributions to be properly acknowledged over time.

The Author ID project is presently in an alpha stage and is being tested by USQ Library (DAIS) employees to gain feedback for improvements. Aspects of the system need to be enhanced before it is ready for production release. These are:

- A view of the author data set
- Improvements to the merge author and author instance interface
- Remove author instances
- Improve popup interface
- Workflow changes for automatically creating an author and author instance whilst editing an ePrint.

This document is a collection of notes that have been collated from email communications to people, user documentation written specifically for this document and information considered to be relevant both internal to USQ and external to USQ.

# Table of Contents

Background.....	3
Technical Notes.....	4
User Guide.....	5
Searching for Authors.....	6
Author Management.....	7
Creating a new author.....	7
Editing an author.....	8
Viewing an Author.....	9
Merging Authors.....	10
Removing an author.....	11
Managing Author Instances.....	12
Viewing author instances.....	12
Creating an author instance.....	13
Editing an author instance.....	13
Viewing Author Instances.....	15
Merging Author Instances.....	15
Removing Author Instances.....	16
ePrint Alterations.....	17
Developer Guide.....	18
Lookup Popup Windows.....	18
InputForm Field's.....	18
Itemref.....	19
Creator.....	19
DataSets.....	20
Author.....	20
Author Instance.....	20
MetaFields.....	21
Externalitemref.....	21
Namevalue.....	22

## Background

In 2009, USQ undertook a series of projects to complete reporting of research outcomes of the University to the Australian Federal Government called "ERA". USQ has a few different systems in place to store this information. We have ePrints which stores a large collection of our papers for areas and was a logical first point to gather information for the reporting. We also have another in house developed system called "RPCS" which is run by our Planning and Quality department which contains very similar information to what ePrints has but also has some of its own data. It also features the concept of a single author identifier which is important to them for reporting. As citations are important to ePrints and email addresses can unfortunately change (e.g. student to staff, email addresses of external organisations for staff who are technically visiting staff but retain their old address, etc) so using this as a unique identifier doesn't work. RPCS uses the staff identifier which works well for them and the majority of their reporting however such a system wouldn't work well for ePrints which can have staff and students submitting papers with papers having external collaborators.

As a part of all of this we are working to modifying ePrints in such a way that we can integrate all of the RPCS information into it and use it as a single source of information. As with all major goals it is going to take time for it to happen but we need the infrastructure in place. A important need for us is in fact working out who the person is (since the cited name might be different) and basically under which role they authored the paper. A common example is someone who works for a Faculty but is writing a paper for a research group at the University (and their time on that funded by that group not the Faculty). The paper might be attributed to another department but the authors contribution shouldn't be attributed to the Faculty they were in at the time but to the research group.

So we quickly develop the need for two structures: one which records the authoritative name (plus other details about the author) and one which stores the instance of the author. The system is flexible enough that you could just use authors and be done with it but for our purposes we wanted to link departments to papers and people, so we have the author instance. This also provides a means for building more interesting queries and metadata, for example you can then search by author and the department to see papers from a research group or from their work in the Faculty.

The author record I built by default has name, user ID, primary email address, alternate email address and an external identifier fields. The idea behind this is that we need a primary email address to pester people about things potentially (distinct from the user because they might not have a user account especially for external collaborators), we also would like to record alternate emails to make associating papers easy to handle and we would also like to know identifiers for the user external to ePrints. Examples of this are our internal student and staff identifiers (and in the future a combined identifier which is unique for a person regardless of role) or perhaps external identifiers for stuff like Scopus which has unique author IDs or their identifier at another institution.

## Technical Notes

The patch is made against ePrints 3.1.0 and appears to apply with a few offsets to 3.1.3 except for a change in the search.pl file that appears to be incompatible.

The system is divided into two major systems. The first are tools for managing authors. Authors have details such as a name, bio, email addresses and external (to ePrints) identifiers. All pretty simple and straight forward. There is also a system called author instance to handle "instances" of an author. This is useful for us more for reporting than anything else because the instance contains quasi-temporal information to link the author and the ePrint together. Examples of this are institution and department fields that could track an author's career as they shift within the institution or as they move outside of an institution. Both of these are created as datasets in ePrints.

