

---

# **Aristotle Documentation**

***Release 0.1.2alpha***

**Jeremy Nelson**

October 28, 2011



# CONTENTS



Aristotle is a bibliographic [Django](#) project for creating discovery and management of born digital and physical artifacts. Aristotle uses a number of other open-source toolset including [EULFedora](#), [EULXML](#), [Sunburnt](#), and [PyMARC](#). For the Discovery interface, Aristotle uses a forked version of the [Kochief](#) Django application.

Project's documentation resides at [aristotle.readthedocs.org](http://aristotle.readthedocs.org).

### About *Bots* and Versions

You may be wondering as you explore Aristotle's code-base and documentation by the use of **bots** classes.

A few years ago I read Daniel Suarez's great first novel [Daemon](#) and then later watched his FORA.tv's video lecture [Daemon: Bot-Mediated Reality](#)

His vision of **bots** inspired a new design pattern I've been using in my own code. The python-based **bots** in Aristotle's framework encapsulate data and behavior for one or more related tasks within a single class. I try to keep each *bot* class as simple and framework agnostic as possible so I use these **bots** logic in multiple places. This design pattern lends itself very well to programming [Micro-Services](#).

Aristotle uses [Semantic Versioning](#) with the following format X.Y.Z or Major.Minor.Patch. We are currently targeting a full public stable API and production stable 0.5.0 by the forth quarter of 2011.

Contents:



# README

Aristotle is an [Django](#)-based bibliographic, repository, and access project. Unless explicitly stated, all software code and documentation, including any and intellectual property, is licensed under the [Apache 2](#) open-source licence and/or [Creative Commons](#) copyright license.

## 1.1 Dependencies

**aristotle** currently depends on

- [django](#)
- [eulfedora](#)
- [eulxml](#)
- [pymarc](#)
- [sunburnt](#)

## 1.2 Current and Future Django Applications

- *aristotle.catalog – Django application for Discovery and Access to Solr Indices* Default summary and detail views of Solr, FRBR, MARC, and Fedora commons objects with facets.
- *aristotle.discovery – Aristotle Discovery Django application* Forked Kochief discovery application, modified for use as the Discovery Interface for Aristotle.
- *aristotle.datasets Datasets Django Application* Generic scientific, social science, and humanity dataset management for use by other applications.
- *aristotle.etc – Electronic Thesis and Dissertation Django application* Electronic Thesis and Dissertation Django application
- *FUTURE frbr* FRBR object-oriented models front-end to Cassandra, Postgres, or sqlite, or Google App Engine (requires additional modules)
- *aristotle.grx – Gold Rush Micro-services Django application* Gold Rush XML microservices application
- *aristotle.marc MARC Records Django Application* MARC record manipulation including MARC record imports into traditional ILS
- *FUTURE schema\_org* Schema.org microformat support
- *aristotle.vendors – Vendors Django Application* Contains vendor specific services

- vendors.iii Innovative Patron API and item XML extraction applications
- vendors.whitewhale White Whale design templates for Colorado College

# INSTALLATION

To install Aristotle, you will first need to install all of the dependencies mentioned in the <aristotle/readme>.

Detailed directions for installing <aristotle/install\_ubuntu>.

Contents:

## 2.1 Installing Aristotle on Ubuntu

### 2.1.1 Overview

Directions for installing *Aristotle* on a [Ubuntu](#) 11.04 server running on virtual machine.

### 2.1.2 Installing supporting software

Before installing the remainder of the Python supporting modules, make sure you have the Gnu C++ compiler installed. These packages were installed in the following order:

1. As super-user, install gcc with apt-get **\$ sudo apt-get install gcc**
2. As super-user, install the dpkg-dev with apt-get **\$ sudo apt-get install dpkg-dev**
3. As super-user, install curl **\$ sudo apt-get install curl**
4. As super-user, install libxml2-dev **\$ sudo apt-get install libxml2-dev**
5. As super-user, install memcached **\$ sudo apt-get install memcached**
6. As super-user, install libxslt1-dev **\$ sudo apt-get install libxslt1-dev**
7. As super-user, install python-dev **\$ sudo apt-get install python2.7-dev**
8. As super-user, install python-flup **\$ sudo apt-get install python-flup**

### 2.1.3 Installing Django

The Tutt Library uses the Python-based web framework called [Django](#). Please follow the steps listed below.

