Orpheus

Manual for version 1.0

André Sartori

University of Cambridge
Office of Scholarly Communication

March 12, 2019

Contents

1 Introduction

Orpheus is a database of academic journals and their policies written in Django. It features web frontends for users and administrators and a RESTful API, tailored for integration with DSpace and other institutional repository platforms.

2 Installation

2.1 Obtaining a copy of Orpheus

You may either download and extract a zip file from https://github.com/osc-cam/orpheus/archive/master.zip or clone the git repository by issuing the following command in your UNIX terminal:

```
$ git clone https://github.com/osc-cam/orpheus.git
```

2.2 Setting up a database backend

Django supports many different database backends[1], which may or may not work with Orpheus out of the box. Orpheus was developed and has only been tested using a PostgreSQL backend, so this is the recommended database engine.

---

[1]see https://docs.djangoproject.com/en/1.11/ref/databases for details and notes

2

Once you have created an empty database in the engine of your choice[2], you will need to edit the value of DATABASES in file `orpheus/settings_local_example.py` to match the parameters of the database you have just created.

While you are editing this file, you may also wish to change the value of SECRET_KEY. If your instance of Orpheus will be reachable from outside of your local network, you should review your settings before deployment[3].

Once you have completed this, rename the file `orpheus/settings_local_example.py` to `orpheus/settings_local.py`

2.3 Creating and activating a virtual environment (optional, but recommended)

To create and activate new virtual environment for Orpheus, issue the following commands:

```
$ python3 -m venv orpheus-env
$ source orpheus-env/bin/activate
```

2.4 Installing the dependencies

To install all Orpheus dependencies, navigate to the folder containing the code you downloaded or cloned from GitHub and issue the command:

```
$ pip3 install -r requirements.txt
```

2.5 Applying migrations

To create the database fields that Orpheus will need to store its data, apply migrations by running:

```
$ python3 manage.py migrate
```

2.6 Creating a superuser

You will need at least one user who can login to Orpheus, so run the following command to create it:

---

[2]if you decide to use PostgreSQL and you are running Ubuntu or derivatives, a good guide on configuring the database is available at `https://www.codeproject.com/Articles/898303/Installing-and-Configuring-PostgreSQL-on-Linux-Min`. For other databases and operational systems, Google is your friend!

[3]see `https://docs.djangoproject.com/en/1.11/howto/deployment/checklist` for recommended settings

```
$ python3 manage.py createsuperuser
```

2.7 Testing your Installation

You should now have a functional development instance of Orpheus! To test it, issue the command:

```
$ python3 manage.py runserver
```

And then point your favourite browser to http://127.0.0.1:8000/. You should see Orpheus' homepage.

2.8 Populating auxiliary tables

Some of Orpheus' tables include fields that are populated using values stored in four small, auxiliary tables (*Licence*, *Outlet*, *Responsibility* and *Version*). Unless these four tables contain at least one value each, some of forms in the frontend for editors will not work.

To populate these four tables you may either use the frontend for administrators or you may import the JSON files that come with the Orpheus source code by issuing the command:[4]

```
$ python3 manage.py loaddata Licence Outlet \
    Responsibility Version
```

2.9 Importing some data

Unless you intend to populate the database manually, you will now probably want to bulk import some data into your Orpheus instance. The `connectors` folder of the Orpheus source code contains a number of Python parsers that can be used to import large datasets made available by a number of publishers and projects. Editing and running these parsers is the recommended method of bulk populating and curating the database.

However, if you just want to load as much data as possible to your Orpheus instance for testing purposes, you may import JSON files providing a recent snapshot of most of the database hosted by the University of Cambridge. You can do so by issuing the command (it will take a while to complete):

```
$ python3 manage.py loaddata Source Node Deal Epmc \
    GoldPolicy GreenPolicy OaStatus
```

