Arches Documentation

Release 2.1

Farallon Geographics, Inc.

Apr 27, 2023

Contents

Contents:

CHAPTER 1

Version and Acknowledgments

1.1 Version

This version is issued to support Version 2.1 of Arches, released in 2014.

1.2 Acknowledgments

1.2.1 Contributors:

Ryan Anderson, Farallon Geographics
Yiannis Avramides, World Monuments Fund
Kieran Byrne, English Heritage
Phil Carlisle, English Heritage
Alison Dalgity, Getty Conservation Institute
Rob Gaston, Farallon Geographics
Edmund Lee, English Heritage
Adam Lodge, Farallon Geographics
David Myers, Getty Conservation Institute
Alexei Peters, Farallon Geographics
Koen Van Daele, Flanders Heritage Agency
Dennis Wuthrich, Farallon Geographics

1.2.2 Cover image

Extract from a technical survey of the 18th century arch of the Ironbridge, England, part of the Ironbridge World Heritage Site. Courtesy of Paul Bryan, English Heritage.

1.2.3 Cover design

Ken Feisel, World Monuments Fund

CHAPTER 2

Introduction

Welcome to the **Arches Installation Guide**. This document is intended to help you quickly get Arches running on your server.

Once you have installed Arches there is a companion book to this one, the [**Arches User Guide**](http://www.archesproject.org/documentation), with more information about how to use the software, and how to join the Arches community, a growing online network of Arches developers and users.

2.1 What do I need to know to use this guide?

Arches is designed to let you manage many types of cultural heritage data. And while we've tried to make it easy to install Arches, you'll need to make some decisions on how you wish to describe and manage the information that goes into Arches. But don't worry, we've also included lots of examples and well-thought-out defaults for setting up the software.

So, installing Arches means loading software and deciding how you want to describe the cultural heritage information that's important to you. We've written this guide assuming that you are comfortable working with your computer's command line, and that you can edit text files with ease.

That's it! Feel free to tap a colleague on the shoulder if you need a little help along the way, or check out the resources at www.archesproject.org for more support.

2.2 Installation Options

It's important to recognize that Arches is designed to manage large, enterprise-scale Cultural Heritage inventories. The software is sophisticated and is designed to be support extensive customization. This means that there are many ways to deploy Arches.

We've written this guide to streamline the installation of Arches Server and the CDS Package in the more straightforward manner possible. These installation instructions provide step-by-step guidelines for installing Arches Server, installing the CDS Package, and configugin Arches for daily use.

CHAPTER 3

System Requirements

There are a few things to know about Arches before we get started.

Arches has the following software dependencies:

- PostgreSQL relational database (version 9.3)
- PostGIS (version 2.x) spatial module for PostgreSQL
- Python (version 2.7.6 - there seem to be issues with later versions of python)
- GEOS

These instructions will walk you through the necessary steps to load all the required dependencies and get Arches up and running quickly.

Please note that Arches has been developed for modern browsers. It supports:

- Firefox
- Chrome
- Safari
- Opera
- Internet Explorer 10 or higher.

3.1 Requirements

- At least 4GB of RAM for evaulation and testing, or 8 to 16GB for production.
- 10GB minimum to install the codebase and test dataset, but diskspace requirements will vary greatly depending on the size of your dataset

CHAPTER 4

Installing Arches on Windows

This chapter presents the steps necessary to acquire and install the Arches core software on your server.

Of course, you may use the same instructions to install Arches on either an in-house or remote server. And, while this guide does not cover it, Arches can be installed across multiple machines for deployments that demand high fault tolerance.

4.1 Step 1: Download Arches

Create the following folder on your computer:

```
arches-web
```

Download the Arches v2.0 code http://archesproject.org/download-arches/ and copy it to the archesproject folder (be sure and unzip if necessary).

4.2 Step 2: Install PostgreSQL 9.3, PostGIS 2.x

Download and install PostgreSQL 9.3 for Windows from:

http://www.postgresql.org/download/.

Notes:

1. Select all the default options when installing and take special note of the password you supply for the superuser (you'll need this later, by default we assume the password 'postgis')
2. At the end of the install leave the checkbox checked to run Stack Builder (which you'll use to install PostGIS) then click the "Finish" button
3. In the Stack Builder drop-down menu, select the local Postgres database (something like PostgreSQL 9.3 on port 5432) then click "Next"
4. Expand the Spatial Extensions link and check the box for PostGIS. Click "Next"

5. Select your download directory and click "Next" again
6. Click "Next" to start the installation of PostGIS
7. Select all the defaults until you get to the screen where it asks for your password. Type in the password that you created earlier.
8. Select "Yes" to register the GDAL_DATA environment variable ,unless you have it currently set to something else and you don't want to override it.
9. Click "Close" and then "Finish" to complete the setup of Postgres with PostGIS
10. Assuming that your Postgres installation is in C:Files.3, add this to your system PATH (If you are unsure how to do this, look here):

    ```
    C:\Program Files\PostgreSQL\9.3\bin
    ```

After installing PostgreSQL you'll need to run 2 commands from a command window (this only applies to PostgreSQL 9.2 and above). Open a command window and at the command prompt type the following:

```
createdb -U postgres -E UTF8 -T template0 template_postgis_20
psql -U postgres -d template_postgis_20 -c "CREATE EXTENSION postgis;"
```

Note: To confirm that you have installed PostgreSQL, open a command window and type: psql. You should be prompted for a password. If so, then PostgreSQL is installed and you may close the command prompt. If you received an error like "psql is not recognized as an internal or external command, operable program or batch file" then you probably need to add an entry to your path.

Warning: PostgreSQL Warnings

If you experience errors during the installation, please check that the setting 'standard_conforming_strings' in 'postgresql.conf' is uncommented and set to 'off'. See this question on stackoverflow for more information.

4.3 Step 3: Install GEOS

Download and install GEOS from http://trac.osgeo.org/osgeo4w/

Notes:

- Make sure to select the x86 version for 32bit machines, or x86-64 for 64bit machines
- Assuming that your OSGeo4W installation is in `C:\OSGeo4W\` directory, add this to your system PATH:

  ```
  C:\OSGeo4W\bin
  ```

4.4 Step 4: Install Python 2.7.6

Download Python at http://www.python.org/download

Notes:

- Download the latest MSI Installer for version 2.7.6 (Note: make sure to select the x86 version for 32bit machines, or x86-64 for 64bit machines)

- Run the installer and select all the default options when prompted.
- Assuming that your Python installation is in `C:\Python27\`, add this to your PATH:

```
C:\Python27\;C:\Python27\Scripts
```

Warning: Python Versions

While you should be able to install versions of Python later then **2.7.6**, we've had users say that they've had trouble with 2.7.7 and 2.7.8 If you want a guaranteed good result stick with version **2.7.6**

4.5 Step 5: Install the latest Java Development Kit (JDK)

Download and install the JDK from http://www.oracle.com/technetwork/java/javase/downloads/index.html

Notes:

1. Select all the default options, except at the very end, uncheck the box that asks to install the Ask.com toolbar.
2. Right-click the "My Computer" icon on your desktop and select "Properties" to set the JAVA_HOME environment variable.
3. Click the "Advanced" tab.
4. Click the "Environment Variables" button.
5. Under "System Variables", click "New".
6. Enter the variable name as JAVA_HOME.
7. Enter the variable value as the installation path for the Java Development Kit. (For example, if you used the default path for the Java Development Kit during installation, use 'C:FilesJava1.7.0_25'. This string should be your variable value.)
8. Click "OK".
9. Click "Apply Changes".
10. Close any command window which was open before you made these changes, and open a new command window. There is no way to reload environment variables from an active command prompt. If the changes do not take effect even after reopening the command window, restart Windows.

4.6 Step 6: Install Web Framework and Search Engine

Arches uses the Django web framework (https://www.djangoproject.com/) and the ElasticSearch engine (http://www.elasticsearch.org/). This step installs these technologies. Open a command window and navigate to:

```
archesproject/install
```

Then run the following bat file. This installs the that Arches uses:

```
install_dependencies.bat
```

Once this bat file runs you'll need to start the Django development server and ElasticSearch search engine, by navigating to the root folder and running the following bat file:

```
runserver.bat
```

Warning: If you get an error similar to this, "Could not import user-defined GEOMETRY_BACKEND"geos"", you might try adding the path to the GDAL dll to your archesproject/settings.py file (for more information see the djagno docs):

GDAL_LIBRARY_PATH = "C:/OSGeo4W/bin/gdalxxx.dll" #<– make sure this path is correct and points to the actual dll, note the forward slashess

Warning: A note about runserver.bat

Windows may automatically open 2 command windows as part of this step (if you double clicked runserver.bat for example). One window is for the inbuilt Django webserver, the other one is for the Elasticsearch search engine.

Warning: If you happen to run runserver.bat again without closing those 2 windows first then you'll end up with mulitple instances of the webserver and search engine running. The search engine can consume a lot of resources and additionaly will try and replicate itself to the new instance, taking up additional memory and hard drive space. If you want to disable this "feature" then

uncomment line 87 in the archesproject/arches/Search/engines/elasticsearch-0.90.3/config/elasticsearch.yml file where it says: #node.max_local_storage_nodes: 1

Warning: Feel free to minimize these windows, but if close them you will stop your either your web server or search engine depending on which one you closed. Before re-running runserver.bat make sure to close both windows.

Note: We use the Django development server to make it easy for you to confirm that Arches has installed properly. You WILL NEED to configure Arches to use a production quality web server such as Apache if you want to use Arches in production. See the chapter on Arches configuration for guidance on how to complete this step.

4.7 Step 7: Install and Build Arches

Open a new command window, and navigate to:

```
archesproject/build
```

Run the following script to build Arches:

```
install_arches_db.bat
```

Note: Database Password

You may be prompted for a password to the Arches database. The default password is: postgis. You can open Settings.py in the arches root folder to change the default arches password, or to install Arches on to an existing instance of PostgreSQL/PostGIS.

4.8 Next Steps

You've just completed the steps needed to install the core Arches application. To confirm that Arches installed properly, open a new command window, and navigate to:

```
archesproject/build
```

Run the following script to build Arches:

```
build.bat
```

Confirm that Arches is running by typing the following URL

```
http://localhost:8000/Arches/index.htm
```

Arches requires that you load a data management package before you can start working with your cultural heritage data. Jump to the chapter entitled **Loading the CDS Package** to install a cultural heritage data management package based on the CIDOC Core Data Standard.

You may also check the archesproject.org website to see whether the community has developed other data management packages.

It's also possible to create your own Arches Data Management Package. You'll want to review an existing package (such as the CDS Package) to see how to structure an Arches package. You can find more information on creating packages by joining the Arches Discussion Forum (http://archesproject.org/forum/).

CHAPTER 5

Installing Arches on a Linux (Ubuntu 12.04)

The following instructions were written with Ubuntu 12.04 LTS in mind. Other versions of Ubuntu or other Linux distributions might have some different steps, but the general outline should be similar. All instructions assume that they are entered through the terminal and that the user has root privileges. Installing on alternate flavors of Linux will follow a similar pattern.

Of course, you may use the same instructions to install Arches either an in-house or remote server. And, while this guide does not cover it, Arches can be installed across multiple machines for deployments that demand high fault tolerance.

5.1 Step 1: Download Arches

Create the following folder on your computer:

```
arches-web
```

Download the Arches v2.0 code http://archesproject.org/download-arches/ and copy it to the archesproject folder (be sure and unzip if necessary).

5.2 Step 2: Install Required Dependencies

This step installs PostgreSQL, PostGIS, GEOS and the latest JDK by running a single script.

Note: If you would rather install the components seperately, simply open the script file and run only the portions of the script you want.

Open a terminal and navigate to:

```
archesproject/install
```

Run the following script to build install PostgreSql, PostGIS, GEOS and the latest JDK:

```
source ubuntu_precise_setup.sh
```

Warning: PostgreSQL Warnings

If you experience errors during the installation, please check that the setting 'standard_conforming_strings' in 'postgresql.conf' is uncommented and set to 'off'.

See this question on stackoverflow for more information.

5.3 Step 3: Install Web Framework and Search Engine

Arches uses the Django web framework (https://www.djangoproject.com/) and the ElasticSearch engine (http://www.elasticsearch.org/). This step installs these technologies. Open a command window and navigate to:

```
archesproject/install
```

Then run the following script:

```
source install_dependencies.sh
```

Once this script runs you'll need to start the Django development server and ElasticSearch search engine, by navigating to the root folder and running the following script:

```
source runserver.sh
```

Warning: If you get an error similar to this, "Could not import user-defined GEOMETRY_BACKEND "geos"", you might try adding the path to the GDAL dll to your archesproject/settings.py file (for more information see the django docs):

```
GDAL_LIBRARY_PATH = '/path/to/libgdal.so'   #<-- make sure this path is correct and␣
↪points to the actual .so file, note the forward slashess
```

Warning: A note about runserver.bat

Running runserver.sh runs two process, one for the inbuilt Django webserver, the other for the Elasticsearch search engine.

If you happen to run runserver.sh again without first killing both of those processes then you'll end up with mulitple instances of the webserver and search engine running. The search engine can consume a lot of resources and additionaly will try and replicate itself to the new instance, taking up additional memory and hard drive space. If you want to disable this "feature" then

uncomment line 87 in the archesproject/arches/Search/engines/elasticsearch-0.90.3/config/elasticsearch.yml file where it says: #node.max_local_storage_nodes: 1

Note: We use the Django development server to make it easy for you to confirm that Arches has installed properly. You WILL NEED to configure Arches to use a production quality web server such as Apache if you want to use Arches in production. See the chapter on Arches configuration for guidance on how to complete this step.

5.4 Step 4: Install and Build Arches

Open a new terminal window, and navigate to:

```
archesproject/build
```

Run the following script to build core Arches:

```
source install_arches_db.sh
```

Note: Database Password

You may be prompted for a password to the Arches database. The default password is: postgis. You can open Settings.py in the arches root folder to change the default arches password, or to install Arches on to an existing instance of PostgreSQL/PostGIS.

5.5 Next Steps

You've just completed the steps needed to install the core Arches application. To confirm that Arches installed properly, open a new command window, and navigate to:

```
archesproject/build
```

Run the following script to build Arches:

```
source build.sh
```

Confirm that Arches is running by typing the following URL

```
http://localhost:8000/Arches/index.htm
```

Arches requires that you load a data management package before you can start working with your cultural heritage data. Jump to the chapter entitled **Loading the CDS Package** to install a cultural heritage data management package based on the CIDOC Core Data Standard.