1. Download stable version of Django (currently 1.3)
2. From the command-line decompress tar file: **\$ tar xzvf Django-1.3.tar.gz**
3. Change directories to the Django-1.3

4. Install Django w/ super-user: **\$ sudo python setup.py install**

## 2.1.4 Installing Aristotle Python Supporting Modules

The Aristotle Django project for all the Tutt Library Python development requires the following Python modules for its functionality. Please download and install in this order of modules (some modules require some dependencies to be installed first)

### **distribute - <http://pypi.python.org/pypi/distribute>**

1. Download the distribute installation script: **\$ wget [http://python-distribute.org/distribute\\_setup.py](http://python-distribute.org/distribute_setup.py)**
2. Install distribute w/super-user: **\$ sudo python distribute\_setup.py install**

### **pip - <http://www.pip-installer.org>**

1. Download the pip installer script: **\$ curl -O <https://raw.githubusercontent.com/pypa/pip/master/contrib/get-pip.py>**
2. Install pip w/super-user **\$ sudo python get-pip.py**

### **httplib2 - <http://code.google.com/p/httplib2/>**

1. Download the latest version of httplib2: **\$ wget <http://httplib2.googlecode.com/files/httplib2-0.7.1.tar.gz>**
2. Extract the tar file: **\$ tar -xf [httplib2-0.7.1.tar.gz](#)**
3. Go to the extracted directory: **\$ cd [httplib2-0.7.1](#)**
4. Install httplib2 with super-user: **\$ sudo python setup.py install**

### **lxml - <http://lxml.de/>**

1. Install lxml using pip w/super-user: **\$ sudo pip install lxml**

### **sunburnt - <https://github.com/tow/sunburnt/>**

1. Install sunburnt using pip w/super-user **\$ sudo pip install sunburnt**

### **eulxml - <https://github.com/emory-libraries/eulxml>**

1. Install eulxml using pip w/super-user **\$ sudo pip install eulxml**

### **eulfedora - <https://github.com/emory-libraries/eulfedora>**

1. Install eulxml using pip w/super-user **\$ sudo pip install eulfedora**

# ARISTOTLE DEVELOPMENT ROADMAP

## 3.1 Current Release 0.1.2alpha

The current version of Aristotle, 0.1.2alpha note.



# ARISTOTLE *0.1.3BETA* RELEASE

## 4.1 Overview

The *0.1.3beta* release for the Aristotle framework includes more functionality and bug fixes and includes an implementation of the multi-threaded MARC-to-Solr parser, complete adoption of the [Pinax](#) Django project by merging the *pinax* branch to the main Aristotle branch.

## 4.2 Issues

The following issues addressed in this release of Aristotle:

- Electronic Facet too broad, should be broken down into types
- Upgrade facets to use jQuery Pagination for long list of facet values
- Advanced Search implemented
- Refworks integration for a list and detail views
- III Patron login
- Export a list of results
- Leave comment
- Search for Item in WorldCat
- Dates for periodicals not being indexed correctly
- Format facet should have icons
- Result list “title” sort not working
- improve temporary functionality of You Can... box
- Display Google Books Cover and Link if available
- **IE display not working...<<https://github.com/jermnelson/Discover-Aristotle/issues/34>>‘\_**
- Migrate framework to Buildout
- give error message when search term not found

## 4.3 Next Release

The next release of Aristotle includes the following.