---

[4]for more details, see https://docs.djangoproject.com/en/1.11/howto/initial-data/
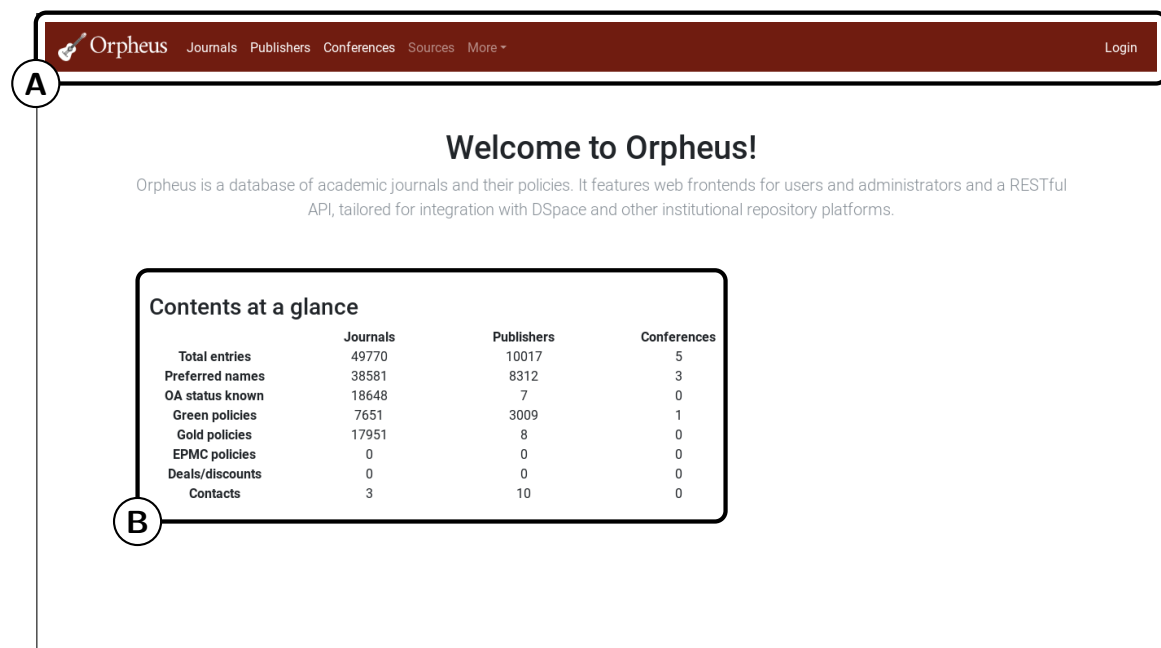
Figure 1: The main elements of the homepage. **A**, navigation bar; **B**, contents at a glance table.

3 Frontend for editors

The Orpheus web frontend for editors is available at https://orpheus-prod.lib.cam.ac.uk. From the *homepage*, users can use the **navigation bar** (Fig. 1A) to navigate to one of the *list views* described in section 3.1 on the following page. The *homepage* also summarises the contents of database in a "contents at a glance" table (Fig. 1B).

> 💡 **Sandbox**
>
> A sandbox for testing and familiarising yourself with the database is available at https://orpheus-dev.lib.cam.ac.uk
>
> To login to the sandbox, please use the following credentials:
> username: `test_curator`
> password: `Orpheus_dev`

3.1 List views

List views allow users to browse and search for specific entries in the Orpheus database (Fig. 2 on page 9). For instance, *Journals*, *Publishers* and *Conferences* are list views displaying records of those entities.

The **search box** on the top right corner of a *list view* page can be used to search for records containing a particular word or string (Fig. 2A), and the **paginator** at the bottom of a *list view* can be used to browse records (Fig. 2D). List views also include an action button *add new*, positioned immediately to the right of the list heading. When clicked, this button takes the user to a web form where they can add a new entry.

A brief description of each available *list view* is provided below.

3.1.1 General list views

Journals   Lists all journal records in the database, in alphabetical order.

Publishers   Lists all publisher records in the database, in alphabetical order.

Conferences   Lists all conference records in the database, in alphabetical order.

Sources   Lists all source records in the database, in the order they were created.

3.1.2 Curatorial list views

Change to alphabetical?

Orpheus comes with a number of lists views intended to facilitate the detection of problems and curation of the database. You will find these views under the **navigation bar** item *More*.

Problematic policies   Lists Orpheus records (journals, publishers or conferences) that contain one or more policies flagged as problematic (checkbox *problematic* ticked). This view is sorted by descending number of children records linked to the listed records, so policies imported from RoMEO that apply to hundreds of journals will be listed at the top.

Children of synonyms Lists Orpheus records in which the *name status* field of the parent record is not set to *preferred name*. In other words, it will typically list journals that are linked to a synonym of a publisher (e.g. OUP), rather thant the publisher's preferred name (Oxford University Press). Ideally, the parent of an Orpheus records should always be a preferred name.

Children of non-publisher nodes Lists Orpheus records in which the *type* field of the parent record is not set to *publisher*. In other words, it will typically list journals that are catalogued as children of other journals rather than of a publisher.