You may also check the archesproject.org website to see whether the community has developed other data management packages.

It's also possible to create your own Arches Data Management Package. You'll want to review an existing package (such as the CDS Package) to see how to structure an Arches package. You can find more information on creating packages by joining the Arches Discussion Forum (http://archesproject.org/forum/).

CHAPTER 6

Installing Arches on a Mac (notes only)

There are no official instructions for installing Arches on a Mac other then to say that it closesly follows the instructions for installing Arches on Ubuntu linux. Several of us have installed Arches successfully on Macs, so we know it can be done.

Here's a good resource for information about installing some of the software that Arches depends on: https://docs.djangoproject.com/en/dev/ref/contrib/gis/install/#macosx

Note: If you run into issues with GDAL you might want to look at this thread http://stackoverflow.com/questions/11294556/missing-libgeos-c-so-on-osx. You may need to add entries into your settings.py file for the locations of your GDAL and GEOS libraries (GEOS_LIBRARY_PATH and GDAL_LIBRARY_PATH settings).

CHAPTER 7

Installing Arches on a Vagrant Development Machine

This chapter summarizes the installation steps a Software Developer should follow for installing Arches on a Vagrant development machine. Developers may wish to install Arches using Vagrant. Vagrant ensures that all developers can work on the same platform. And if you decide to use a local Mercurial client (like SourceTree), Vagrant will allow you to quickly test your code in a virtual environment.

Check out http://www.vagrantup.com/ for more information.

7.1 Step 1: Clone Arches

Start by getting a copy of the Arches code using a local Mercurial client:

```
hg clone https://bitbucket.org/arches/arches ~/projects/arches
```

Then, (assuming you want to develop using the CDS package) clone the CDS repo into your packages folder:

```
hg clone https://bitbucket.org/arches/cds ~/projects/arches/archesproject/packages/cds
```

And create a settings local file as described in "Loading the CDS Package", if you're using linux, you can run the following commands: touch ~/projects/arches/settings_local.py echo "INSTALLED_PACKAGES = ('cds',)" >> ~/projects/arches/settings_local.py

7.2 Step 2: Install Vagrant

Navigate to http://www.vagrantup.com; download and install the Vagrant software for your operating system (OS).

7.3 Step 3: Install VirtualBox

VirtualBox is an open source Virtual Machine application.

Navigate to https://www.virtualbox.org; download and install VirtualBox for your OS.

7.4 Step 4: Download Virtual Machine File

Open a command prompt and type the following command to download an empty Ubuntu server (version 12.04, 32-bit) VM file

```
vagrant box add precise32 \ http://files.vagrantup.com/precise32.box
```

You can consult the Vagrant website if you prefer to use an alternate base OS for your Arches VM.

Note: Pro Tip: Windows Installation

For some versions of Windows you may get an error when running the vagrant box add command. Try running the command again, this time WITHOUT the "/" after the "precise32". Check the latest vagrant documentation for additional troubleshooting help.

7.5 Step 5: Build Arches

Open a command prompt and navigate to folder where you've cloned arches and run:

```
vagrant up
```

Note: Pro Tip

Making Arches run with less than 2 GB of RAM I>Before you "vagrant up", open this file: arches/Vagrantfile with a text editor. Scroll down to vb.customize ["modifyvm", :id, "–memory", "2048"]. Replace the 2048 with 1024, and Arches will only request 1GB of RAM.

Vagrant will create a Virtual Machine with a complete Arches install, including all required software dependencies, as well as a sample dataset ready for you to test Arches with.

Note: How long will it take to build Arches?

Depending on your computer, it may take anywhere from 10 minutes to an hour or two to build Arches.You'll know that Arches is ready when the command prompt returns.

7.6 Step 6: Start Arches

At the command prompt type:

```
vagrant ssh
```

This command will log you onto your virtual Linux server. To start Arches, run the following commands:

```
cd /vagrant ./runserver-vagrant.sh
```

Note: Windows Users: Get an ssh client

Because Windows doesn't usually come with ssh, you'll need to install a ssh client like PuTTY. See the Vagrant documentation (http://docs-v1.vagrantup.com/v1/docs/getting-started/ssh.html) for more information.

Arches will start and be ready for use in just a moment or two. To access Arches, open a browser (note: Arches works with Firefox, Chrome, Safari, Opera, and Internet Explorer 10 or higher), and type:

http://localhost:8000/Arches/index.htm#

You should see the following screen:

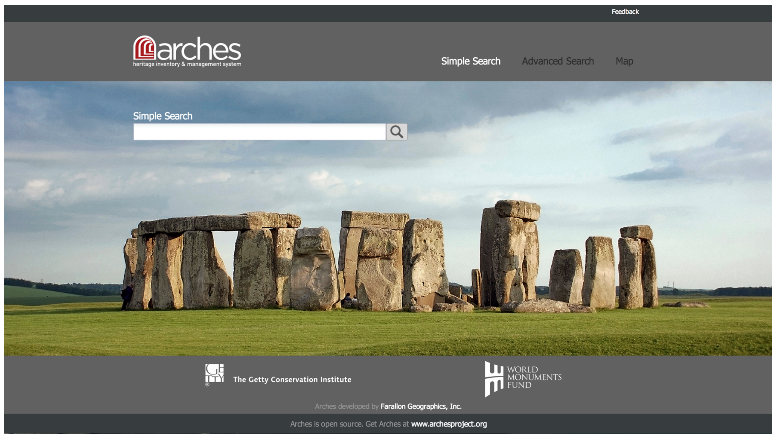

Feedback
arches
heritage inventory & management system
Simple Search
Advanced Search
Map
Simple Search
The Getty Conservation Institute
WORLD MONUMENTS FUND
Arches developed by Farallon Geographics, Inc.
Arches is open source. Get Arches at www.archesproject.org

Fig. 1: Arches Default Screen: Simple Search

If you want to test adding or editing cultural heritage data in Arches, you'll need to sign in. Click "Map" in the upper right corner of the screen, then click

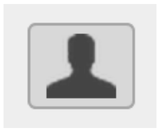


Fig. 2: Arches Sign In

The Username and Password are both "admin" (omit the quotes when entering the username and password).

For more help in understanding the Arches app, check out the **Arches User Guide** to learn how to work with Arches.

Warning: Using the test data-set

All test data supplied is for experimentation in Arches only. The records are not certified for accuracy or completeness.

7.6.1 A Note on Vagrant

The Vagrant install is only intended for developer use and is not intended for use in production. Users should not store production data in a Vagrant generated virtual machine. Developers should familiarize themselves with the vagrant command line tools as outlined in the Vagrant documentation: http://docs.vagrantup.com/v2/cli/index.html

CHAPTER 8

The Core Data Standard (CDS) Package

The Arches community has developed the CDS Package v1.0 as an initial data management module for Arches Server v2.0.0. The CDS Package demonstrates the structure and contents of an Arches Package, and implements the following standards:

- CIDOC Conceptual Reference Model (CRM) (www.cidoc-crm.org). The CRM provides definitions and a formal structure for describing the implicit and explicit concepts and relationships used in cultural heritage documentation.
- International Core Data Standard for Archaeological and Architectural Heritage. This is a soon-to-be finalized standard for the inventory of both archaeological and architectural heritage, which is based on the earlier Core Data Index to Historic Buildings and Monuments of the Architectural Heritage (adopted by the Council of Europe in 1992) and the Core Data Standard for Archaeological Sites and Monuments (adopted by CIDOC in 1995). The new standard under preparation (referred to here as the "CDS") was used to identify the data fields of the CDS. Organizations that deploy the CDS can customize those data fields to meet their specific requirements.

You may wish to familiarize yourself with these data standards as part of your the CDS installation and deployment effort. If you ever wish to further customize the CDS, familiarity with both will help you greatly along the way.

Some of the key contents of the CDS Package include:

- Resource Graphs
- Authority Documents and Thesauri
- Data Import Files

The following sections summarize these components of an Arches Package.

8.1 Resource Graphs

Arches is designed to manage cultural heritage data anywhere in the world. Needless to say, that's an ambitious goal. After all, architecture considered culturally significant in San Francisco - a city founded in 1776 - might not merit much comment in Cairo or London.

So, how does Arches resolve this?

Arches Server requires a set of Resource Graphs that define the set of resource types that you wish to include in your inventory and the terms that you will use to describe them.

Note: What is an Arches Resource Graph?

In Arches, the term “Resource Graph” refers to a class of cultural heritage records. Things like “Architectural Heritage”, “Investigation Activity”, and “Person” are all examples of Resource Graphs. Think of a Resource Graph as defining the attributes for a particular category of information that Arches will manage.

A Resource is simply one instance of a particular Resource Graph.

Resource Graphs are described following the CIDOC Conceptual Reference Model (CRM). The CRM is an ontology for cultural heritage information that has been developed by a the International Committee for Documentation (CIDOC) of the International Council of Museums. Arches uses the CRM because it was adopted by the International Organization for Standardization as the standard (ISO 21127:2006) for the interchange of cultural heritage information.

8.1.1 CDS Package Resource Graphs

The CDS Package defines the following Resource Graphs:

```
ARCHAEOLOGICAL HERITAGE (ARTIFACT).E18
ARCHAEOLOGICAL HERITAGE (SITE).E27
ARCHITECTURAL HERITAGE.E18
LANDSCAPE HERITAGE.E27
MARITIME HERITAGE.E18
INVESTIGATION.E7
MANAGEMENT.E7
DESIGNATION AND PROTECTION.E7
HISTORICAL EVENT.E5
DOCUMENT.E31
IMAGE.E38
PERSON.E21
ORGANIZATION.E74
```

Heritage Resources: Heritage Resources are archaeological, built, landscape, or other immovable cultural heritage. In the CDS, Heritage Resources include:

Archaeological Heritage (element) : a single archaeological entity that could stand alone or be an element of a larger archaeological group (e.g., a bath house within a Roman villa)

Archaeological Heritage (site) : an area of archaeological potential or an area of known or discovered archaeological elements

Note: What’s the difference between archaeological elements and sites?

While conceptually these categories overlap, the CDS differentiates between the two because of the way that they are represented using the CIDOC CRM.

Architectural Heritage : culturally significant buildings, structures, and groups thereof

Landscape Heritage : areas of land designed and created intentionally by man, such as garden and parkland landscapes constructed for aesthetic reasons, organically evolved areas of land resulting from an initial social, economic, administrative, and/or religious imperative and that has developed its present form by association with and in response to its natural environment, or areas of land that are culturally significant due to powerful religious, artistic or cultural associations of the natural element rather than material cultural evidence, which may be insignificant or even absent (UNESCO)

Maritime Heritage : underwater heritage (both under sea and inland), which may include heritage inundated by sea level rise and dam construction, shipwrecks and aircraft, as well as heritage afloat (e.g., ships, sailing vessels)

Activities Activities are events or actions that may take place during a given time span and at a location or area. In the CDS, Activities include:

Investigation Activity : an activity undertaken with the explicit intention of gathering information about, and understanding of, a Heritage Resource, and the creation of an information source to record that information and understanding

Management Activity : an activity undertaken to prevent damage to, promote the survival of, and promote the understanding and appreciation of Heritage Resources.

Designation and Protection Activity : an activity which implements or revokes statutory and non-statutory designation and protection regimes which may apply to Heritage Resources

Historical Event : any activity that took place in the past, including both human and natural events

Documents Documents are information carriers such as books, texts, periodicals, inscriptions, audio files, video files, 3-D models, or images. In the CDS, Documents include:

Document : an information carrier, other than an Image, whether physical or digital, eg. books, maps, pdfs, word-processed documents

Image : an information carrier that represent an external form, whether physical or digital, eg. photographs, slides, drawings, jpegs or tiff files.

Actors Actors refer to individuals or groups of people. In the CDS, Actors include:

Person : real persons (i.e., who live or are assumed to have lived)

Organization : A group or legally identifiable body

8.2 Authority Documents

Like any good data-editing tool, Arches supports consistent and valid data entry. Almost every one is familiar with this concept: it's the ability to select a valid value from a list:

Notice that in this Arches data entry form, you're able to select the kind of Archaeological Heritage Element from among the list of approved types or classifications. In Arches, these lists are called "controlled vocabularies".

But what if you want to use a particular list of period names, instead of the names that come with Arches? Arches allows you to define exactly the set of names you want to use for every resource.

8.2.1 Thesauri

So, why do we use the term "controlled vocabulary" if all we're doing is selecting values from a drop-down list? The answer is that Arches treats each drop-down list as if it's a thesaurus.

And what's so special about thesauri? They provide a potent way organize concepts and a convenient way to attach labels to concepts. This makes data entry and searching a large database for specific resources much easier.

Lets say we want to create a list of dwellings where people live as a drop-down list in Arches. Our list might look like this:

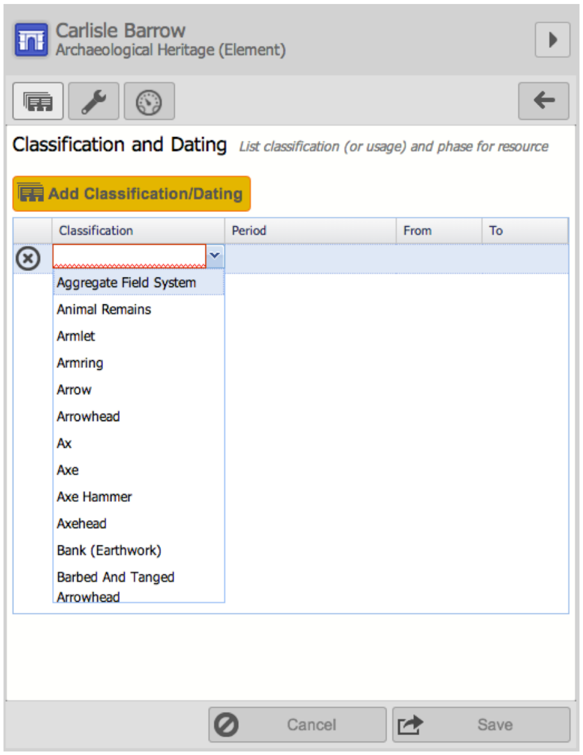

Carlisle Barrow
Archaeological Heritage (Element)
Classification and Dating
List classification (or usage) and phase for resource
Add Classification/Dating
Classification
Period
From
To
Aggregate Field System
Animal Remains
Armlet
Armring
Arrow
Arrowhead
Ax
Axe
Axe Hammer
Axehead
Bank (Earthwork)
Barbed And Tanged Arrowhead
Cancel
Save

Fig. 1: Selecting a value from a "Controlled Vocabulary"

```
dwelling
  apartment
  house
  hut
  log cabin
```

The thesaurus allows you to define concepts and give them labels. For example, "dwelling" is the label associated with the concept of "a place where people live."