I've also added a few features that I wanted to get things working. The first is the popup feature to handle light search windows where AJAX drop downs don't necessarily make sense. I've created popups for the user, author and author\_instance datasets. The next is two new metafields: a name value pair and a "external" itemref. The name value pair is what it seems and is used by author to handle the external ID system. The external ItemRef field type is similar to the itemref type but it is designed to look at non-dataset tables within the set. We're planning on using this for our department tree so that we can link author instances to given departments which have a historic aspect (some of the reporting we've got to do go back nearly a decade including reporting what logical faculty a subdepartment was in at the time). There are other uses potentially as it allows people to flexibly search and handle a side table without impacting heavily on ePrints. I've also added two new InputForm's as per your suggestion to handle altering the display of the creator and itemref fields to handle the popup window situation that I created.

# User Guide

The Author ID system adds a two new entries to the administrator panel. These are:

- Search authors (located above “Search users”)
- New Author (located below “Create user”)

## Admin

Status	- Status information for this server.
Search items	- Search for items in this repository.
Issues Search	- Search for issues discovered in current records.
Search authors	- Search for authors recorded in this repository.
Search users	- Search for users of this repository.
Search history	- Search for actions that have occurred on items in this repository.
Create user	- Create a new user account
New author	- Create a new author within ePrints to assign distinct citations to a single identity.
Stop Indexer	- Stops the full-text indexer.
Regenerate Abstracts	- Cause abstract pages to be regenerated when they are next requested.
Regenerate Views	- Cause views pages to be regenerated when they are next requested.
View Configuration	- View configuration files for this repository.
Send Test Email	- Test your email configuration by sending a test email.
Edit subject	- Modify the subject tree(s)
Manage Metadata Fields	- Add or remove metadata fields
Database Schema	- Show the database schema.

The system also adds an option to the ePrint item edit screen to handle look ups.

The Author ID system also provides various new capabilities for developers of ePrints including new metafields (ExternalItemRef, Name-Value), new datasets and a popup window capability.

## Searching for Authors

The administrator screen presents the option to search for authors.

**Search authors**

<input type="button" value="Search"/> <input type="button" value="Reset the form"/>		
<b>Author Name:</b>	<input type="text" value="all of"/> <input type="text"/>	<a href="#">About this field</a>
<b>Author Unique ID:</b>	<input type="text"/>	<a href="#">About this field</a>
<b>Author User ID:</b>	<input type="text"/>	<a href="#">About this field</a>
<b>Primary Email Address/Alternate Email Addresses:</b>	<input type="text" value="all of"/> <input type="text"/>	<a href="#">About this field</a>
<b>External IDs:</b>	<input type="text" value="all of"/> <input type="text"/>	<a href="#">About this field</a>
<b>Retrieved records must fulfill:</b>	<input type="text" value="all of these conditions"/>	
<b>Order the results:</b>	<input type="text" value="Order by author name"/>	
<input type="button" value="Search"/> <input type="button" value="Reset the form"/>		

This screen provides the ability to enter details about an author and search for them. For example one could create a search based on the user ID to find a user based on their user identifier. Selecting an author from the list will result in the “view author” screen being displayed for that given author.

**Author User ID is 365**

Displaying results 1 to 1 of 1. <a href="#">Refine search</a> <a href="#">New search</a>	
Order the results:	<input type="text" value="Order by author name"/> <input type="button" value="Reorder"/>
Export 1 results as	<input type="text" value="EP3 XML"/> <input type="button" value="Export"/>
Moffatt, Sam	
Displaying results 1 to 1 of 1. <a href="#">Refine search</a> <a href="#">New search</a>	
Order the results:	<input type="text" value="Order by author name"/> <input type="button" value="Reorder"/>

See the “Author Management” section about the “View Author” screen.

## ***Author Management***

Authors are a new dataset provided by this patch which enables ePrints administrators the ability to uniquely define authors details regardless of the cited form on their paper. Author management provides methods for creating, editing, merging and deleting authors. Optionally the system provides the ability to define 'author instances' which provide the ability to have define specific instances of a given author. Author instances are discussed in the section entitle “Author Instance Management”.

### **Creating a new author**

To create a new author select “new author”. In the screen that comes up enter the author's preferred name:

**New author**

Add a new author to the system by specifying their name details.

Given Name:

Family Name:

The details entered in this screen will then be taken and populated into the edit author screen.