Grandchildren Lists Orpheus records whose parent record has a parent record. Although having grandparents could be useful to express, for example, relationships between journals, imprints and publishers (*Cell* is published by *Cell Press*, an imprint of *Elsevier*), Orpheus' logic for policy inheritance does not currently support inheritance from grandparents. The negative impact on database performance of implementing another level of inheritance is probably not worth the potential benefit.

Synonyms of synonyms Lists Orpheus records whose *synonym of* field points to a journal/publisher record in which the *name status* field is not set to *preferred name*. In other words, it will typically list journals that are deemed synonyms of other synonyms, rather than being linked directly to the preferred name.

Synonyms with linked policies Lists Orpheus records containing one or more attached policies and whose *name status* field is not set to *preferred name*.

3.1.3 Search box behaviour

Search boxes in Orpheus may be used to restrict list views to items containing a word or phrase of interest. The search engine is case insensitive, so both *System analysis and applied information science* and *ACS Biomaterials Science and Engineering* will be included in the results of a search for the string "science". Strings entered in the search box are evaluated as a whole. Thus, *ACS Biomaterials Science and Engineering* will *not* be included in the results of a search for "Science Engineering".

> 💡 **Searching by ISSN**
>
> The list views *Journals*, *Publishers* and *Conferences* also support searches by ISSN. This can be particularly useful to quickly find journal titles consisting of short and common words, such as "Science" and "Nature".

> ❗ Be careful with leading or trailing spaces when entering a string into the search box. These will be included in the search, so if you search for "Nature ", you will *not* find the journal "Nature" (but you will find "Nature Astronomy").

3.2 Detail views

If a user clicks on a record in a *list view* they are taken to the *detail view* of that record. A *detail view* displays most or all data available in Orpheus for a particular record. Figure 3 on page 10 shows the detail view of *Advanced Materials*.

3.3 Quality indicators

Orpheus has a number of fields indicating the quality of data for every journal/publisher name and policy in the database:

- *Vetted*: Indicates that data was created or modified by the an editor. All data entered or edited using the Orpheus web frontend has this field ticked by default.

- *Vetted date*: Automatic timestamp recording the date a record or policy was first vetted by an editor.

- *Last checked*: Date a policy was last checked by an editor.

- *Superseded*: Indicates that policy is no longer valid (it was probably superseded by a revised policy).

- *Superseded date*: Date an editor first realised that policy is no longer valid.

- *Problematic*: Indicates ambiguous/unclear policies requiring discussion/revision.

Orpheus uses the fields above to provide users with an icon that visually indicates the curation quality of data. These icons precede the main heading of each record and policy in the database (see Fig. 7 on page 17 for an example of the placement) and their meaning is:

Figure 2: Elements of an Orpheus *list view*. **A**, search box; **B**, "add new" action button; **C**, list of entries (journals in this case), showing up to 50 records; **D**, paginator.

Figure 3: Bird-eye view of the *detail view* of a journal. See section 3.2 on page 8 for an explanation of the different elements.

- ⊗: Policy should not be trusted. It is either superseded or the field *problematic* is checked.

- ❓: Policy has not been vetted by an editor yet, but it can be trusted. All records and policies created during the initial bulk population of the database (see section 5.1 on page 13 for details) were at that point marked with this icon.

- ✓: Policy was vetted by editor it is not superseded nor marked as problematic. This is the highest level of quality implemented in Orpheus.

> **!** Note that an orange question mark image (❓) is also used in Orpheus, but this is *not* a quality indicator. Instead, it indicates that field *version embargo months* is empty (unknown), just to avoid confusion with absence of embargo (*version embargo months*=0).

4 Frontend for administrators

The Orpheus web frontend for administrators, available at https://orpheus-prod.lib.cam.ac.uk/admin, provides tools for user management and editing of database tables that cannot be altered by editors (Fig. 4 on the next page). For example, new values for the tables *Licences* and *Outlets*, which are used to populate value lists in forms used by editors, can be added via this interface.

> **!** Be careful when editing existing values of valuelist tables via the admin interface. Some of the values are used by API methods and altering these could break the Orpheus API. Remember that with great power comes great responsibility.

5 Strategy for quickly expanding coverage of self-archiving policies

Increasing Orpheus coverage of self-archiving policies pertaining to accepted versions deposited in institutional repositories is paramount to achieving high-quality Apollo deposits with the correct embargo period applied.

This section provides an overview of the current state of the Orpheus database regarding self-archiving (green) policies and explains how we can most efficiently use and edit Orpheus to quickly populate it with reliable embargo periods.

Figure 4: Orpheus frontend for administrators.