Note also that each concept under "dwelling" is a more precise (or narrower) type of dwelling. In this example, there are four additional narrower concepts of dwelling. One such concept (eg.: a place where families live) has the term "house" associated with it.

This ability to build hierarchies of concepts makes thesauri very powerful. We can use the structure of a thesaurus to help make clear the relationships between concepts.

But wait just a moment. We know that in some areas, people use different terms to mean the same thing. A thesaurus allows us to attach many labels to the same concept. For example:

```
dwellings
  apartment
  house [aisled house, bungalow, chalet]
  hut
  log cabin
```

This is a representation that adds more, alternate labels for the concept "house". In this example, the labels "aisled house," "bungalow," and "chalet" all represent alternate labels for the same concept. Note that we could add "Haus" and "maison" if we wanted to include labels for the same concept in additional languages such as German or French.

It's up to you to decide how you want to organize your terms. Is a "chalet" really an alternate label for house, or should it be considered a narrower concept of dwelling?

At this point you may be wondering why we've made a simple drop-down list so complicated.

Well, by using thesauri to power the drop-down lists, Arches will know that when you search for a resource using the term "chalet" you really mean "house". And if you search for a resource using the term "dwelling", you'll get all the narrower concepts for dwelling, even if you didn't know that a "hut" a "dwelling".

8.2.2 Understanding the Structure of an Authority Document

Let's look at an authority document in more detail:

```
ARCHITECTURAL HERITAGE TYPE AUTHORITY DOCUMENT
```

This authority document holds the list of Architectural Heritage types that the CDS will use. Types are just what you'd guess: a list of the kinds of Architectural Heritage that you will allow people to choose from when they enter a new Architectural Heritage resource into Arches.

Your ARCHITECTURAL HERITAGE TYPE Authority Document can contain any entries that you think are appropriate. Here are some examples:

```
Air Raid Siren
Church
House
Chalet
Fountain
Street Lamp
```

Adding these concepts to Arches would require you to update the Architectural Heritage authority document. The good news is that all Arches authority files are simple text files with the following 5 columns:

ConceptID: a unique identifier for the concept

PrefLabel: the default label that you wish to use for this concept

AltLabels: alternative ways to identify the concept

Parent ConceptID: the unique identifier of a broader concept (this will construct hierarchies)

Provider: identifies the source of the concept

Building an authority document is just a matter of adding a row for every concept that you want to include in your drop down lists. In our example, the authority document might look like this:

ConceptID	PrefLabel	AltLabels	Parent ConceptID	Provider
1	Air Raid Siren			EH
2	Church	House of Worship		EH
3	House			EH
4	Chalet		3	EH
5	Fountain	fount;jet;spout		EH
6	Street Lamp			EH

Notice that in one line, we can add a concept, define its preferred label, optionally provide alternate labels, and define broader/narrower relationships between concepts.

There are a few things to be aware of:

Warning: ConceptIDs must be unique!

Make sure that you use a truly unique identifier for each record that you want to load into Arches.

Note: Use a semicolon to separate more than one Alternate Label

Commas are used to separate columns, so we need a different way to indicate all the different alternate labels for a concept

Note: Narrower Terms

Use the ConceptID of the broader term in the "Parent ConceptID" field if you want to tell Arches that concepts are related. Arches will use this information to make searching the inventory better and more intuitive.

Note: Required Terms

Arches has only one required term: The term "Primary" is required to appear in NAME TYPE AUTHORITY DOCUMENT. Why? Because Arches supports multiple names for any resource, it needs a way to define the preferred name for a resource.

8.2.3 Understanding Authority Document Values

You might have noticed that a few Authority Documents seem to come in pairs. For example:

```
CULTURAL PERIOD AUTHORITY DOCUMENT.csv
CULTURAL PERIOD AUTHORITY DOCUMENT.values.csv
```

What's going on here? Sometimes its convenient to know a bit more about some of the concepts in a thesaurus than just a set of labels and its parent. For example, when discussing specific cultural periods (such as "Middle Paleolithic"), it's often useful to reference the start and end years of the period.

Arches "values" files let us do this. Open CULTURAL PERIOD AUTHORITY DOCUMENT.values.csv and you'll see the following:

Note the first row in this file: **conceptid**, **Value**, **Value Type**, **Provider**

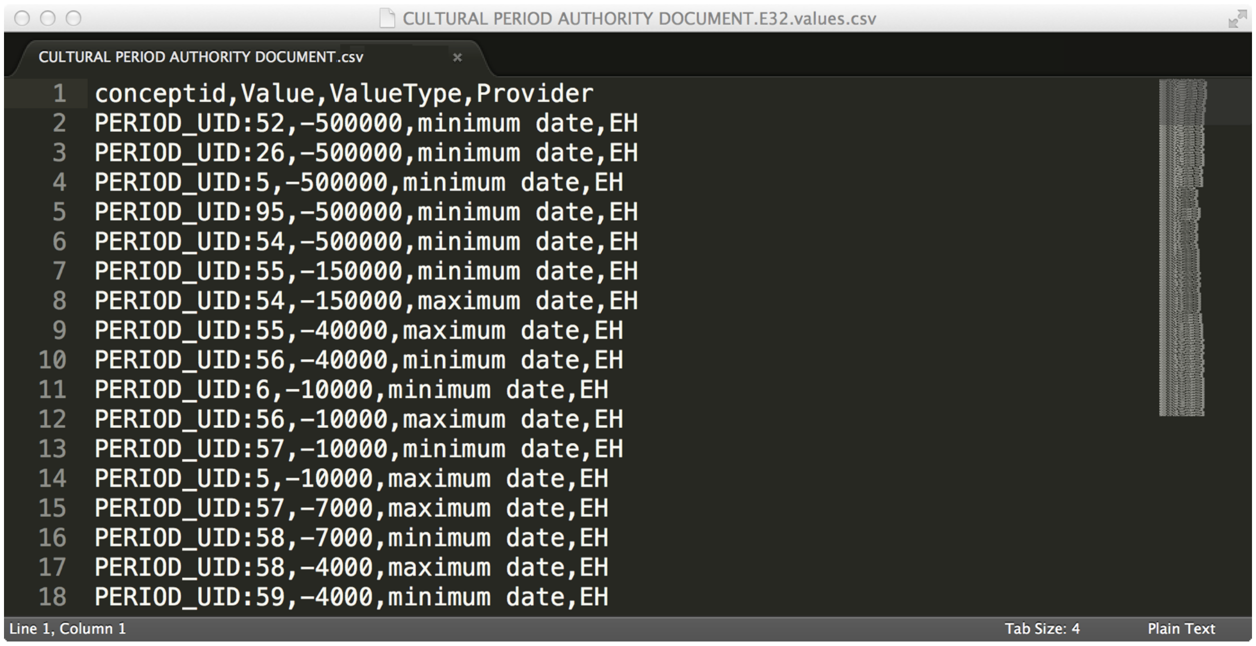
These headers define a simple structure of the value file. You simply have to reference a conceptID from the CULTURAL PERIOD AUTHORITY DOCUMENT.csv file, provide the value and its "type", and the provider of the value.

For cultural periods, the values file allows you to attach the start year and end year for each Period concept in the CULTURAL PERIOD AUTHORITY DOCUMENT.csv file. Look at lines 7 and 9 in the image above and you will see:

```
7  PERIOD_UID:55, -150000, minimum date, EH
9  PERIOD_UID:55, -40000, maximum date, EH
```

If you look up "PERIOD_UID:55" in CULTURAL PERIOD AUTHORITY DOCUMENT.csv, you'll see that it's record is:

```
PERIOD_UID:55,MIDDLE PALAEOLITHIC,,PERIOD_UID:5,EH
```



CULTURAL PERIOD AUTHORITY DOCUMENT.E32.values.csv

CULTURAL PERIOD AUTHORITY DOCUMENT.csv

```
1  conceptid,Value,ValueType,Provider
2  PERIOD_UID:52,-500000,minimum date,EH
3  PERIOD_UID:26,-500000,minimum date,EH
4  PERIOD_UID:5,-500000,minimum date,EH
5  PERIOD_UID:95,-500000,minimum date,EH
6  PERIOD_UID:54,-500000,minimum date,EH
7  PERIOD_UID:55,-150000,minimum date,EH
8  PERIOD_UID:54,-150000,maximum date,EH
9  PERIOD_UID:55,-40000,maximum date,EH
10 PERIOD_UID:56,-40000,minimum date,EH
11 PERIOD_UID:6,-10000,minimum date,EH
12 PERIOD_UID:56,-10000,maximum date,EH
13 PERIOD_UID:57,-10000,minimum date,EH
14 PERIOD_UID:5,-10000,maximum date,EH
15 PERIOD_UID:57,-7000,maximum date,EH
16 PERIOD_UID:58,-7000,minimum date,EH
17 PERIOD_UID:58,-4000,maximum date,EH
18 PERIOD_UID:59,-4000,minimum date,EH
```

Line 1, Column 1 Tab Size: 4 Plain Text

Fig. 2: Cultural Period Values

So, in the CDS pacakge, the default list of Cultural Periods includes the Middle Palaeolithic (as “PERIOD_UID:55” in CULTURAL PERIOD AUTHORITY DOCUMENT.csv), and that period is defined to start at -150,000 years (row 7 in CULTURAL PERIOD AUTHORITY DOCUMENT.values.csv) and end at -40,000 years (row 9 in the same file).

What Value Types Can I Define?

You can define any set of key/value pairs that you wish. Arches will load your values and associate them with the concept identifier that you provide.

By default, Arches defines values for the following Authority Document value files:

```
ADMINISTRATIVE SUBDIVISION AUTHORITY DOCUMENT.values.csv
ARCHES RESOURCE CROSS-REFERENCE RELATIONSHIP TYPE AUTHORITY DOCUMENT.values.csv
CONDITION AUTHORITY DOCUMENT.values.csv
CULTURAL PERIOD AUTHORITY DOCUMENT.values.csv
TYPE OF DESIGNATION OR PROTECTION AUTHORITY DOCUMENT.values.csv
UNIT OF MEASUREMENT AUTHORITY DOCUMENT.values.csv
```

If you look at these files, you’ll see that Arches already knows how to handle the following value types:

SortOrder You can use the *.values.csv file to define the order in which terms appear for:

```
UNIT OF MEASUREMENT AUTHORITY DOCUMENT.csv
TYPE OF DESIGNATION OR PROTECTION AUTHORITY DOCUMENT.csv
CONDITION AUTHORITY DOCUMENT.csv
```

Minimum Date, Maximum Date You can use the *.values.csv file to define the starting and ending dates for:

```
CULTURAL PERIOD AUTHORITY DOCUMENT.values.csv
```

Geometry, Administrative Area Type You can use the ADMINISTRATIVE SUBDIVISION AUTHORITY DOCUMENT.values.csv file to define administrative subdivision types and the geometry associated with each administrative area. For example:

```
LOC_UID:97317,PARISH,admin area type filter,EH
```

would define the concept LOC_UID:97317 (in the ADMINISTRATIVE SUBDIVISION AUTHORITY DOCUMENT.csv file) as an "admin area type filter" with a value of "PARISH". And

```
LOC_UID:97345,"POLYGON((-1.32078640715254 54.3728610449506,-1.33713112471483 54.
↪3586411276027,-1.3584180849332 54.371070777731,-1.34399265906457 54.376710180739,-1.
↪32078640715254 54.3728610449506))",geometry,EH
```

would define the concept LOC_UID:97345 (in the ADMINISTRATIVE SUBDIVISION AUTHORITY DOCUMENT.csv file) as having a geometry with a set of particular coordinates that make up a polygon.

Resource Cross-Reference, Cross-Reference Type Filter Arches allows you to build relationships between resources. For example, you can relate a Person resource to an Architectural Heritage resource. Arches implements an Authority File that allows you to define the types relationships between resources.

```
ARCHES RESOURCE CROSS-REFERENCE RELATIONSHIP TYPE AUTHORITY DOCUMENT
```

Custom Key/Values Pairs Arches will allow you to load any set of key/value pairs, treating the information that you load as descriptive data about the concept.

If you want to use your key/value pairs to support data entry (like the from/to dates for Cultural Periods), you may need to extend the Arches codebase he codebase to use your key/value pairs.

8.2.4 Modifying the Default Authority Files

The CDS package comes with a set of concepts and labels for each controlled vocabulary. Of course, you are welcome to use the package defaults but you'll probably want to use your own concepts and terms.

To modify the default authority files, just open an authority document with a text editor and begin. You can remove records from a file, or add new records that better represent the concepts, labels, and concept-relationships that you use to describe cultural heritage.

Note: Back up the original Authority Documents First!

It's a good idea to make copies of the original files, just in case you want to go back and confirm that your edits make sense.

Note: Don't Forget about the "values" Files!

If you change an Authority Document that has an associated values file, you'll want to make sure that the conceptIDs match between the Authority Document and the corresponding value file. By the way, Arches will work without any "values" files. They're just a convenient way to augment Authority Documents.

Note: Geometry in Authority Files

Sometimes the additional "values" that you want to associate to concepts within an authority file are geometries that can be represented on the map or used for spatial analysis. For example, if you have a controlled list of states or provinces, you may wish to assocaite the relevant geometry to describe its borders.

You can use the *.values file to pass in a geometry associated to a concept using a text-based format called Well Known Text (or WKT for short.) See http://en.wikipedia.org/wiki/Well-known_text for a good description of WKT. Arches assumes that all WKT passed into the system is in Lat/Lon coordinates using WSG84 datum (also known as EPSG 4326 http://spatialreference.org/ref/epsg/4326/).

The CDS includes 13 Resource Types grouped into four categories. And each resource has a set of attributes associated with it. The CDS comes with a default set of controlled vocabularies and automatically matches each vocabulary with the appropriate resource attribute.