## Editing an author

### Edit Author

<input type="button" value="Cancel edit"/> <input type="button" value="Save author"/>			
<b>Author Name</b> <span style="float: right;"><a href="#">About this field</a></span>			
Title	Family Name	Given Name / Initials	
<input type="text"/>	<input type="text" value="Moffatt"/>	<input type="text" value="Sam"/>	
<b>Author User ID</b> <span style="float: right;"><a href="#">About this field</a></span>			
<input type="text" value="365"/>	Mr Sam Moffatt	<input type="button" value="Lookup"/> <input type="button" value="Find entry"/>	
<b>Primary Email Address</b> <span style="float: right;"><a href="#">About this field</a></span>			
<input type="text"/>			
<b>Alternate Email Addresses</b> <span style="float: right;"><a href="#">About this field</a></span>			
1.	<input type="text"/>	▼	
2.	<input type="text"/>	▼ ▲	
3.	<input type="text"/>	▼ ▲	
<input type="button" value="More input rows"/>			
<b>External IDs</b> <span style="float: right;"><a href="#">About this field</a></span>			
	Name	Value	
1.	<input type="text"/>	<input type="text"/>	▼
2.	<input type="text"/>	<input type="text"/>	▼ ▲
3.	<input type="text"/>	<input type="text"/>	▼ ▲
<input type="button" value="More input rows"/>			
<b>Author Biography</b> <span style="float: right;"><a href="#">About this field</a></span>			
<input type="text"/>			
<input type="button" value="Cancel edit"/> <input type="button" value="Save author"/>			

This screen enables the user to enter information about the user. The first line is the author's name, prefilled from the initial screen. The next is an ItemRef link to the ePrint's user dataset to enable an author to be linked to a user account. This is a standard ItemRef with the ability to use a popup window to aide the discovery of the user ID.

The primary email address is provided in a single text field to aide extraction of the information and integration with external systems (such as reporting and notification). Placing this field here enables the system to retrieve the information with the author quickly. The alternate email addresses are provided to enable the system to record email addresses that the author might alternatively user and



should be known by. When the primary email address is updated it places the old primary address automatically into the alternate email address field. Once an email is deleted from the alternate email address field though it is removed from the system.

The next section is the “External IDs” section. These are name-value pairs that are designed to allow the system to very flexibly store identifiers external to the system. Examples of some uses are storing the author's staff identifier and their student identifier or even storing external identifiers such as Scopus' Unique Author Identifier.

The last field is the author biography field. This field was added for comment and has generally been regarded as pointless. At one point the idea was that the biography might appear at the top of a page listing their papers. The future of this field is uncertain.

Once the details have been filled in, the author can be saved to the database. At this point you will be presented with the author screen which enables a variety of options for handling the user.

## Viewing an Author

The view author screen presents options with regards to handling an author and their instances. The options directly available include:

- merge authors
- list author instances
- create author instance
- edit author
- remove author

### View Author

Moffatt, Sam

Merge authors	- Merge another author into this ones
List Author Instances	- View specific instances of this author
Create author instance	- Create a new instance identifier for an author
Edit Author	- Edit an authors detail
Remove author	- Remove this author from the system

The merge author option takes two authors and merges them into a single author, the list author instances screen lists the instances that this author has associated with their record, the “create author instance” option permits the creation of a new author instance, the edit author screen presents the ability to edit the details of the author in the same screen that author creation utilises and the remove author option permits the permanent removal of the user from the system.

## Merging Authors

The merge author option provides the ability for duplicate author records of the same person to be merged into a single author record. This process also re-links the author instance records from the old author to this author.

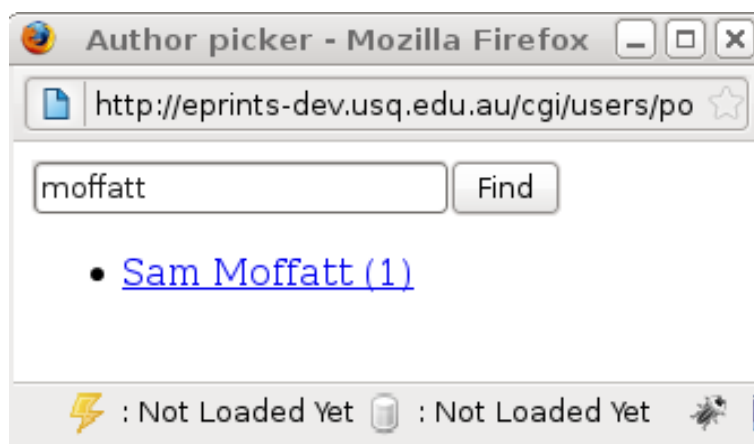
### Merge authors

Please select the other author that you wish to merge into this author entry. Note that this row will remain and the other entry will have instances repointed to this one. The next screen will allow you to control the final values.