5.1 Pre-populated database

The Orpheus' database was pre-populated in August 2018 by importing a number of policies datasets from:

1. Taylor & Francis

2. Version 14 of Andrew Gray's "Elsevier embargo periods, 2013-2018"

3. Elsevier's OA Price List

4. Wiley's "Author Compliance tool"

5. Cambridge University Press

6. Oxford University Press

7. Directory of Open Access Journals

8. Sherpa RoMEO

With the exception of the Sherpa RoMEO dataset, all these data sources contain data that can be easily parsed and understood by a computer, and hence data was linked to individual journals, resulting in 18,634 journals with a known OA status (i.e. categorised as fully Open Access, hybrid or subscription), 7,646 with a known embargo period and 17,939 journals with a known APC value and/or gold OA licence (Table 1 on page 15).

RoMEO data could not be reliably interpreted by a computer but RoMEO contains only 3,006 self-archiving policies, each of which linked to all applicable journals. Hence, these 3,006 policies were imported into Orpheus at the Publisher level and marked as requiring editorial revision ('problematic' field ticked). *Once revision of these policies is completed, Orpheus will be able to provide reliable embargo information to Apollo for all 38,577 journals in the database, rather than the 7,646 currently covered.*

5.2 General use and editing workflow

Figure 5 on the next page maps the workflow for general use and curation of Orpheus. Members of the Open Access team making decisions on funding should use Orpheus as the first resouce to find out a journal's OA status, embargo period, APC value and any other relevant journal policy.

If a record for the queried journal is not found in Orpheus (please bear in mind that the journal might be represented by a synonym), proceed as we have done in the past and search publisher websites, SHERPA RoMEO and other external
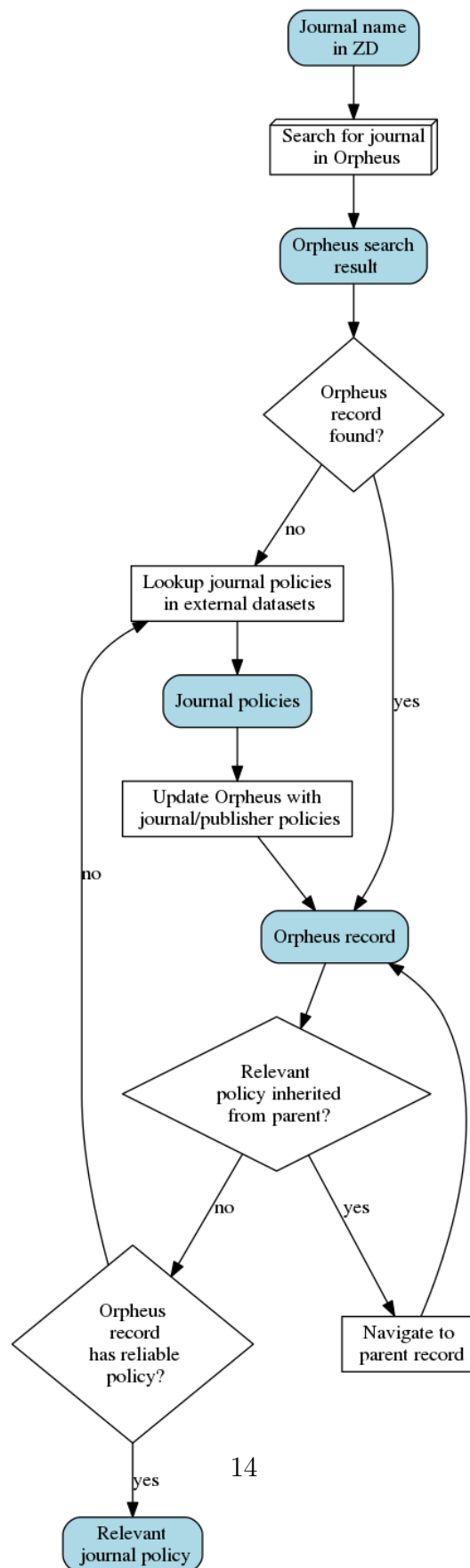
Figure 5: Workflow for integrating Orpheus curation and RCUK/COAF funding decision.

Table 1: Overview of Orpheus database following bulk imports

	Journals	Publishers
Total entries	49751	10013
Preferred names	38577	8309
OA status known	18634	0
Green policies	7646	3006
Gold policies	17939	0

sources instead. Once you have found data about the missing journal, please add it to Orpheus. Similarly, if the journal already exists in Orpheus, but the relevant policy is missing, please add it to the database once you have found it elsewhere.