In Arches the term "Authority Documents" is used to denote the files that hold the concepts, preferred label, alternate labels, and parent concept (to support hierarchies of concepts) for each drop-down list. Some Authority Documents are common to all Resource Types, while others are particular to specific categories of Resource Types. Here is a list of all of them:

Authority Documents common to all Resource Types:

```
ABSOLUTE AND SCIENTIFIC DATING METHOD AUTHORITY DOCUMENT.csv
ADMINISTRATIVE SUBDIVISION AUTHORITY DOCUMENT.csv
ADMINISTRATIVE SUBDIVISION AUTHORITY DOCUMENT.values.csv
ADMINISTRATIVE SUBDIVISION TYPE AUTHORITY DOCUMENT.csv
ARCHES RESOURCE CROSS-REFERENCE RELATIONSHIP TYPE AUTHORITY DOCUMENT.csv
ARCHES RESOURCE CROSS-REFERENCE RELATIONSHIP TYPE AUTHORITY DOCUMENT.values.csv
CONDITION AUTHORITY DOCUMENT.csv
CONDITION AUTHORITY DOCUMENT.values.csv
CULTURAL PERIOD AUTHORITY DOCUMENT.csv
CULTURAL PERIOD AUTHORITY DOCUMENT.values.csv
DESIGNATION AND PROTECTION TYPE AUTHORITY DOCUMENT.csv
EXTERNAL XREF TYPE AUTHORITY DOCUMENT.csv
GEOMETRY QUALIFIER AUTHORITY DOCUMENT.csv
MATERIAL AUTHORITY DOCUMENT.csv
MEASUREMENT TYPE AUTHORITY DOCUMENT.csv
NAME TYPE AUTHORITY DOCUMENT.csv
SUBJECT AUTHORITY DOCUMENT.csv
TYPE OF DESIGNATION OR PROTECTION AUTHORITY DOCUMENT.csv
TYPE OF DESIGNATION OR PROTECTION AUTHORITY DOCUMENT.values.csv
UNIT OF MEASUREMENT AUTHORITY DOCUMENT.csv
UNIT OF MEASUREMENT AUTHORITY DOCUMENT.values.csv
```

Archaeological Heritage Authority Documents:

```
ARCHAEOLOGICAL COMPONENT TYPE AUTHORITY DOCUMENT.csv
ARCHAEOLOGICAL HERITAGE (ARTIFACT) TYPE AUTHORITY DOCUMENT.csv
ARCHAEOLOGICAL HERITAGE (SITE) TYPE AUTHORITY DOCUMENT.csv
ARCHAEOLOGICAL TECHNIQUE AUTHORITY DOCUMENT.csv
```

Architectural Heritage Authority Documents:

```
ARCHITECTURAL COMPONENT TYPE AUTHORITY DOCUMENT.csv
ARCHITECTURAL HERITAGE TYPE AUTHORITY DOCUMENT.csv
ARCHITECTURAL TECHNIQUE AUTHORITY DOCUMENT.csv
```

Landscape Heritage Authority Documents:

```
LANDSCAPE COMPONENT TYPE AUTHORITY DOCUMENT.csv
LANDSCAPE HERITAGE TYPE AUTHORITY DOCUMENT.csv
LANDSCAPE TECHNIQUE AUTHORITY DOCUMENT.csv
```

Maritime Heritage Authority Documents:

```
MARITIME COMPONENT TYPE AUTHORITY DOCUMENT.csv
MARITIME HERITAGE TYPE AUTHORITY DOCUMENT.csv
MARITIME TECHNIQUE AUTHORITY DOCUMENT.csv
```

Investigation Activity Authority Document:

```
INVESTIGATION TYPE AUTHORITY DOCUMENT.csv
```

Management Activity Authority Document:

```
MANAGEMENT TYPE AUTHORITY DOCUMENT.csv
```

Historical Event Authority Document:

```
HISTORICAL EVENT TYPE AUTHORITY DOCUMENT.csv
```

Document Authority Document:

```
DOCUMENT COMPONENT TYPE AUTHORITY DOCUMENT.csv
DOCUMENT TECHNIQUE AUTHORITY DOCUMENT.csv
DOCUMENT TYPE AUTHORITY DOCUMENT.csv
```

Image Authority Documents:

```
IMAGE COMPONENT TYPE AUTHORITY DOCUMENT.csv
IMAGE MATERIAL AUTHORITY DOCUMENT.csv
IMAGE TECHNIQUE AUTHORITY DOCUMENT.csv
IMAGE TYPE AUTHORITY DOCUMENT.csv
```

Person Authority Document:

```
TITLE AUTHORITY DOCUMENT.csv
```

You can find all the Authority Documents in the following folder:

```
source_data/concepts/authority_files
```

Note: What's up with all the ".E Numbers"?

Resource Types, attributes, and all other entities in the CDS are instances of CIDOC CRM classes. The CRM uses a "E.xx" naming style to define its classes, so we append the CRM class identifier to our CDS entities so that its clear what each CDS entity actually represents.

8.3 Data Import Files

As you've seen, the CDS package comes with the following 13 Resource Types:

- Heritage Resources
  - Archaeological Heritage (element)
  - Archaeological Heritage (site)
  - Architectural Heritage

  - Landscape Heritage
  - Maritime Heritage
- Activities
  - Investigation activity
  - Management activity
  - Designation and protection activity
  - Historical event
- Documents
  - Document
  - Image
- Actors
  - Person
  - Organization

Each of these Resource Types has a set of attributes. Attributes are really just pieces of information about each resource. For example, one of the Attributes of the Archaeological Heritage (Element) resource is its cultural period. In the Arches user interface, Attributes are organized into Information Themes. See the companion **Arches User Guide** for more information on the user interface.

Some attributes are common to many Resource Types. "Name" is a good example, which is common to all Resource Types. Every resource can have a name associated with it. Most Resource Types have quite a few attributes.

Note: Why do I need to know about attributes?

If you already have heritage data in a database or spreadsheet, you can build a file that will let you import your data into Arches. And to do this, you need to know exactly which attributes each resource has.

To see exactly which attributes are associated with a resource, open the this file within the CDS repo:

```
source_data/resource_graphs/CDS attributes.csv
```

Here are the first few records of CDS attributes.csv:

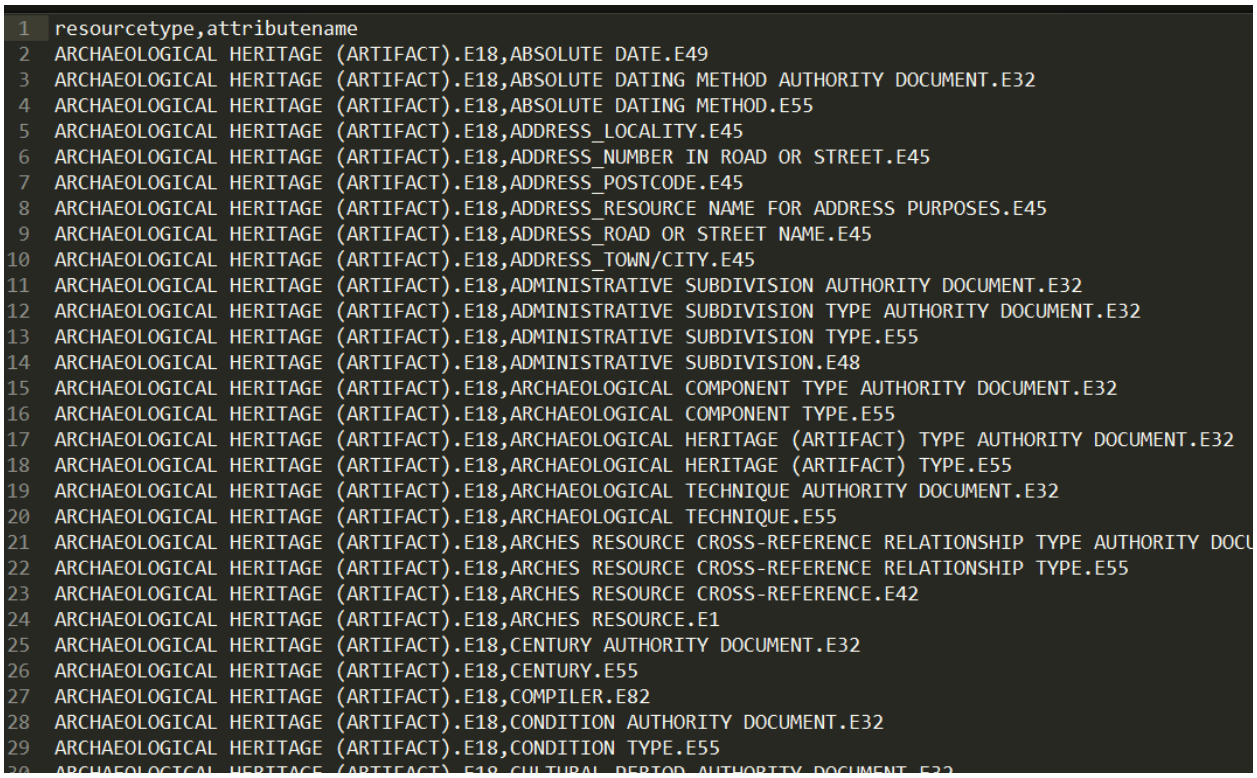
Each line in this file simply states that an Arches resource has an attribute that is associated with it. Notice the three columns:

ResourceType: The Arches Resource Type of the Resource **AttributeName**: The attribute associated with the Resource Type **AuthorityDocument**: If this column is populated, then the attribute value must be a unique identifier found in the authority file that is named in this column.

You can think of the Resource Attributes.txt file as a listing of the set of data that you can load into Arches.

Note: What's up with all the ".E Numbers"?

Resources, attributes, and all other entities in Arches are instances of CIDOC CRM classes. The CRM uses a "E.xx" naming style to define its classes, so we append the CRM class identifier to our Arches entities so that its clear what each Arches entity actually represents.



```
resourcetype,attributename
ARCHAEOLOGICAL HERITAGE (ARTIFACT).E18,ABSOLUTE DATE.E49
ARCHAEOLOGICAL HERITAGE (ARTIFACT).E18,ABSOLUTE DATING METHOD AUTHORITY DOCUMENT.E32
ARCHAEOLOGICAL HERITAGE (ARTIFACT).E18,ABSOLUTE DATING METHOD.E55
ARCHAEOLOGICAL HERITAGE (ARTIFACT).E18,ADDRESS_LOCALITY.E45
ARCHAEOLOGICAL HERITAGE (ARTIFACT).E18,ADDRESS_NUMBER IN ROAD OR STREET.E45
ARCHAEOLOGICAL HERITAGE (ARTIFACT).E18,ADDRESS_POSTCODE.E45
ARCHAEOLOGICAL HERITAGE (ARTIFACT).E18,ADDRESS_RESOURCE NAME FOR ADDRESS PURPOSES.E45
ARCHAEOLOGICAL HERITAGE (ARTIFACT).E18,ADDRESS_ROAD OR STREET NAME.E45
ARCHAEOLOGICAL HERITAGE (ARTIFACT).E18,ADDRESS_TOWN/CITY.E45
ARCHAEOLOGICAL HERITAGE (ARTIFACT).E18,ADMINISTRATIVE SUBDIVISION AUTHORITY DOCUMENT.E32
ARCHAEOLOGICAL HERITAGE (ARTIFACT).E18,ADMINISTRATIVE SUBDIVISION TYPE AUTHORITY DOCUMENT.E32
ARCHAEOLOGICAL HERITAGE (ARTIFACT).E18,ADMINISTRATIVE SUBDIVISION TYPE.E55
ARCHAEOLOGICAL HERITAGE (ARTIFACT).E18,ADMINISTRATIVE SUBDIVISION.E48
ARCHAEOLOGICAL HERITAGE (ARTIFACT).E18,ARCHAEOLOGICAL COMPONENT TYPE AUTHORITY DOCUMENT.E32
ARCHAEOLOGICAL HERITAGE (ARTIFACT).E18,ARCHAEOLOGICAL COMPONENT TYPE.E55
ARCHAEOLOGICAL HERITAGE (ARTIFACT).E18,ARCHAEOLOGICAL HERITAGE (ARTIFACT) TYPE AUTHORITY DOCUMENT.E32
ARCHAEOLOGICAL HERITAGE (ARTIFACT).E18,ARCHAEOLOGICAL HERITAGE (ARTIFACT) TYPE.E55
ARCHAEOLOGICAL HERITAGE (ARTIFACT).E18,ARCHAEOLOGICAL TECHNIQUE AUTHORITY DOCUMENT.E32
ARCHAEOLOGICAL HERITAGE (ARTIFACT).E18,ARCHAEOLOGICAL TECHNIQUE.E55
ARCHAEOLOGICAL HERITAGE (ARTIFACT).E18,ARCHES RESOURCE CROSS-REFERENCE RELATIONSHIP TYPE AUTHORITY DOCU
ARCHAEOLOGICAL HERITAGE (ARTIFACT).E18,ARCHES RESOURCE CROSS-REFERENCE RELATIONSHIP TYPE.E55
ARCHAEOLOGICAL HERITAGE (ARTIFACT).E18,ARCHES RESOURCE CROSS-REFERENCE.E42
ARCHAEOLOGICAL HERITAGE (ARTIFACT).E18,ARCHES RESOURCE.E1
ARCHAEOLOGICAL HERITAGE (ARTIFACT).E18,CENTURY AUTHORITY DOCUMENT.E32
ARCHAEOLOGICAL HERITAGE (ARTIFACT).E18,CENTURY.E55
ARCHAEOLOGICAL HERITAGE (ARTIFACT).E18,COMPILER.E82
ARCHAEOLOGICAL HERITAGE (ARTIFACT).E18,CONDITION AUTHORITY DOCUMENT.E32
ARCHAEOLOGICAL HERITAGE (ARTIFACT).E18,CONDITION TYPE.E55
ARCHAEOLOGICAL HERITAGE (ARTIFACT).E18,CULTURAL PERIOD AUTHORITY DOCUMENT.E32
```

Fig. 3: Resources Attributes.txt

8.3.1 Creating a Data Import File

The CDS comes with a file that you can use to load information into Arches. The is located at:

```
source_data/resource_info.csv
```

The package reads the content of resource_info.csv and loads it into a database. If you open this file from the CDS package, you'll see that it already contains data for a set of heritage resources. You'll need to replace the information in this file with your own data if you want to automatically populate Arches with data.

By the way, don't let the file name fool you, "resource_info.csv" is a | delimited file (a "pipe" delimited file) that illustrates the structure of a valid Arches import file.

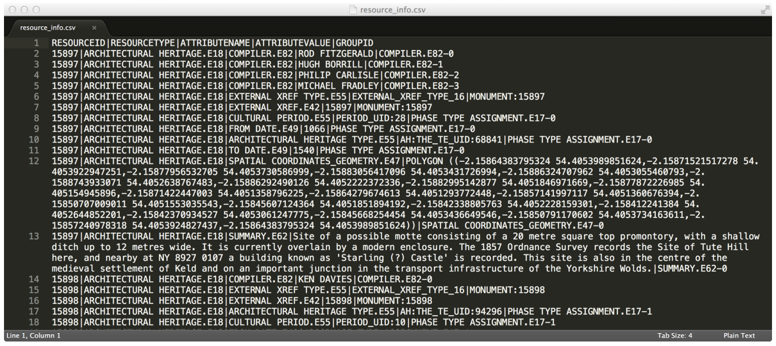
Note: Why use a "|" instead of commas?