Primary author: [Moffatt, Sam](#)

Duplicate author ID:

The primary author is the record that will continue to exist and is taken to be the correct version (used as the defaults in the next screen). The duplicate author's record will be deleted and any author instances associated with that record linked to the primary author. The "Find Author" button searches for an author record, using the value of the duplicate author ID field (if set) as the basis of a search. In the popup window you can select the author and it will put the author's ID into the duplicate author ID field.



Once the duplicate author has been select you can start to merge the authors:

### Merge authors

This screen permits you to merge the two author records. The base record is the default value. The values of the base records and the duplicate are displayed for reference.

Primary author: **Moffatt, Sam**

Duplicate author: **screen, greater**

bio			<input type="text"/>
author	Moffatt, Sam	screen, greater	<input type="text"/> <input type="text" value="Moffatt"/> <input type="text" value="Sam"/>
userid	Mr Sam Moffatt	Ms Amy Charney	<input type="text" value="365"/> Mr Sam Moffatt
primary_email	moffatts@usq.edu.au		<input type="text" value="moffatts@usq.edu.au"/>

The merge author screen presents a list of the single value fields in the system and provides the ability to integrate them. The interface is presently very horribly lacking unfortunately. The default values are populated from the primary author and both the primary author and duplicate author is presented in two columns before the fields, first the primary author and then the duplicate author. The values of the fields are taken to be the new values.

Multiple entry fields, such as the external ID's and alternate emails, are automatically merged. Duplicates may be created and have to be removed by hand (this is a conscious decision). These fields don't appear in this screen.

After merging the author you will be returned to the "view author" screen of the primary author. At this point it is strongly encouraged to edit the author and review the multiple entry fields.

### Removing an author

An author record can be removed by two processes: by being merged into another record (the old record is deleted) or by being explicitly deleted from the author view screen. A simple screen is displayed confirming that the author will be removed and then the author will be removed.

### Remove author

Are you sure you want to delete this author? This operation cannot be undone!

The system will redirect to the ePrint item list and display the results of the operation.

## Managing Author Instances

Author instances are specific instances of a given author that are designed to store information about the user that might be temporal or be parallel and disjunct.

In a simple example, an author begins work at the University in the Faculty of Science. The author produces some papers whilst they are in this faculty and thus have an author record creator and an author instance created for this faculty. The author then shifts to the Faculty of Education and creates a new paper whilst in this faculty. Their author record remains in tact as none of the information in it has actually changed (e.g. their name has remained constant as has their email address) however a new author instance record is created to match the Faculty of Education. Whilst in the Faculty of Education the author works on a paper with a few other staff members from the Faculty of Arts. In this case their time was funded by the Faculty of Education however the paper comes from the Faculty of Arts. The author is linked to the paper with the Education author instance even though the paper is majority written by Faculty of Arts staff and the eprint record shows this fact. In an alternate situation the author is funded by the Faculty of Arts and thus has another author instance recorded created parallel to their Education record (by default instance records only record creation date and have no end date) and their contribution to the paper is associated with their Arts author instance.

In this way ePrints can flexibly record information about contributions made by staff within the University as they shift over time. Funding is often allocated on the basis of research completed by staff within distinct faculties and this permits this information to be easily recorded.

Author instances, like authors, are a full blown ePrints data set. This means that repositories can override the default fields provided by both of these data sets to add information to either sets of records.




## Viewing author instances

The “list author instances” screen provides the ability to review the instances created for that author. This is only accessible by viewing an author.

**List Author Instances**

About this field [Author Instances](#)

[Create author instance](#)

Author Instance Unique Identifier	Author Unique Identifier	Instance Creation Date	Instance Display Name	
46	Moffatt, Sam	4 August 2009 11:51:31 +10:00	Faculty of Science	 <a href="#">Merge author instances</a>
48	Moffatt, Sam	10 August 2009 15:18:51 +10:00	Ples	 <a href="#">Merge author instances</a>
49	Moffatt, Sam	10 August 2009 15:19:01 +10:00	Division of ICT Services	 <a href="#">Merge author instances</a>

Preferred Author Instance  [Add Column](#)

The screen lists the current author instances, their unique identifiers, their author user, the date it was created and the display name of the author. The final panel provides options for the instance such as editing that instance or using the instance as the primary instance for a merge.

## Creating an author instance

An author instance can be created on either the “view author” screen or the “list author instances” screen. Creating an author requires an author ID however this is usually pre-populated from the previous screen.