# ARISTOTLE 0.1.4 RELEASE

## 5.1 Overview

The 0.1.4 release for the Aristotle framework is the first public version to be ready when the new Tutt Library top-level site is released. The 0.1.4 release freezes new functionality in the Discovery application and includes the first public release of the *GoldRush Microservices* and *ETD* applications.

## 5.2 Issues

The following issues addressed in this release of Aristotle:

- Electronic Facet too broad, should be broken down into types
- Upgrade facets to use jQuery Pagination for long list of facet values
- Leave comment
- Format facet should have icons

## 5.3 Future Releases

The following functionality is on the road-map for Aristotle.

---

**Note:** The following future plans for Aristotle are subject to changing priorities for Colorado College Tutt Library.

---

## 5.4 Past Releases

The following release-specific documentation is available.



# ARISTOTLE *0.1.1ALPHA* RELEASE

## 6.1 Overview

This version is the first public release after forking and extending [Kochief](#) Django project. Many of the issues in the *0.1.1alpha* release were addressed with [Kochief](#).

## 6.2 Issues

The following issues are addressed in this release of Aristotle:

- [APA, MLA and other Citations in Detail view](#)
- [Add Hold list item](#)
- [“Start over with a new search” on detail page](#)
- [“Return to search results” on item detail](#)
- [Display LC full subject heading](#)
- [improve temporary functionality of You Can... box](#)
- [parser format labels for marc leader](#)
- [Location and call number in results display](#)
- [Full title should be linkable in result list](#)
- [Repeat pagination widget to bottom of result list](#)



# ARISTOTLE.DISCOVERY – ARISTOTLE DISCOVERY DJANGO APPLICATION

## 7.1 Overview

The `aristotle.discovery` Django application started from a forked version of the [Kochief](#) Discovery interface and catalogue manager Django project.

## 7.2 Dependencies

The **discovery** application doesn't have any external dependencies

## 7.3 Indexing MARC files

To index MARC records `aristotle.discovery` into the `marc_catalog` Solr core, you will need to run the index management command from `aristotle` base directory.

## 7.4 *aristotle.discovery.views* Module Contents

## 7.5 *aristotle.discovery.parsers.marc* MARC record Solr Import Parser

## 7.6 *aristotle.discovery.parsers.tutt\_maps* Location Maps for Tutt Library

## 7.7 *aristotle.discovery.templatetags.discovery\_extras* Template Tags

## 7.8 *aristotle.discovery.management.commands.index* Management Commands



# ARISTOTLE . CATALOG – DJANGO APPLICATION FOR DISCOVERY AND ACCESS TO SOLR INDICIES

## 8.1 Overview

The Django catalog application creates a basic facet view of Solr index of MARC and other types of bibliographic MODS records from a Fedora Repository.

## 8.2 *aristotle.catalog.views* Module Contents

## 8.3 *aristotle.catalog.templatetags* Module Contents

## 8.4 *aristotle.catalog.models* Module Contents

## 8.5 *aristotle.catalog.tests* Unit Tests



# ARISTOTLE . DATASETS DATASETS DJANGO APPLICATION

## 9.1 Overview

The **Datasets Django Application** provides a set of forms and models for managing *small* data sets such as generated from a student's senior project or for supporting faculty research projects of the size typical for a liberal arts college.



# ARISTOTLE .ETD – ELECTRONIC THESIS AND DISSERTATION DJANGO APPLICATION

## 10.1 Overview

The Electronic Thesis and Dissertation Django application was originally a pylons application developed at Colorado College. It has been migrated to the Django and now is part of the Discover Aristotle bibliographic, repository, and access framework being actively developed at Colorado College Tutt Library and other institutions. Unless explicitly set, all software code, ideas, configurations, and any associated intellectual work is licensed under the Apache 2 open source license or under a Creative Commons copyright.

## 10.2 Dependencies

The **etd** application requires the following modules from Emory University, [EULFedora](#) and [EULXML](#). If you want to support student datasets, you'll need into enable Aristotle's *aristotle.datasets Datasets Django Application* application. If you want to supports MARC record generation from an object's MODS metadata, you will need to enable Aristotle's *aristotle.marc MARC Records Django Application*.