The Arches import file allows you to include text blocks (so that you can import free text). Text often includes commas, but rarely includes a |. So we use the | character to distinguish columns in the resource_info.csv file.

Here are some representative records from a load file:

And here's how to understand what's going on: Each line in the file defines a resource and an attribute to load into Arches. Each of the column headers **RESOURCEID**, **RESOURCETYPE**, **ATTRIBUTENAME**, **ATTRIBUTE-VALUE**, and **GROUPID** define the content of the file. Most are self-explanatory, with the exception of the "GROUPID" field (which we'll look at in more detail later).

Let's look at line 1:



```
 1  RESOURCEID|RESOURCETYPE|ATTRIBUTENAME|ATTRIBUTEVALUE|GROUPID
 2  15897|ARCHITECTURAL HERITAGE.E18|COMPILER.E82|ROD FITZGERALD|COMPILER.E82-0
 3  15897|ARCHITECTURAL HERITAGE.E18|COMPILER.E82|HUGH BORRILL|COMPILER.E82-1
 4  15897|ARCHITECTURAL HERITAGE.E18|COMPILER.E82|PHILIP CARLISLE|COMPILER.E82-2
 5  15897|ARCHITECTURAL HERITAGE.E18|COMPILER.E82|MICHAEL FRADLEY|COMPILER.E82-3
 6  15897|ARCHITECTURAL HERITAGE.E18|EXTERNAL XREF TYPE.E55|EXTERNAL_XREF_TYPE_16|MONUMENT:15897
 7  15897|ARCHITECTURAL HERITAGE.E18|EXTERNAL XREF.E42|15897|MONUMENT:15897
 8  15897|ARCHITECTURAL HERITAGE.E18|CULTURAL PERIOD.E55|PERIOD_UID:28|PHASE TYPE ASSIGNMENT.E17-0
 9  15897|ARCHITECTURAL HERITAGE.E18|FROM DATE.E49|1066|PHASE TYPE ASSIGNMENT.E17-0
10  15897|ARCHITECTURAL HERITAGE.E18|ARCHITECTURAL HERITAGE TYPE.E55|AH:THE_TE_UID:68841|PHASE TYPE ASSIGNMENT.E17-0
11  15897|ARCHITECTURAL HERITAGE.E18|TO DATE.E49|1540|PHASE TYPE ASSIGNMENT.E17-0
12  15897|ARCHITECTURAL HERITAGE.E18|SPATIAL COORDINATES_GEOMETRY.E47|POLYGON ((-2.15864383795324 54.4053989851624,-2.15871521517278 54.
    4053922947251,-2.15877956532705 54.4053730586999,-2.15883056417096 54.4053431726994,-2.15886324707962 54.4053055460793,-2.
    1588743933071 54.4052638767483,-2.15886292490126 54.4052222372336,-2.15882995142877 54.4051846971669,-2.15877872226985 54.
    405154945896,-2.15871422447003 54.4051358796225,-2.15864279674613 54.4051293772448,-2.15857141997117 54.4051360676394,-2.
    15850707009011 54.4051553035543,-2.15845607124364 54.4051851894192,-2.15842338805763 54.4052228159301,-2.158412241384 54.
    4052644852201,-2.15842370934527 54.4053061247775,-2.15845668254454 54.4053436649546,-2.15850791170602 54.4053734163611,-2.
    15857240978318 54.4053924827437,-2.15864383795324 54.4053989851624))|SPATIAL COORDINATES_GEOMETRY.E47-0
13  15897|ARCHITECTURAL HERITAGE.E18|SUMMARY.E62|Site of a possible motte consisting of a 20 metre square top promontory, with a shallow
    ditch up to 12 metres wide. It is currently overlain by a modern enclosure. The 1857 Ordnance Survey records the Site of Tute Hill
    here, and nearby at NY 8927 0107 a building known as 'Starling (?) Castle' is recorded. This site is also in the centre of the
    medieval settlement of Keld and on an important junction in the transport infrastructure of the Yorkshire Wolds.|SUMMARY.E62-0
14  15898|ARCHITECTURAL HERITAGE.E18|COMPILER.E82|KEN DAVIES|COMPILER.E82-0
15  15898|ARCHITECTURAL HERITAGE.E18|EXTERNAL XREF TYPE.E55|EXTERNAL_XREF_TYPE_16|MONUMENT:15898
16  15898|ARCHITECTURAL HERITAGE.E18|EXTERNAL XREF.E42|15898|MONUMENT:15898
17  15898|ARCHITECTURAL HERITAGE.E18|ARCHITECTURAL HERITAGE TYPE.E55|AH:THE_TE_UID:94296|PHASE TYPE ASSIGNMENT.E17-1
18  15898|ARCHITECTURAL HERITAGE.E18|CULTURAL PERIOD.E55|PERIOD_UID:10|PHASE TYPE ASSIGNMENT.E17-1
```

Fig. 4: Resource_Info.csv

```
15897|ARCHITECTURAL HERITAGE.E18|COMPILER.E82|ROD FITZGERALD|COMPILER.E82-0
```

This record tells Arches that:

- We're going to load information about an "ARCHITECTURAL HERITAGE.E18" resource (**RESOURCETYPE**)
- Our resource can be uniquely identified by the string "15897" (**RESOURCEID**)
- We're going to load a value for the "COMPILER.E82" attribute (**ATTRIBUTENAME**)
- The value of the Compiler.E82 attribute is "ROD FITZGERALD" (**ATTRIBUTEVALUE**)
- This record is part of a group of records identified as "COMPILER.E82-0" (**GROUPID**)

Notice that we can import lots of information about a single resource simply by referencing the same **RESOURCEID** and **RESOURCETYPE**.

8.3.2 Adding Multiple Values to an Attribute

Why all this formalism and complexity?

Because many cultural heritage objects have more than value for a given attribute. For example, a single resource can often have many names. Or the characteristics of a resource may change over time. Without allowing for multiple values for an attribute, Arches wouldn't be able to track the evolution of a resource.

So, the Arches data import file structure is important because many resource attributes may have multiple values. Indeed, an Architectural Heritage resource may be associated with several cultural periods, have many addresses, and have several protection grades.

That's why Arches allows you load many attribute values for a single resource attribute.

Notice in the file shown that we can add several compilers into Arches for the same resource. In fact, lines 3, 4, and 5 also define compilation records. In general, Arches will allow you to add multiple values for a given attribute.

Note: Single Attributes

At present, Arches' user interface can show multiple values for all entities except for the Summary Description, Distinguishing Features, and Location Description entities.

8.3.3 Attribute Goups

Now look at lines 8 through 11 in the example file.

Individually, each row assigns a value for a specific attribute. The FROM DATE.E49 and TO DATE.E49 attributes are years. Line 8 seems to be describing a Cultural Period, and Line 10 the Architectural Resource type.

But notice that each record in rows 8 through 11 all share the same "PHASE TYPE ASSIGNMENT.E17-0" **GROUPID**. This means that these 4 records together describe a resource.

By grouping these rows together with the same **GROUPID**, you can tell Arches that these 4 records are not independent; rather together they describe the resource.

Here's how to interpret what's going on. Each of the records starts with:

```
15897|ARCHITECTURAL HERITAGE.E18
```

This means that each record contains information about the Architectural Heritage.E18 Resource with the unique identifier of 15897. Okay, let's focus on what comes next:

```
 8  CULTURAL PERIOD.E55|PERIOD_UID:28|PHASE TYPE ASSIGNMENT.E17-0
 9  FROM DATE.E49|1066|PHASE TYPE ASSIGNMENT.E17-0
10  ARCHITECTURAL HERITAGE TYPE.E55|AH:THE_TE_UID:68841|PHASE TYPE ASSIGNMENT.E17-0
11  TO DATE.E49|1540|PHASE TYPE ASSIGNMENT.E17-0
```

Look at line 8. Its **ATTRIBUTENAME** is "CULTURAL PERIOD.E55", and the value of this attribute is "PERIOD_UID:28". Recall from the Chapter "Step 3: Load Controlled Vocabularies" that we use an Authority Document for Cultural Periods. So, the **ATTRIBUTEVALUE** "PERIOD_UID:28" in line 8 is a pointer to a concept in CULTURAL PERIOD AUTHORITY DOCUMENT.csv (which works out to "Medieval").

OK, now look at line 9. Its **ATTRIBUTENAME** is "FROM DATE.E49" and its **ATTRIBUTEVALUE** is "1066". Simple enough: "from date" is the year 1066.

Line 10 tells Arches to use the unique identifier "AH:THE_TE_UID:68841" in ARCHITECTURAL HERITAGE TYPE AUTHORITY DOCUMENT.csv to define the value for "ARCHITECTURAL HERITAGE TYPE.E55" (a "Motte"), and line 11 says that the resource has a "TO DATE.E49" value of "1540".

And here's the payoff: by giving these 4 records the same **GROUPID**, we are telling Arches that:

> "The ARCHITECTURAL HERITAGE.E18 resource with unique identifier of "15897" was a "Motte" during the "Medieval" cultural period, specifically between the years 1066 and 1540."

Recall that Arches will support many combinations of cultural period, heritage type, and from/to dates for a single resource.

Now, we can understand that the purpose of the **GROUPID** is to allow for grouping a set of related attributes into a coherent record.

The actual values that you use for **GROUPID** are arbitrary, just make sure that you use a unique value for each set of attributes that you want to group.

One last note on **GROUPID** and resource_info: notice that in this resource_info file records with the same **GROUPID** are grouped together(located on adjacent lines). This grouping within the resource_info file is necessary to ensure your grouped attributes are not duplicated upon import into Arches.

Note: Required Data

By design, Arches can accommodate a wide variety of data. The only real data requirement is this: **If you want to name a resource, you must provide one name with a NAME TYPE.E55 of "Primary."** Because you can give a resource as many names as you want, Arches uses the "Primary" designation to identify the preferred name of the resource.

8.3.4 Which Attributes can be Grouped?

Not all attributes can (or should) be grouped together. Based on the *Core Data Standard for Archaeological and Architectural Heritage* (introduced in the "Deploying Arches" chapter), Arches allows for the following attributes to be grouped:

- **NAME.E41** and **NAME TYPE.E55**
- **SPATIAL COORDINATES_GEOMETRY.E47** and **SPATIAL COORDINATES QUALIFIER**
- Any combination of attributes starting with **ADRESS_**
- Any combination of **PERIOD AUTHORITY DOCUMENT**, **FROM DATE.E49**, **TO DATE.E49**, and **HERITAGE RESOURCE TYPE AUTHORITY DOCUMENT** (e.g.: Archaeological Heritage (Element), Archaeological Heritage (Site), Architectural Heritage, Landscape Heritage, Maritime Heritgae)

8.3.5 Loading Data

Loading data into Arches requires three things. Those are:

1. Validly Structured and Populated Resource Graphs

   for defining the resources, their attributes, and how they are related to each other

2. Validly Structured Authority Files and ENTITY_TYPE_X_ADOC file

   for defining the values that can be associated with attributes constrained by controlled vocbularies

3. A resource_info.csv file with content that adheres to the constraints imposed by the resource graphs and the authority files.

   resource_info contains the business data that is to be imported to the Arches database.

Once you've prepared your data load file, you may insert its contents into Arches. Be sure to confirm that you are referencing your Authority Documents properly (e.g.: the unique ID you use in your load file should also exist in the appropriate Authority Document).

Arches will load and index each of the cultural heritage data records that you've defined in your data file. Don't worry if it takes a few attempts to get your data to load cleanly. You can always re-run the Arches build scripts to get a pristine instance of Arches.

CHAPTER 9

Loading the CDS Package With Sample Data

Arches manages cultual heritage resources through the use of packages. A package defines the types of resources that Arches should manage and the resources' attributes. This can include things like the web based forms neccessary to create and edit resources, reports, and services to support search. In reality, though, it can include almost anything the package developer wants to add into Arches.

You will need to load at least one Arches Package into your Arches Server to begin managing a cultural heritage inventory. These are these steps demonstrate the process for the CDS Package.

9.1 Step 1: Download the CDS Package

Get a copy of the CDS package by going to the Packages section at http://archesproject.org/download-arches/

Create a folder called "cds" in archesproject/packages and download the code to that folder (unzip if neccesary)

I>you should have a directory structure simlar to this:

```
archesproject/
    arches/
    ...
    packages/
        cds/
            install/
            media/
            models/
            ...
    settings.py
```

9.2 Step 2: Install the Package

Open the archesproject/settings.py in your favorite text editor.

Add an entry in the INSTALLED_PACKAGES setting coresponding to the name of the folder you created to hold the CDS Package in the packages directory (following our example above, it would look like this)

```
INSTALLED_PACKAGES = (
    # entries here should correspond to folders within the packages directory
    'cds',
)
```

I> Don't forget the trailing comma for a single entry in a python tuple

If you don't want to load the test data that comes with the package, open the package's settings.py file (don't confuse this with the Arches settings.py file from above) and set **LOAD_TEST_DATA = False**

Open a command window, and navigate to:

```
archesproject/build
```

Run the following script to install the CDS package:

```
install_packages.bat (on Windows)

OR
install_packages.sh (on Linux)
```

Congatulations! You've just completed the steps needed to install the core Arches application and CDS package on your server. This package comes loaded with sample data provided by English Heritage.

Go to the following URL and check it out!

```
http://localhost:8000/Arches/index.htm
```

CHAPTER 10

Loading the CDS Package With Your Own Data

Remember, a package defines the available resource types and attributes that you can use to describe resources within those types. What it does not define is the content of the data itself. Therefore, it is possible to "map" your own data into the resource types and attributes that are defined in the CDS Package.

The button pushing for loading the package with your own data is identical to the workflow described above. However, before you "Install the Package", replace the content of the source data files with your own custom content.

Note: If you do this *after* you "Install the Package," then your changes will not be reflected in the Arches database. This must be done *before* you install the package.

There are two types of source data files to concern yourself with in this process 1. Authority Documents 2. resource_info

10.1 Step 1: Modify Authority Documents

The purpose and structure of authority documents are described in some detail in Section 7 of this document. First and foremost, populate the authority document files with the content that you wish to use for the attributes that are constrained by controlled vocabularies.