### Create author instance

Enter the author ID of the author you wish to create a new instance.

Author ID:

Once an author instance is created, the author instance editor will be selected.

## Editing an author instance

Author instances can be edited using a familiar interface.

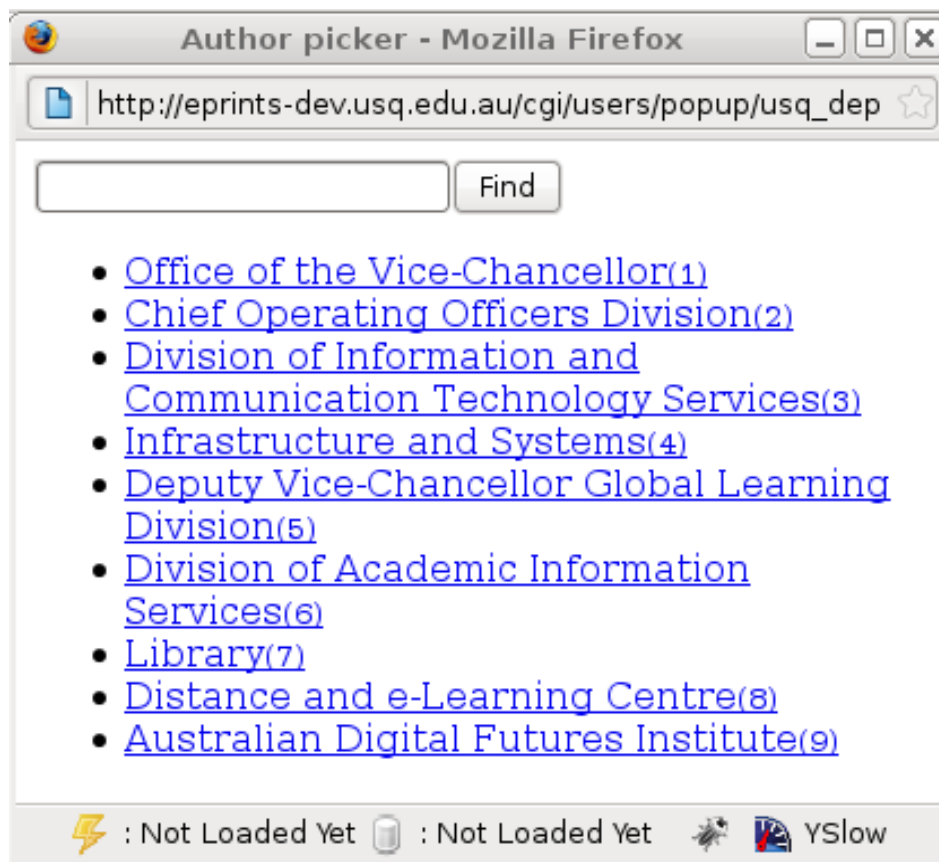
### Edit Author Instance

<input type="button" value="Cancel"/> <input type="button" value="Save"/>	
<b>★ Author Unique Identifier</b>	<a href="#">About this field</a>
<input type="text" value="1"/> <span style="margin-left: 20px;">Moffatt, Sam</span>	<input type="button" value="Lookup"/> <input type="button" value="Find entry"/>
<b>Instance Display Name</b>	<a href="#">About this field</a>
<input type="text" value="Faculty of Science"/>	
<b>Preferred Author Instance</b>	<a href="#">About this field</a>
<input type="checkbox"/>	
<b>["author_instance_fieldname_deptid" not defined]</b>	
<input type="text"/>	<input type="button" value="Lookup"/> <input type="button" value="Find entry"/>
<input type="button" value="Cancel"/> <input type="button" value="Save"/>	

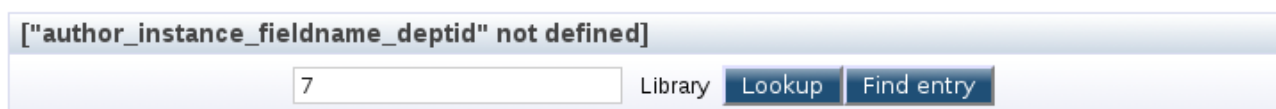
This interface features the base fields: author unique identifier, display name and preferred flag. The unique identifier links this author instance to a given author. The display name is used to provide a handle on the given instance and is presented in the author instance list for an author. The preferred author instance flag displays that this author instance should be used over any other instance. The preferred flag is a mutually exclusive flag, setting this on an author instance will unset it from the previous instance.