## 10.3 *aristotle.etd.views* Module Contents

## 10.4 *aristotle.etd.models* Supporting Models for ETD

## 10.5 *aristotle.etd.tests* Unit tests for ETD



# ARISTOTLE . GRX – GOLD RUSH MICRO-SERVICES DJANGO APPLICATION

## 11.1 Overview

The Gold Rush Micro-services Django application provides an user interface that queries our open-url provider, [Gold Rush](#), and then a specific Solr core to generate a mash-up for inclusion in our library's website and other web applications.

## 11.2 *aristotle.grx.views* Module Contents

## 11.3 *aristotle.grx.models* Module Contents

## 11.4 *aristotle.grx.bots.grxbots* Module Contents

## 11.5 *aristotle.grx.bots.solrbots* Module Contents



# ARISTOTLE .MARC MARC RECORDS DJANGO APPLICATION

## 12.1 Overview

The **MARC Records Django Application** is a collection of utilities for manipulating MARC records. This application allows for importing and exporting MARC records in an ILS, typically normalizing the various vendors MARC records to a single standard.

This application also allows MARC records to be generated out of a Fedora digital repository for importing into an ILS or for external authority control by such organizations as OCLC.

## 12.2 *aristotle.apps.marc.views* Contents

## 12.3 *aristotle.apps.marc.forms* Contents

## 12.4 *aristotle.apps.marc.models* Contents

## 12.5 *aristotle.apps.marc.bots.marcbots* Contents



# CHILD MARC BOTS FOR SPECIFIC PUBLISHERS

- 13.1 *aristotle.apps.marc.bots.awbots* American West Bots
- 13.2 *aristotle.apps.marc.bots.eccobots* Eighteenth Century Collections Online (ECCO)
- 13.3 *aristotle.apps.marc.bots.galebots* Gale Publishing files
- 13.4 *aristotle.apps.marc.bots.gutenbergbots* Project Gutenberg
- 13.5 *aristotle.marc.bots.ltibots* LTI Bots
- 13.6 *aristotle.marc.bots.opbots* Oxford Press Bots
- 13.7 *aristotle.marc.bots.springerbots* Springer EBooks Bots



# ARISTOTLE . VENDORS – VENDORS DJANGO APPLICATION

## 14.1 Overview

Aristotle isolates specific utilities and web services that interact with various vendors into a sub-applications in a master Vendor Django Application for a couple of reasons. The first is that allows common functionality to be shared among different applications in the Aristotle project. For example, the both the *aristotle.catalog – Django application for Discovery and Access to Solr Indices* and *aristotle.etc – Electronic Thesis and Dissertation Django application* modules use the Innovative Patron API microservice to authenticate students for different uses.

## 14.2 EBSCO Django Application

**FUTURE FEATURE** Provides XML search interface for Ebsco Publishing’s full-text and citation databases

## 14.3 III Django Application

A set of utilities for interacting with Innovative’s Millennium ILS includes Item status and Patron API interactions.

### 14.3.1 *vendors.iii.models*

### 14.3.2 *vendors.iii.views*

### 14.3.3 *vendors.iii.bots.iiibots*

## 14.4 OCLC Django Application

**FUTURE FEATURE** Provides access to WorldCat and other OCLC web services as well as FTP upload of MARC records for authority control.

## 14.5 White Whale Django Application

Contains the design templates from White Whale for Colorado College's discovery layer as well as a basic internal library website pages.

### 14.5.1 Templates

- *templates/whitewhale/accessability.html* provides accessible navigation for web-page
- *templates/whitewhale/cc-collapsed-header.html* standard Collapsed header with main navigation for site
- *cc-footer.html* standard Colorado College footer

# INDICES AND TABLES

- *genindex*
- *modindex*
- *search*