If you want to do anything fancy with your controlled vocabularies like associate them with geometries, add attributes supporting attributes to them, or arrange them hierarchically, then take a very close look at Chapter 7.

10.2 Step 2: Populate resource_info

The purpose and structure of resource_info is described in some detail in Section 7 of this document. The key points to remember as you construct your resource_info file are the following:

1. Make sure that the data in the attributename column is constrained to valid attributename values from "CDS attributes.csv". This file defines the set of resource types and attributes for given resource types within the CDS package.

2. Make sure that the attributevalue for any attributename ending with '.E55' is a valid conceptid as defined in one of the authority files... that you ostensibly edited.

Be sure to read up on information from Chapter 7 describing the purpose of resourceid, resourcetype, and groupid (especially groupid) columns within the resource_info file.

10.3 Step 3: Install the Package

Follow the direction outlined in Step 2 in the first workflow of this chapter.

10.3.1 Lastly

There may be a point at which the resource types and their respective attributes do not sufficiently meet the business need that you are using Arches to fulfill. That is the point when you need create or use a differnt "package." See chapter XXX for guidance on defining and building your own Arches package.

CHAPTER 11

Running and Configuring Arches

In this chapter you'll learn how to define important configuration settings and define user accounts. You will have the ability to customize Arches, but also the potential to alter the application in profound ways. Make sure that you back-up all the default settings before you commit your changes!

11.1 Running Arches

Starting Arches is easy: you just have to make sure that you run the "runserver" command in the Arches root folder. Stopping Arches is also easy: just press Control-C in the terminal window that you used to start Arches.

By default, users can log onto Arches (from the host server) with the following URL:

```
http://localhost:8000/Arches/index.htm#
```

You may wish to configure your web server to support access for users from client machines and the internet. This will require you to replace the "localhost:8000" part of the URL with a domain or IP address.

Based on the resources (RAM, storage, CPU(s)) of the server that you've installed Arches on, you may wish to tune how you allocate memory to PostgreSQL (the underlying Arches database) and your web server (e.g.: Apache or IIS). You can find guidance on these topics by visiting the PostgreSQL, Microsoft, or Linux web sites.

11.2 Running Arches with Apache

If you plan on running Arches in a production setting then it is highly recommended that you use Apache (or some other production quality webserver).

Warning: Do not use the in built django web server in production. See the Django docs for more information about using django with Apache

11.2.1 Step 1: Download and install Apache

Windows

Go to the Apache web site, and follow the instructions for downloading and installing Apache for Windows.

Ubuntu

From a terminal window type the following to install apache:

```
sudo apt-get update; sudo apt-get install apache2
```

To confirm that Apache installed correctly, you should be able to go to http://localhost in your browser and get the standard "**It Works!**" page like:

```
It works!
This is the default web page for this server.
The web server software is running but no content has been added, yet.
```

11.2.2 Step 2: Download and install mod_wsgi for Apache

Windows

Go to the installation instructions page and download and install the mod_wsgi module for Windows Binaries.

Ubuntu

From a terminal type the following to install mod_wsgi:

```
sudo apt-get install libapache2-mod-wsgi
sudo nano /etc/apache2/mods-available/wsgi.load
```

add the next string to the file

```
LoadModule wsgi_module /usr/lib/apache2/modules/mod_wsgi.so
```

save, then

```
sudo a2enmod wsgi
sudo service apache2 restart
```

11.2.3 Step 3: Edit your Apach httpd.conf file

Windows and Ubuntu

Add the following lines to your httpd.conf (don't forget to replace the dummy path names below to the ones corresponding to your setup)

```
WSGIScriptAlias /<base_url_name>/path/to/arches_web/wsgi.py
WSGIPythonPath /path/to/arches_web/archesproject/virtualenv/ENV/lib/python2.7/site-
↪packages

<Directory /path/to/arches_web/>
<Files wsgi.py>
Order deny,allow
Require all granted
</Files>
</Directory>
```

Note: General Notes

Using the example above, your site would be available from the following url: http://localhost/base_url_name/Arches

Don't forget to restart apache after making these modifications

Note: Ubuntu Notes

Make sure that apache has (chmod 755) priveleges to all the directories down to the location where your Arches codebase resides

11.2.4 Step 4: Edit your settings.py file

In your favorite editor, open the archesproject/settings.py file and edit the following variables:

- Point the STATIC_ROOT variable to the location in apache were static files are served.
- Point the STATIC_URL variable to name of the folder you create to hold the static files (see below)

Windows

STATIC_ROOT = 'C:/Program Files/Apache Software Foundation/Apache2.2/htdocs/media/' <– make sure to create the media directory if it doesn't already exist, also make sure the path is correct STATIC_URL = '/media/'

Ubuntu

STATIC_ROOT = '/var/www/media/' <– make sure to create the media directory if it doesn't already exist STATIC_URL = '/media/'

After doing this don't forget to run

```
archesproject/build/build(.bat/.sh)
```

This will copy all the files in archesproject/arches/Media folder to the directory you created above (in this example, called 'media')

You're Done! Go to http://localhost/base_url_name/Arches to browse the Arches site.

11.3 Arches Configuration

Arches is really two applications in one:

- A map-based cultural heritage data management tool
- A searchable online inventory of cultural heritage

Before you start using Arches, you'll probably need to update a few configuration settings. Perhaps the most important of these settings are:

- Managing user accounts
- Updating default map settings (mapConfig)
- Managing basemaps and GIS layers
- Managing resource layers

Arches implements Django, a high-level Python Web framework that supports rapid and robust application development. One of several nice features that Django offers is a simple Administration Console.

Arches can be configured using the Django Administration Console. You may access the console by typing:

```
http://localhost:8000/admin
```

Be sure to replace "localhost:8000" with the domain name or IP address of the server hosting your version of Arches.

Next, you'll need to log in. Arches comes with the following default account:

```
user: admin
password: admin
```

Note: Change the default "admin" password!

Its too easy for someone to guess this password! See the section on managing user accounts and change the default password for this account as soon as possible!

Once you've signed in to the Django Administration Console, you should see a screen that looks something like this:

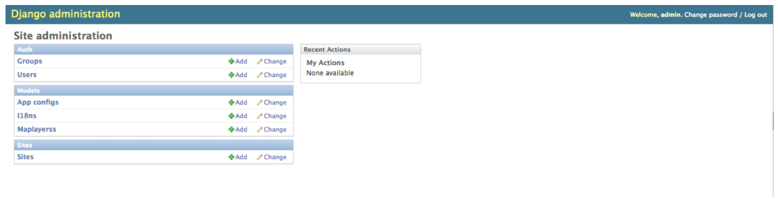


Fig. 1: Django's Site Administration Console

This interface will let you Manage users, define the state of the map display in Arches, configure resources for display, and add GIS maps to Arches.

11.4 Managing User Accounts

Arches implements a simple user access model with the following 3 tiers of users:

- Guests

- Editors
- Administrators

Guests can access Arches and view information (either via the search screens or through the Map screen) without having to sign on to the system.

Editors require a user account to add, edit, or delete data.

Administrators may configure Arches and manage users.

11.4.1 Updating the Default Administrator Password

Arches automatically creates a single Administrator account with the following credentials:

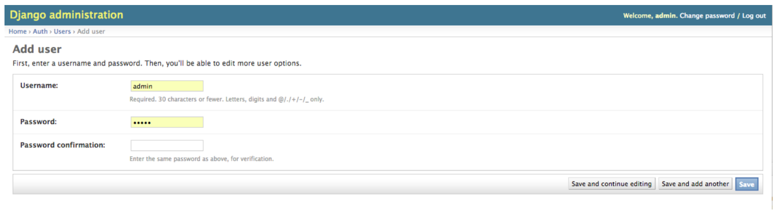
```
user: admin
password: admin
```

You will definitely want to change the default password. To do so, log in and click the “Change password” link at the upper right. You’ll be prompted to type in your current password (“admin” in this case), as well as a new password.

Click the “Change my password” button when you’re done (and be sure that you don’t forget your new password!).

11.4.2 Adding a New Editor Account

From the Django Administration Console, simply click the “Add” button next to “Users”. You’ll see a screen that looks something like this:


Django administration
Welcome, admin. Change password / Log out
Home › Auth › Users › Add user
Save and continue editing
Save and add another
Save

Add user

First, enter a username and password. Then, you'll be able to edit more user options.

Username: admin

Required. 30 characters or fewer. Letters, digits and @/./+/-/_ only.

Password: •••••

Password confirmation:

Enter the same password as above, for verification.

Fig. 2: New User Input Form

Add the Username and Password for the user account, then click the “Save” button. Django will confirm that you are adding a unique user name and valid password. If so, you’ll see the following screen:

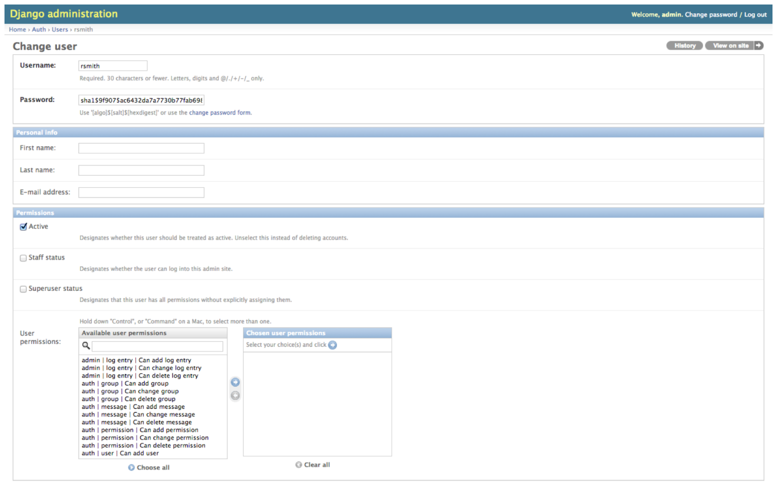
At this point, you can click the “Save” button at the bottom of the form to create a user account with data-editing privileges.

Note: Optional User Information

The “Change user” screen allows you to add additional information about the user, such as name and e-mail address. You may optionally also grant specific administrative permissions to this account in the “User permissions” section of the screen. Arches only requires a unique username and valid password to create an Editor account.

Feel free to explore the User adminstration functions available in the Django Console. You’ll see that it provides a suite of easy-to-use tools for managing your Arches users.

One thing to note: the Django Administrator Console allows you to define Groups. As of September 2013, Arches does not recognize Django Groups for authentication.



Change user

Username: rsmith
Required. 30 characters or fewer. Letters, digits and @/./+/-/_ only.

Password: sha1$9f907$ac6432da7a7730b77fab69
Use '[algo]$[salt]$[hexdigest]' or use the change password form.

First name:

Last name:

E-mail address:

[✓] Active
Designates whether this user should be treated as active. Unselect this instead of deleting accounts.

[ ] Staff status
Designates whether the user can log into this admin site.

[ ] Superuser status
Designates that this user has all permissions without explicitly assigning them.

User permissions:
Hold down "Control", or "Command" on a Mac, to select more than one.

Available user permissions

```
admin | log entry | Can add log entry
admin | log entry | Can change log entry
admin | log entry | Can delete log entry
auth | group | Can add group
auth | group | Can change group
auth | group | Can delete group
auth | message | Can add message
auth | message | Can change message
auth | message | Can delete message
auth | permission | Can add permission
auth | permission | Can change permission
auth | permission | Can delete permission
auth | user | Can add user
```

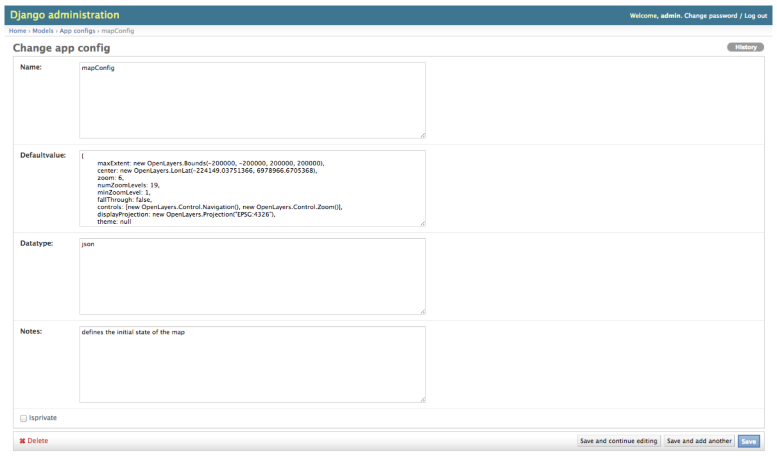
Chosen user permissions
Select your choice(s) and click

Fig. 3: New User Detais

11.5 Setting Default Map Settings

Now that you have Arches running, you'll probably want to configure Arches for your particular geographic area of interest. You do this by defining the map center point and initial zoom scale that you want Arches to use.

From the main Django Administration Console, **Click App configs** and then select **mapConfig** to access the configuration settings for the map display in Arches display. You should see the following screen:


Django administration
Welcome, admin. Change password / Log out
Home › Models › App configs › mapConfig
History
Delete
Save and continue editing
Save and add another
Save

Change app config

Name: mapConfig

Defaultvalue:

```
{
        maxExtent: new OpenLayers.Bounds(-200000, -200000, 200000, 200000),
        center: new OpenLayers.LonLat(-224149.03751366, 6978966.6705368),
        zoom: 6,
        numZoomLevels: 19,
        minZoomLevel: 1,
        fallThrough: false,
        controls: [new OpenLayers.Control.Navigation(), new OpenLayers.Control.Zoom()],
        displayProjection: new OpenLayers.Projection("EPSG:4326"),
        theme: null
```

Datatype: json

Notes: defines the initial state of the map

[ ] Isprivate

Fig. 4: mapConfig

Notice that this page lists the variable (Name: mapConfig), its Defaultvalue, Datatype, and associated Notes. We're most interested in the Defaultvalue:

```
{
 maxExtent: new OpenLayers.Bounds(-200000, -200000, 200000, 200000),
 center: new OpenLayers.LonLat(-224149.03751366, 6978966.6705368),
 zoom: 6,
 numZoomLevels: 19,
 minZoomLevel: 1,
 fallThrough: false,
```

(continues on next page)

(continued from previous page)

```
 controls: [new OpenLayers.Control.Navigation(), new OpenLayers.Control.Zoom()],
 displayProjection: new OpenLayers.Projection("EPSG:4326"),
 theme: null
}
```

In particular, notice that Defaultvalue includes a definition for the map center ("center") and initial zoom level ("zoom").