In this screen shot there is also a custom field called “department ID”. This field is a new field type called “External Item Reference” (ExternalItemRef) which links a field in ePrints to an external data source in a way similar to how an ItemRef which links ePrints data sets together.

The lookup buttons validates the contents of the field and populates it with a value. This is true of both author unique identifier and the custom department ID field. The “Find Entry” option is a new ability which pops up with a new window to select an author or a department respectively. The author picker is used for the author unique identifier field and a custom USQ department picker is used for the department field:



When the department has been selected, its ID will be copied into the field and the page will be updated to reflect this new value:



Saving the author instance will commit the changes to the database.

## Viewing Author Instances

Author instances, like other items, have a view screen that presents a selection of options as well as the display name of the instance and the author record that the instance is linked to. The screen presents you with the ability to edit the given instance

### View author instance

School of Law

Moffatt, Sam

Edit Author Instance

- Provide details about this particular author instance

Merge author instances

- Merge another author instance into this one

## Merging Author Instances

Like authors, author instances can be merged to remove duplicate entries. Unlike merging authors, merging author instances doesn't have any side effect beyond removing the duplicate record. This causes a consistency issue potentially as legacy ePrint creator records aren't updated to point to the new ID. A solution to this problem hasn't been worked out as of yet.

### Merge author instances

Cancel merge

Merge instances

Please select the other author that you wish to merge into this author entry. Note that this row will remain and the other entry will have instances repointed to this one. The next screen will allow you to control the final values.

Primary instance: [Faculty of Science](#)

Duplicate instance ID:

Find instance

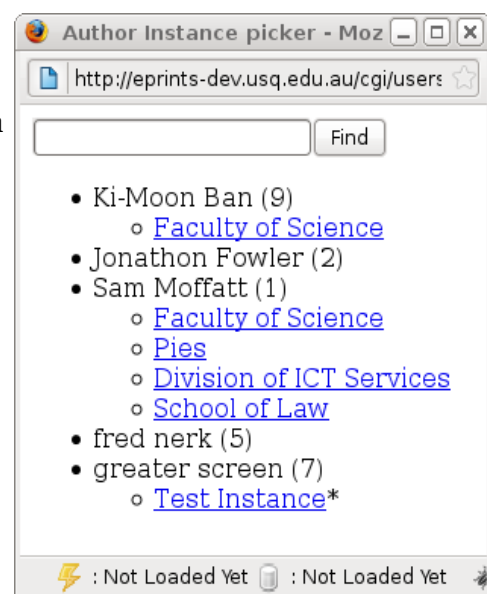
Cancel merge

Merge instances

This interface is very similar to what is displayed for an author except the term 'instance' is used instead.

The "Find Instance" button provides the ability to display a popup window tuned to author instances (unfortunately it presently doesn't respect the author ID):

The asterisk (\*) denotes that the given instance is the preferred instance for that author. The popup is a generic popup and some work will need to be done to provide an interface that is better for merging authors. As with the author and department popups presented earlier, selecting the instance fills in the relevant author instance ID into the duplicate ID field.



The Merge author instance screens is also similar to the set up of the author merge screen. It presents a list of fields and their values with the default values taken from the primary instance. As with authors, the primary instance is located in the first column, the duplicate instance is in the second column and the fields whose values will be stored are located in the final field.

### Merge author instances

Cancel merge
Merge instances

This screen permits you to merge the two author instance records. The base record is the default value and will be retained. The values of the base records and the duplicate are displayed for reference. In most instances the Author ID should not be changed, you should merge the authors first or alter the individual author instance record.

Primary instance: [Faculty of Science](#)  
Duplicate instance: [Pies](#)

authorid	Moffatt, Moffatt, Sam Sam		<input type="text" value="1"/>
display_name	Faculty of Science Pies		<input type="text" value="Faculty of Science"/>
preferred	No No		<input type="checkbox"/>
created_date	4 August 2009 11:51:31 +10:00	10 August 2009 15:18:51 +10:00	Year: <input type="text" value="2009"/> Month: <input type="text" value="August"/> Day: <input type="text" value="04"/> Hour: <input type="text" value="01"/> Minute: <input type="text" value="51"/> Second: <input type="text" value="31"/>
deptid	Library	Division of Information and Communication Technology Services Library	<input type="text" value="7"/>

Cancel merge
Merge instances

As with authors, any fields that are multiple entry fields (none by default in author instances) will be automatically merged. Duplicates in multiple entry fields may occur and should be manually removed. Editing the author instance after it has been merged to validate its consistency is again strongly encouraged.