If the relevant policy in Orpheus is linked to the publisher record instead of directly to the journal of interest, you will see the notice "[inherited from publisher: <publisher name>". In this case, instead of adding or editing policies at the journal level, it is more efficient to navigate to the publisher record and add/edit the policy there instead, because the corrected policy will then be inherited by all journals linked to that publisher record.

Problematic self-archiving (green) policies inherited from RoMEO are marked with a red cross (✖). These policies cannot be trusted until an Orpheus editor has revised and translated its contents. Figures 6 on the following page and 7 on page 17 are an example of an Orpheus journal record with policies inherited from RoMEO before and after policy revision. These policies apply to 5 other journals (Journal of Cosmology and Astroparticle Physics; Journal of Instrumentation; Journal of Physics A: Mathematical and Theoretical; Journal of Physics G: Nuclear and Particle Physics; Journal of Statistical Mechanics: Theory and Experiment).

6 Cambridge-specific features and integrations

This section documents Orpheus features that are tailored for Apollo, the Cambridge institutional repository, and supporting infrastructure. It is probably only of interest to members of the University of Cambridge Office of Scholarly Communication.

6.1 Integration with Zendesk (Orpheus lookup application)

Orpheus lookup is a Zendesk application developed in collaboration with Agustina Martinez. The source code and documentation is available at `https://github.com/osc-cam/orpheus-lookup`.

Figure 6: Example of a journal with self-archiving policies inherited from RoMEO.

Orpheus   Journals Publishers Conferences Sources More ▾   asartori ▾

Classical and Quantum Gravity (Journal) [edit journal] [add synonym] [delete]

Name status	Preferred name
ISSN	0264-9381
e-ISSN	1361-6382
Publisher	IOP Publishing [1412]
Source	SHERPA/RoMEO [view]
Orpheus id	116704

Open Access status [add open access status]

Gold policy [add gold policy]

Green policy [add green policy]

● VoR (arXiv, Commercial repository, institutio...) [inherited from publisher: IOP Publishing [1412]]

Policy applies to	arXiv; Commercial repository; Non-commercial institutional repository; Non-commercial subject repository; Personal website; PubMed Central; Social platforms (Research Gate, etc)
Manuscript version(s)	VoR
Deposit allowed	⊗
Source	IOP Publishing: Author Rights Policy [view]

● AM (institutional repository, subject reposi...) [inherited from publisher: IOP Publishing [1412]]

Policy applies to	Non-commercial institutional repository; Non-commercial subject repository; PubMed Central
Manuscript version(s)	AM
Deposit allowed	●
Embargo period	12 months
Green licence	CC BY-NC-ND
Source	IOP Publishing: Author Rights Policy [view]

● Preprint (arXiv, institutional repository, subject...) [inherited from publisher: IOP Publishing [1412]]

Policy applies to	arXiv; Non-commercial institutional repository; Non-commercial subject repository; Personal website
Manuscript version(s)	Preprint
Deposit allowed	●
Embargo period	0 months
Green licence	Custom
Source	IOP Publishing: Preprint pre-publication policy [view]

● AM (arXiv, Personal website) [inherited from publisher: IOP Publishing [1412]]

Policy applies to	arXiv; Personal website
Manuscript version(s)	AM
Deposit allowed	●
Embargo period	0 months
Green licence	Custom
Source	IOP Publishing: Author Rights Policy [view]

Figure 7: Journal shown in figure 6 on the previous page, showing the effect of translation of RoMEO publisher policy.

Figure 8: Example of Zendesk ticket in which field "#Journal title" was modified by an "Update on metadata received from Symplectic". **A**, current value of field; **B**, value of field at ticket creation.

The application allows *Zendesk* agents to query Orpheus for data on the journal associated to an individual Zendesk ticket. The following ticket fields are used for querying, in this order of precedence (i.e. if one of these fields is not empty, then it is used for querying and the next fields in the list are ignored):

1. #Apollo ISSN

2. #Apollo eISSN

3. #Journal title

> **!** Note that the value of these fields may change after ticket creation and *Orpheus lookup* always uses the current value of the field. Hence, the queried journal title may be distinct from what was recorded in the first Zendesk public comment (the one generated at ticket creation) as exemplified in Fig. 8.

7 Getting help

This manual should be the first point of call for help using and editing Orpheus. If you cannot find the answer to your question on these pages, please make a request for improved documentation (see 8 on the following page).

You can also seek help from other users/editors in the Orpheus Slack channel. If all else fails, you can try talking to the author or e-mailing him at afs25@cam.ac.uk.

8 Reporting bugs, requesting new features

Please use the Orpheus' project board in Asana to report any bugs, request new features, etc.

Index