This:

```
center: new OpenLayers.LonLat(-224149.03751366, 6978966.6705368),
zoom: 6,
```

tells Arches where to set the center of the map and the initial extent to use for the map that you will use to create and display your cultural heritage information.

The following coordinates set the map center point:

```
-224149.03751366, 6978966.6705368
```

By default, Arches centers the map near Nottingham, England. The -224149.03751366, 6978966.6705368 numbers are the coordinates for this point in the Spherical Mercator projection.

To recenter the map, replace the default coordinates with the proper coordinates for your preferred area. You can use this website http://www.maptiler.org/google-maps-coordinates-tile-bounds-projection/ to determine the proper coordinates if you don't know them by heart.

Note: Pro Tip: Getting Coordinates from the Arches Map

If you're comfortable using the developer tools that come with Google Chrome, Firefox, or Safari, you can have Arches tell you the map center point. Simply navigate the map to your area of interest and type "arches.appPanel.mapPanel.map.getCenter().toShortString()" in the JavaScript console. Arches will return the center point of the current map.

Once you've determined the center coordinates and map zoom level that you want to use, simply replace the default values with the coordinates and zoom level for your area of interest.

Note: Save a copy of the default values!

You may wish to copy and paste the Defaultvalue to a text file before making any changes to it. This way, you can be sure and recover to a working map if you run into problems updating the map center coordinates.

Click the save button when you are done. To update Arches with this new information, navigate to the arches/build folder and run:

```
build.bat (Windows) or ./build.sh (Linux)
```

Refresh your browser (you may have to clear your cache) and navigate to the Map screen in Arches to verify your new values.

11.6 Managing Basemaps and External GIS Layers

Arches lets you define the basemaps and GIS layers (technically map services) that your users will be able to work with.

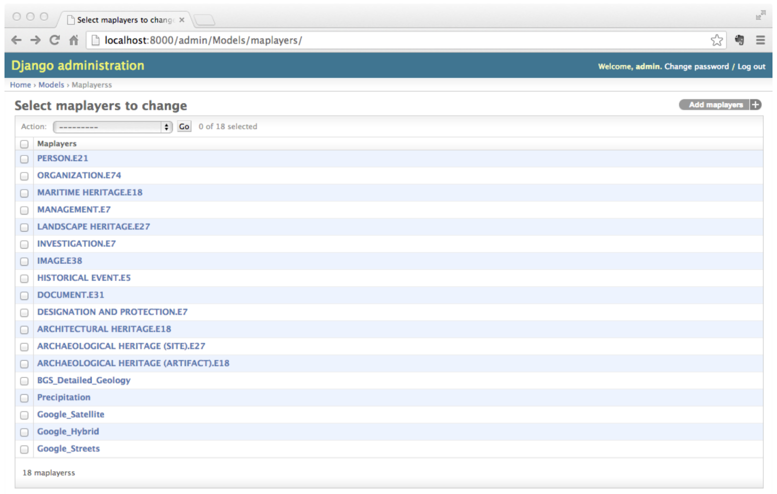
Basemaps are the map(s) that your users will select to view the location of resources. Google Streets, Bing Satellite, and OpenStreetMap are all examples of commonly used basemaps for web applications.

GIS map services are maps that can be geo-referenced over a basemap, allowing users to compare geographic information (such as zoning districts or flood-prone areas) to both the basemap and resources. Common examples include maps from data providers such as regional or local govenerments.

In either case, Arches uses the OpenLayers library (http://openlayers.org) to manage the display of basemaps and GIS map services.

11.6.1 Adding a new Basemap/GIS Service to Arches

Open the Arches Administrative console and select Maplayerss.



Select maplayers to change

localhost:8000/admin/Models/maplayers/

Django administration

Welcome, admin. Change password / Log out

Home › Models › Maplayerss

Select maplayers to change

Add maplayers

Action: --------- Go 0 of 18 selected

Maplayers
PERSON.E21
ORGANIZATION.E74
MARITIME HERITAGE.E18
MANAGEMENT.E7
LANDSCAPE HERITAGE.E27
INVESTIGATION.E7
IMAGE.E38
HISTORICAL EVENT.E5
DOCUMENT.E31
DESIGNATION AND PROTECTION.E7
ARCHITECTURAL HERITAGE.E18
ARCHAEOLOGICAL HERITAGE (SITE).E27
ARCHAEOLOGICAL HERITAGE (ARTIFACT).E18
BGS_Detailed_Geology
Precipitation
Google_Satellite
Google_Hybrid
Google_Streets

18 maplayerss

Fig. 5: A listing of the resource map layers

Arches uses Google basemaps by default. But you can add additional basemaps or replace the default maps if you wish.

Note that Arches includes a Map Layer for every resource (e.g.: the map layers ending in E.xx), as well as GIS mapservices and basemaps (e.g.; Precipitation, Google_Hybrid).

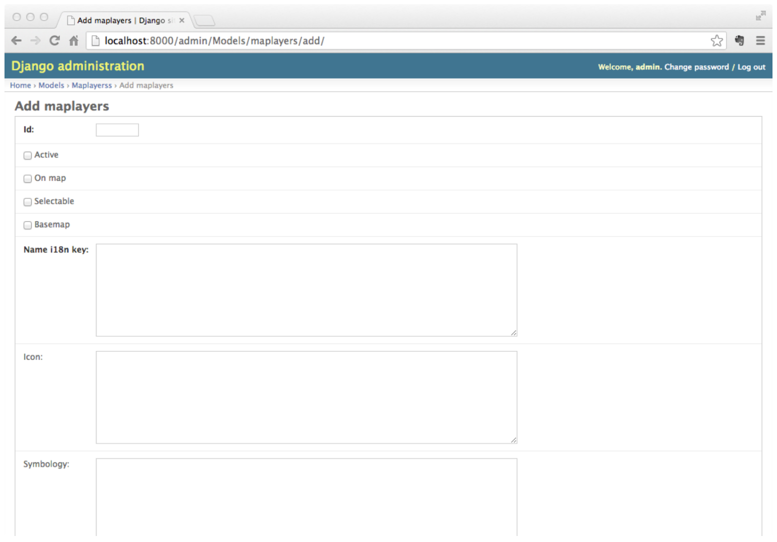
Really, the only technical difference between a basemap and a GIS mapservice is that at least one basemap must be visible at all times.

Press the "Add maplayers" button in the upper right corner. You'll see a data entry form that looks something like:

Step 1: Add Id Each Arches map layer needs to have a unique identifier. Add a number that is not currently being used to identify your new External GIS Layer.

Arches comes with several map layers by default, and map layer id's start at 1. In our example, Arches already has 18 map layers, so the next available number would be 20 (1 + 18 = 19 layers already sequentially indexed).

If you're not sure what number to use, try a number like 1000. Don't worry, Django's administration console will tell you if you pick a number that already being used by Arches.


Add maplayers | Django s
localhost:8000/admin/Models/maplayers/add/
Django administration
Welcome, admin. Change password / Log out
Home › Models › Maplayerss › Add maplayers

Add maplayers

Id:

Active

On map

Selectable

Basemap

Name i18n key:

Icon:

Symbology:

Fig. 6: Add Maplayer Form

Step 2: Identify as Basemap or GIS Mapservice Layer Next decide if you want to create a basemap or a GIS map service. The Arches user interface lists basemaps in a separate grid, allowing users to toggle between maps.

In contrast, GIS map services appear as "External Services" in the Map Layers grid. Users may display as many maps from the Map Layers grid as they wish.

If you want to create a new basemap, click the "Basemap" check box.

Step 3: Name your Layer Type the name of your new map layer into the "Name i18n key" field. Pick a simple, friendly, and short name as Arches will use it in the list of map layers available for your users.

Step 4: Define the icon for your layer Type the name of the icon you want to associate with your new layer into the "Icon" field. Arches will look for your icon in:

```
arches/Arches/Media/images/Asseticons
```

Step 5: Define the Layer Group for your Layer Arches organizes maps into groups. Basemaps are members of the "Basemaps" group. GIS mapservices are members of the "External_Services" group.

Step 6: Define the Layer Arches can display any layer type supported by OpenLayers version 2.13. Luckily, this is a pretty long list. Check out the OpenLayers API (click on "Layer") to see the full list of 30 or so layers that can be added to Arches.

For example, to add an esri map service to Arches, you could type:

```
return new OpenLayers.Layer.XYZ("ESRI Topo",
"http://server.arcgisonline.com/ArcGIS/rest/services/World_Topo_Map/MapServer/
↪tile/${z}/${y}/${x}",
  {
    sphericalMercator: true,
    isBaseLayer: false,
    zoomOffset: 1
  }
);
```

Arches passes this snippet of JavaScript to OpenLayers for rendering map services. The key elements here are:

- The name of the service: "ESRI Topo"

- Service URL: "http://server.arcgisonline.com/ArcGIS/rest/services/World_Topo_Map/MapServer/tile/${z}/${y}/${x}"
- sphericalMercator: true. This is a flag that tells OpenLayers to use the common webmapping projection
- isBaseLayer: false. Overrides the ESRI default.

You may wish to consult the OpenLayers documentation for a more detailed summary of map service parameters.

Step 7: Save Press the "Save" button to complete the layer definition.

Step 8: Register your Layer Name with Arches

From the console home page, click on i18ns. Then click on the "Add i18n" button in the upper right part of the page.

Note: What is i18n?

i18n is quick way to refer to "internationalization" (i +18 letters + n). Arches supports multiple languages for its user interface (UI), and uses i18n to keep track of each label (and its language)in the UI.

Type in the name of your layer (entered in Step 3 above) into the "key" field. Enter the label you want the Arches UI to show for your layer in the "value" field.

Next, enter the language identifier for your layer name in the "Languageid" field. Use

```
en-us
```

for the version of English spoken in the United States. You can consult the web to determine the specific code if you want to use a language other than English.

Finally, enter the following string:

```
Arches.widgets.LayerLibrary
```

into the Widgetname field. This tells Arches where in the UI to place your label. Press the "save" button.

You've just completed the steps necessary to add new basemaps or map layers to Arches. You should see a new entry in the i18n form that looks something like:

You'll need to complete Steps 1 through Step 8 for each layer that you wish to add to Arches

Step 9: Update Arches Once you've completed entering your layer information, you'll need to re-build Arches. Go to the terminal window running Arches and press the Contol button and C button:

```
control-C
```

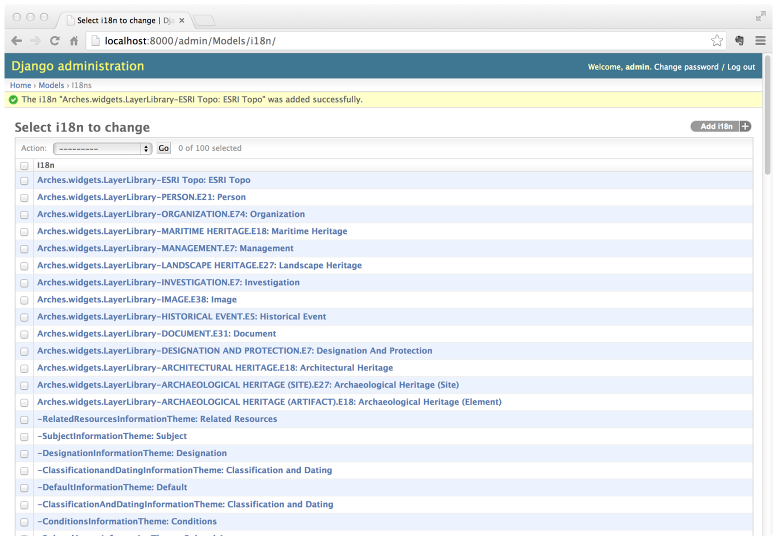
To stop the Arches server. Navigate to:

```
arches/build
```

and run the build script:

```
build.bat (./build.sh in Linux)
```

This script will update Arches to include your new layers. Once the script completes (usually in just a minute or two), navigate to the Arches root folder and start Arches:



Select i18n to change | Dja

localhost:8000/admin/Models/i18n/

Django administration

Welcome, **admin**. Change password / Log out

Home › Models › I18ns

The i18n "Arches.widgets.LayerLibrary–ESRI Topo: ESRI Topo" was added successfully.

Select i18n to change

Add i18n

Action: --------- Go 0 of 100 selected

I18n
Arches.widgets.LayerLibrary–ESRI Topo: ESRI Topo
Arches.widgets.LayerLibrary–PERSON.E21: Person
Arches.widgets.LayerLibrary–ORGANIZATION.E74: Organization
Arches.widgets.LayerLibrary–MARITIME HERITAGE.E18: Maritime Heritage
Arches.widgets.LayerLibrary–MANAGEMENT.E7: Management
Arches.widgets.LayerLibrary–LANDSCAPE HERITAGE.E27: Landscape Heritage
Arches.widgets.LayerLibrary–INVESTIGATION.E7: Investigation
Arches.widgets.LayerLibrary–IMAGE.E38: Image
Arches.widgets.LayerLibrary–HISTORICAL EVENT.E5: Historical Event
Arches.widgets.LayerLibrary–DOCUMENT.E31: Document
Arches.widgets.LayerLibrary–DESIGNATION AND PROTECTION.E7: Designation And Protection
Arches.widgets.LayerLibrary–ARCHITECTURAL HERITAGE.E18: Architectural Heritage
Arches.widgets.LayerLibrary–ARCHAEOLOGICAL HERITAGE (SITE).E27: Archaeological Heritage (Site)
Arches.widgets.LayerLibrary–ARCHAEOLOGICAL HERITAGE (ARTIFACT).E18: Archaeological Heritage (Element)
–RelatedResourcesInformationTheme: Related Resources
–SubjectInformationTheme: Subject
–DesignationInformationTheme: Designation
–ClassificationandDatingInformationTheme: Classification and Dating
–DefaultInformationTheme: Default
–ClassificationAndDatingInformationTheme: Classification and Dating
–ConditionsInformationTheme: Conditions

Fig. 7: New Map Layer Label

```
runserver.bat (./runserver.sh in Linux)
```

Log back on to Arches (you may need to clear your browser's cache), and open the Basemaps Widget or the Map Layers widget to see your new maps.

11.6.2 Removing a GIS Map Service from Arches

Removing an existing basemap or GIS map service layer is simple. Just open the maplayerss console, click the check box next to the map layers that you want to remove, Select "delete selected maplayerss" from the "action" dropdown, then press the "go" button.

Django will prompt you to confirm your selection. Click "Yes, I'm sure" to delete the selected layers.