### Removing Author Instances

At the present moment there is no interface to remove author instances. This is a slight oversight that will be resolved later. An author instance can be merged out of existence however this results in an author instance still existing.



## ePrint Alterations

The ePrint edit screen has been altered to provide the ability to easily find an entry. The addition is the use of the “Find Entry” button which is ubiquitous throughout the rest of the author ID work. As with other aspects of the system it presents a popup dialogue which provides an option to select the author instance. This popup is the same as the popup that appears when merging two author instances together.

The screenshot displays the 'Authors' edit interface. It features a table with columns for 'Family Name', 'Given Name / Initials', and 'Authors email (if known)'. The first row contains 'Lane' and 'Andrew M.'. The second row contains 'Terry' and 'Peter C.'. Below the table is a 'More input rows' button. To the right of the table are four 'Find entry' buttons. A popup window titled 'Author Instance picker - Mc' is open, showing a search field with 'lane' and a 'Find' button. The search results list 'Andrew M. Lane (13)' with a sub-entry 'Faculty of Arts\*'. The URL 'http://eprints-dev.usq.edu.au/cgi/use' is visible in the popup's address bar. There are also 'About this field' links on the right side of the interface.

As with the other popups, selecting the particular instance will populate the appropriate field. Unlike other edit screens where this style interface is used, the instance display name will not be displayed however selecting find entry will automatically populate the search field with the instance ID and usually display the correct value (or a similar match).

## Developer Guide

The Author ID patch provides numerous features that might be useful for developers of the ePrints system. This comes in the form of popups, InputForm Field's, new DataSets and new MetaFields.

### ***Lookup Popup Windows***

The Author ID system makes use of some new popup windows. These are used to easily track down information in a larger data set where an AJAX call might not be the best solution to the problem. They are designed to behave in a way similar to the core AJAX code but provide easier functionality for the system overall.

The Author ID patch provides three popup windows that serve as examples for trivial implementations of further popups. The popups provided are for users, authors and author instances. Each of these provide the ability to feed them search terms which are used in their individual search system.

Each popup requires a selection of parameters to work properly. These include the "fieldname" of the field that the value should be located in once a link is clicked, the base name of the form (required only when the form should be validated), a flag called "validate" to control if the form should be validated (set `validate=true` in the GET params to validate automatically) and a search term input (dependent on the actual field).

The system provides two new JavaScript functions to support popup windows. These are "popup" and "popup\_appendfield". The "popup" function accepts a URL to display, a width and a height of the window. It displays the windows with the given height and width as well as focusing the window. The "popup\_appendfield" function accepts a fieldname, URL, width and height. The fieldname is an ePrints field name where it grabs the element by the ID, appends its value to the given URL and then calls popup with this newly constructed URL and the passed width and height. This is useful for creating a popup which has the final part of its URL as a search term and populating this from the on screen form field.

In most cases it isn't required to call these directly. The code to work out what should be passed to them is encapsulated in the InputForm Field's "ItemRef" and "Creator" which are included with the patch.

### ***InputForm Field's***

The Author ID patch adds two new InputForm Field's: Itemref and Creator. These input forms extend the built in Itemref output and the ePrint item creator output respectively to provide integration with the popup windows that are used throughout the Author ID system.

Both fields make use of two new attributes in the workflow file: *external\_lookup\_url* and *external\_lookup\_params*. The *external\_lookup\_url* designates where in the system the popup exists (it might be hosted on another server for example) that the system should direct the user to and the

*external\_lookup\_params* provides the values expected. See the individual field's for sample examples of this in action.

## Itemref

The Itemref field's purpose is to render the standard Itemref field with a new “Find Entry” button next to it. The prime aim of this is to alter the output of the core ePrint's field without having to alter the core of ePrints.

```
<component type="Field::Itemref">
  <field ref="userid"
  external_lookup_url="{config{perl_url}}/users/popup/user"
  external_lookup_params="userid=" />
</component>
```

In this example it is calling the default user selection popup from within ePrints with the “userid” param used to search.

```
<component type="Field::Itemref">
  <field ref="authorid"
  external_lookup_url="{config{perl_url}}/users/popup/author"
  external_lookup_params="authorid=" />
</component>
```

In this example it uses the provided author popup with the “authorid” param being the search term.

## Creator

The Creator field's purpose is to provide an override in the ePrint workflow to permit the easy selection of an author or author instance based on the

```
<component type="Field::Creator">
  <field ref="creators" required="yes"
  input_lookup_params="eprintid={eprintid}"
  external_lookup_url="{config{perl_url}}/users/popup/author_instance"
  external_lookup_params="authorid=" />
</component>
```

This simple change tells ePrints to use the custom “Field::Creator” class when rendering this field and adds the “Find Entry” button to the output. The *external\_lookup\_url* and *external\_lookup\_params* are configuration items which control where the popup goes to (see above). In this case it is pointing to the *author\_instance* popup which displays a list of available author instances. This could easily be change to just the author popup if there were no plans to use authors.

## **DataSets**

The Author ID patch creates two new ePrints datasets called “Authors” and “Author Instances”. These are the storage system behind the new functionality added in the system. As both of these are proper full blown ePrints datasets, any repository local customisation can be completed on these datasets similar to how customisation of the core datasets and their fields can be completed.

## **Author**

The Author dataset is designed to store long term information about an author that doesn't necessarily change or should be recorded statically. For example these include items like their name, their email address, any other email addresses and any external identifiers for the user. The information in this table is designed to not be heavily edited or changed over its life time and should be capable of being safely reused as required.

The default fields in the authors data set are:

- Author ID (primary key; uneditable)
- Author Name
- ePrints user ID (ItemRef link to the ePrints user data set)
- Primary Email Address
- Alternate Email Address (Multiple entry field)
- External Identifiers (Name-Value pair multiple entry field; see MetaFields for more information about the Name-Value pair field type)
- Biography

Users of the system are welcome to expand these fields with their own repository local customisations.

## **Author Instance**

The Author Instance records the particular “instances” of an author as they progress through their life. This dataset could be considered a child dataset of the Author system and is intrinsically linked back to an author record. The author instance table is aimed at storing information that progressively changes in individual subrecords of the author. Author instances thus allow the recording of how information changes and the ability to distinctly attribute papers to instances regardless of what other instances might be available.

The default fields in the author instance table are:

- Author Instance ID (primary key; uneditable)
- Author ID (ItemRef link to the ePrints eprint data set)
- Display Name
- Creation date

- Preferred flag

The preferred flag is managed by the author instance data object which ensures that the preferred flag is mutually set for a given author instance record of the author.

As with the Author dataset, this is a full blown data set which permits the addition of custom fields. One such custom fields is “deptid” which is used by USQ to link instances back to departments within our department tree. This is done to permit the association of paper submission with given departments for funding purposes.

## **MetaFields**

The Author ID patch creates two new metafields which are used by the system to support the storage of new data formats. The fields added are the Externalitemref field and the Namevalue field. The Externalitemref field is an extension of the ItemRef field but designed to handle simple tables not managed by ePrints. The Namevalue field is used by the author records to flexibly store the External Identifiers in name-value pairs.

## **Externalitemref**

The Externalitemref field, or External Item Reference, uses the ePrints MySQL connection to provide integration into external tables. It requires the specification of an externaltable, externalkey and externalname. Optionally it supports the external\_lookup\_url and external\_lookup\_params for integration of a popup window using similar values in these fields to the InputForm field's.

The externaltable option specifies a table that ePrints should examine to source its data from. The externalkey option specifies the primary key of the table. This should be the field that contains the value supplied by the system. The least option is the externalname field which should match the field which stores the name of the record. These are then looked up and put together to provide the value and the name next to the field similar to how the built-in Itemref field behaves.

Whilst not shipped with the default system, an external item ref forms a part of the USQ prototype interface. Here is a sample from the author\_instance\_fields.pl (in your archive configuration directory, /opt/eprints3/archivename/cfg/cfg.d/author\_instance\_fields.pl):

```
$c->{fields}->{author_instance} = [  
    {  
        'name' => 'deptid',  
        'type' => 'externalitemref',  
        'externaltable' => 'usq_dept_tree',  
        'externalkey' => 'deptid',  
        'externalname' => 'dept_name',  
    },  
];
```

This example demonstrates a new field named 'deptid' which is an Externalitemref. It uses the

'usq\_dept\_tree' table and the primary key of this table is 'deptid'. A textual description of the row is stored in the 'dept\_name' field.

## **Namevalue**

The Namevalue field is designed to provide a simple name-value pair field that can be used to flexible store keyed data. It is a simple field which provides flexible input methods. It doesn't provide many points for configuration beyond the standard configuration options (e.g. multiple, index, etc).