11.7 Cosmetic Settings

Arches comes with some settings that are purely a matter of taste. Things like the default image on the search page, and the colors we use to symbolize resource geometries on a map.

You can easily update Arches to use images, colors, and geometry styles that you find more attractive.

11.7.1 Changing the Image on the Arches Splash Screen

You can easily replace the image of Stonehenge on Arches' screen screen with something a bit more appropriate for your area. Open:

```
arches/Arches/Media/css
```

in a text editor.

You'll find css classes for most of the UI components used by Arches. You're free to replace Arches defaults with values for colors, fonts, weights, and images with your own favorites.

Most of the class names should be reasonably self explanatory. For example, find

```
.quick-search-content
```

this is the css class for Arches default page (e.g.: the "quick search" page). Note the background-image value defines a jpeg image in the /images sub-folder. Replace this value with an reference to the image you want to use instead of Stonehenge.

You can follow similar steps to change the colors and fonts that Arches uses by default. Once you've made your changes, go to the terminal window running Arches and press:

```
control-C
```

To stop the Arches server. Navigate to:

```
arches/build
```

and run the build script:

```
build.bat (./build.sh in Linux)
```

This script will update Arches with your new css settings. Once the script completes (usually in just a minute or two), navigate to the Arches root folder and start Arches:

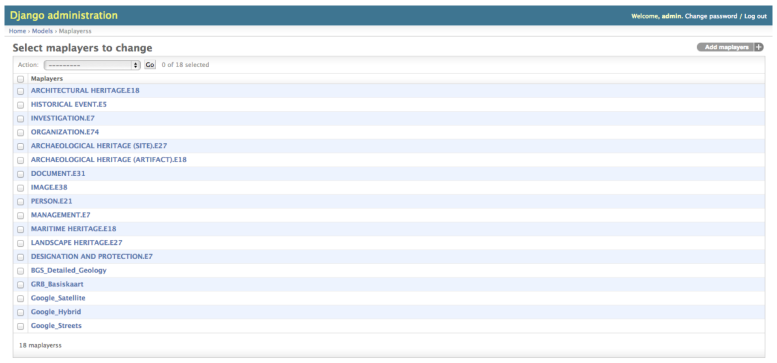
```
runserver.bat (./runserver.sh in Linux)
```

Log back on to Arches (you may need to clear your browser's cache) to view your changes.

11.7.2 Changing the Style of a Resource Layer

Arches allows you to define the map display for each resource. Although Arches provides reasonable default settings, you can override the color, transparency, line widths, and clustering display for any resource that you wish.

From the main Django Administration Console, click on **Maplayerss**. You'll see a screen similar to this:


Django administration
Welcome, admin. Change password / Log out
Home › Models › Maplayerss
Add maplayers
Action: ---------
Go

Select maplayers to change

0 of 18 selected

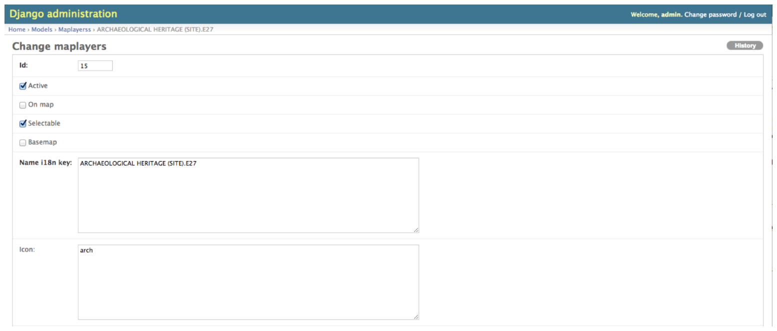
Maplayers
ARCHITECTURAL HERITAGE.E18
HISTORICAL EVENT.E5
INVESTIGATION.E7
ORGANIZATION.E74
ARCHAEOLOGICAL HERITAGE (SITE).E27
ARCHAEOLOGICAL HERITAGE (ARTIFACT).E18
DOCUMENT.E31
IMAGE.E38
PERSON.E21
MANAGEMENT.E7
MARITIME HERITAGE.E18
LANDSCAPE HERITAGE.E27
DESIGNATION AND PROTECTION.E7
BGS_Detailed_Geology
GRB_Basiskaart
Google_Satellite
Google_Hybrid
Google_Streets

18 maplayerss

Fig. 8: A listing of the resource map layers

Click on a resource, such as ARCHAEOLOGICAL HERITAGE (SITE).E27. You'll see a summary of information about how this resource is displayed as a map layer:

Scroll down to the "Layer" variable. You'll see something that starts out like:


Django administration
Welcome, admin. Change password / Log out
Home › Models › Maplayerss › ARCHAEOLOGICAL HERITAGE (SITE).E27
History

Change maplayers

Id: 15

[✓ Active]

On map

[✓ Selectable]

Basemap

Name i18n key: ARCHAEOLOGICAL HERITAGE (SITE).E27

Icon: arch

Fig. 9: Some of the display properties for the Archaeological Heritage(Site).E27 resource

```
var clusterStrategy = new OpenLayers.Strategy.Cluster({distance: 100, threshold: 3});
  var layer = new OpenLayers.Layer.Vector("ARCHAEOLOGICAL HERITAGE (SITE).E27", {
    rendererOptions: {zIndexing: true},
    styleMap: new OpenLayers.StyleMap({
        "default": new OpenLayers.Style({
             cursor: "pointer",
             strokeColor: "#2eb027",
             fillColor: "#2eb027",
             strokeOpacity: "${getStrokeOpacity}",
             fillOpacity: "${getFillOpacity}",
             pointRadius: "${getRadius}",
             strokeWidth: "${getStrokeWidth}",
             labelOutlineWidth: 1,
             fontColor: "#ffffff",
             fontOpacity: 1,
             fontSize: "12px",
             fontWeight:"bold",
             label: "${getLabel}"
    },{
  ...
```

OK, its a bit intimidating. But fear not, it will all make sense soon. This is a bit of JavaScript that Arches uses to define how the layer for each Resource Type is displayed on a map. It includes instructions on how to cluster resources of the same type when viewing data at regional scales, and it also tells Arches how to render each individual resource when the map is zoomed in.

Note: OpenLayers Mapping Library

Arches uses the OpenLayers (http://openlayers.org) library to manage the display of all mapping data. The JavaScript you see associated with each Resource Type is parsed by OpenLayers. Advanced System Administrators can reference the OpenLayers API and readily extend the default map display instructions.

Arches uses several techniques to display heritage resources on a map. They are:

- **Default Display:** These are the display parameters (colors, line weights, and fonts) that Arches uses to draw resources on the map.
- **Temporary Display:** These are the display parameters Arches uses when you place your cursor over a resource. This lets Arches highlight a specific resource when you move your cursor over the map.
- **Clusters:** Clusters are groupings of resources that are displayed at regional map scales (e.g: when you have the map zoomed out).

Changing the Default Display for a Resource You can control how individual resources are drawn on the map by updating this snippet of JavaScript:

```
"default": new OpenLayers.Style({
   cursor: "pointer",
   strokeColor: "#2eb027",
   fillColor: "#2eb027",
   strokeOpacity: "${getStrokeOpacity}",
   fillOpacity: "${getFillOpacity}",
   pointRadius: "${getRadius}",
   strokeWidth: "${getStrokeWidth}",
   labelOutlineWidth: 1,
   fontColor: "#ffffff",
   fontOpacity: 1,
   fontSize: "12px",
   fontWeight:"bold",
   label: "${getLabel}"
},
```

Each line in this bit of JavaScript tells Arches how to draw the geometry associated with an individual resource on a map.

For example, "strokeColor" defines the color used to draw lines (such as for linear features and the outlines of polygonal features). "fillColor" defines the color of the inside of polygons. "fillOpacity" and "strokeOpacity" define the transparency of the geometries associated with a resource.

So, you can change the color used to draw an ARCHAEOLOGICAL HERITAGE (SITE).E27 line geometry (for example) from green to white simply by replacing

```
strokeColor: "#2eb027",
```

with:

```
strokeColor: "#ffffff",
```

Notice that some of the drawing variables (such as "fillOpacity") have values starting with a "$". These variables inherit values that Arches sets for all resources. You're free to replace these values with your own if you want a particular resource to have its own specific drawing style.

When you've finished defining the drawing variables, click the "Save" button to save your edits. To update Arches with this new information, navigate to the arches/build folder and run:

```
build.bat (Windows) or ./build.sh (Linux)
```

Refresh your browser (you may have to clear your cache) and navigate to the Map screen in Arches to verify your new values.

Changing the Temporary (Highlight) Display for a Resource You can control how individual resources are highlighted on the map when a user moves the cursor over the resource on the Arches map. It works just like the default display, except that you update this snippet of JavaScript:

```
"temporary": new OpenLayers.Style({
   cursor: "pointer",
   strokeColor: "#2eb027",
   fillColor: "#2eb027",
   strokeOpacity: "${getStrokeOpacity}",
   fillOpacity: "${getFillOpacity}",
   pointRadius: "${getRadius}",
```

(continues on next page)

(continued from previous page)

```
    strokeWidth: "${getStrokeWidth}",
    labelOutlineWidth: 1,
    fontColor: "#ffffff",
    fontOpacity: 1,
    fontSize: "14px",
    fontWeight:"bold",
    label: "${getLabel}"
},
```

When you've finished defining the drawing variables, click the "Save" button to save your edits. To update Arches with this new information, navigate to the arches/build folder and run:

```
build.bat (Windows) or ./build.sh (Linux)
```

Refresh your browser (you may have to clear your cache) and navigate to the Map screen in Arches to verify your new values.

Clusters If you're new to mapping, "clusters" may not be a familiar concept. So, what are clusters? They are simply groups of geometries that represent the location of resources.

You'll notice that at regional mapping scales, Arches automatically clusters resource vectors. Why does Arches do this? Well, if your Arches dataset is large, Arches would have to draw each individual geometry for every resource. But at regional scales, with the map zoomed out, you wouldn't even be able to see the details of most of these geometries. Worse, your computer's browser would probably be overwhelmed by trying to draw thousands of small vectors.

So, Arches clusters resources to ensure that you can work with very large cultural heritage datasets.

Updating the way Arches renders clusters on the map is a bit of an advanced concept. It's really a function of the OpenLayers JavaScript library that Arches uses to map resources. Check out the OpenLayers API before you jump into this.

11.8 Limiting the Types of Resource to Load

Although most packages will provide a suite of Resource Type definitions, you are not required to use them all. Before you can start creating and managing cultural heritage data, you'll need to decide which Resource Types you want to use and then load the corresponding Resource Type definitions into Arches.

11.8.1 Select Resources to Load

Navigate to and open:

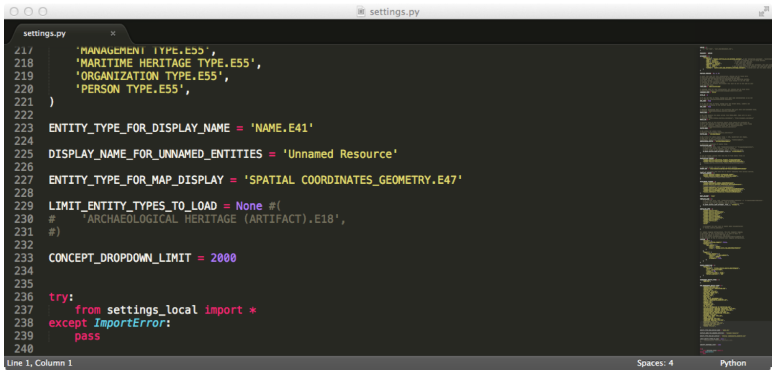
```
archesproject/settings.py
```

Scroll down until you find the "LIMIT_ENTITIES_TO_LOAD" parameter:

By default, Arches loads all the Resource Type definitions (hence the "limit to load" is None). You can define the specific definitions to load by providing a comma-delimited listing of just the resources you want to manage:

```
LIMIT_ENTITY_TYPES_TO_LOAD = ('ARCHAEOLOGICAL HERITAGE (ARTIFACT).E18', 'LANDSCAPE␣
↪HERITAGE.E27)
```

This would limit Arches to just loading the definitions for the Archaeological Heritage element) and Landscape Heritage types (from the CDS package in this case).



```
    'MANAGEMENT TYPE.E55',
    'MARITIME HERITAGE TYPE.E55',
    'ORGANIZATION TYPE.E55',
    'PERSON TYPE.E55',
)

ENTITY_TYPE_FOR_DISPLAY_NAME = 'NAME.E41'

DISPLAY_NAME_FOR_UNNAMED_ENTITIES = 'Unnamed Resource'

ENTITY_TYPE_FOR_MAP_DISPLAY = 'SPATIAL COORDINATES_GEOMETRY.E47'

LIMIT_ENTITY_TYPES_TO_LOAD = None #(
#    'ARCHAEOLOGICAL HERITAGE (ARTIFACT).E18',
#)

CONCEPT_DROPDOWN_LIMIT = 2000


try:
    from settings_local import *
except ImportError:
    pass

```

Fig. 10: Settings.py: Simple Search

CHAPTER 12

Next Steps: Using Arches

Now that you've installed and configured Arches, you can start building your inventory of cultural heritage. A few things that you'll probably want to consider:

12.1 Backup Your Arches Database

Arches uses the PostgreSQL/PostGIS relational database. You can download an application called PGAdmin (http://www.pgadmin.org/) to ease many of the basic Arches databasem management tasks.

PGAdmin has a clean and easy-to-understand user interface, and it will make database backups easy. But be forwarned! PGAdmin will also allow you to work with every aspect of the Arches data model, so tread carefully!

12.2 Performance Tuning

Arches doesn't attempt to tune any of the many performance variables that PostgreSQL exposes. But, based on the size and complexity of your dataset, you may want to re-allocate system resources to PostgreSQL.

People have written entire books on how to squeeze the best performance out of data management systems. Read-up on the basics before you attempt to optimize your system!

12.3 User Guide

Check out the Arches User Guide (http://archesproject.org/documentation/) for tips on how to use Arches for data creation and maintence, as well as how to effectively filter your cultural heritage inventory.

12.4 Feedback and Support

Check http://www.archeproject.org if you'd like to provide feedback on your experience with Arches, or if you're interested in getting support to customize or maintain Arches.

CHAPTER 13

Indices and tables

- genindex
- modindex
- search

CHAPTER 14

Modules

Modules are only viewable on your local fully built system.

Enable the following lines to view module documentation in your local system

```
.. automodule:: archesproject.arches.Models.entity
   :members:
   :undoc-members:
   :inherited-members:
```