

## Arches Implementation Considerations

*(updated September 10, 2019)*

The Arches Cultural Heritage Data Management Platform is robust enterprise-level software that heritage organizations can freely download, install, configure, and if desired customize and extend to meet their data management needs.

The purpose of this document is to help organizations to understand, for evaluation and planning purposes, what might be required to implement Arches according to their own requirements. The basic process for implementing Arches typically includes the following steps:

- Determine whether Arches is the right fit for your purposes, and whether you have access to the necessary technical expertise (in-house or externally) to implement Arches based on the steps below.
- Determine how you will host Arches (either on the cloud or your own servers).
- Install Arches' dependencies and the Arches software.
- Decide how you wish to model your data. You may use published Arches Resource Models, build your own, or perhaps utilize resource models from other members of the Arches community.
- Determine whether you need to modify or extend Arches based on your data modeling decisions and general requirements.
- If needed, prepare your existing/legacy data for import into Arches. This will include defining controlled vocabularies and cleaning and structuring your existing data.
- Create a strategy for ongoing support and maintenance of your Arches implementation and participation in the larger Arches community.

The remainder of this document expands on the above steps and is organized into five main sections:

- [Installation Considerations](#)
- [Data Considerations](#)
- [Configuration Considerations](#)
- [Customizing and Extending Arches](#)
- [Ongoing Support and Community Participation](#)

Each section describes the considerations and the recommended technical skills needed to be successful. In general, the more experience your technical support has with the following technologies, the easier the implementation process will be: Python, Django, Elasticsearch, Postgres/PostGIS, HTML/CSS, command line familiarity, Knockout, RequireJS, and Mapbox GL JS.

If your IT support doesn't have the skills and resources available to install, configure, customize, and/or maintain Arches, you may want to contract with a service provider experienced with implementing Arches. A listing of recognized Arches service providers is available on the Arches web site:

<https://www.archesproject.org/service-providers/> It is important to note that some service providers offer Arches technical training and knowledge transfer, which can be an option for organizations with IT staff.

## Installation Considerations

Arches will need to reside on a server, either on an in-house server or on a cloud hosting service, or perhaps even a combination of the two. If this is not the type of software installation that you or your team are familiar with, please consider the following points:

### ***Institutional hosting requirements and rules***

Begin by establishing if your institution has rules you'll need to follow when hosting databases and websites. In some cases, an in-house server is the only option, and in others, no such option will exist. This is important to consider up front, as it will impact the overall cost of the project.

### ***Which cloud hosting service will work best for Arches?***

If you are going to use a cloud hosting service there are many good options—Google, Microsoft and Amazon all offer cloud hosting services, as well as smaller companies like DigitalOcean. Any of these will work for Arches. However, at this point, the most extensive tests and deployments have been done using Amazon Web Services (AWS). We have found AWS to be very suitable in that it is flexible, well-documented, and cost-effective for most organizations.

### ***Technical specifications***

The [Arches documentation](#) recommends at least 4GB of RAM for evaluation and testing (8-16gb for production) and 2GB minimum disk space to install the code base. However, required disk space depends on the size and type of the data you'll be storing. Do you have a lot of photos or videos? These types of files will use much more disk space than simple database records, for example.

### ***Which operating system (OS) will work best for Arches?***

Arches works on Linux, Windows, and Mac servers. Linux (especially Ubuntu) is the most widely used operating system for Arches. A mapping library used by the tileserver included in Arches is not compatible with Windows (meaning you can't serve GeoTIFFs, for example, but otherwise everything else should work the same).

*Recommended skills/knowledge:* In order to install Arches on a server, the following skills and knowledge are necessary: command line familiarity, systems/server administration experience, Python, Django, Elasticsearch, and Postgres/PostGIS.

## Data Considerations

Arches was designed to be flexible enough to accommodate data of many different types and formats. As such, implementing organizations have the flexibility to decide what data they want to manage with Arches and how that data is organized and accessed. The following are important points to consider when planning how to configure Arches based on your data requirements:

### ***Legacy data***

Most organizations that are implementing Arches will have legacy (existing) datasets that will be migrated into Arches. Legacy data types that can be imported into Arches include spatial data, tabular data, and any kind of digital file (e.g., PDFs, images, videos, sound recordings). What's important to note is that inspecting legacy data can help to determine if previous patterns of data collection and usage are going to be continued using Arches, or if changes will be made to your organization's data creation and management methodology. In any event, the first step is to make a thorough review of the legacy data you wish to migrate, including the identification of data fields in each dataset.

### ***Data structure and organization***

By examining and determining what legacy data you want to import into Arches, it may become clear that new data fields and reorganization of the existing data are needed to serve your institutional goals. Remember, migrating to a new data management system is the best time to update and improve on old procedures! Once your organization has determined what data and processes you need to manage with Arches, you can start to dynamically define your Arches database and data entry forms using the [Arches Designer](#). Arches allows you to organize the data for each resource type (such as heritage resources, activities, historical events) in a Resource Model, which is a data model based on a graph structure. How you organize the data in a Resource Model determines the data entry forms and the contents of reports, as well as how the data can be searched. Also note that, by default, Arches uses the CIDOC Conceptual Reference Model (CRM), an ISO standard for cultural heritage information, as the semantic ontology for each Resource Model.

### ***Controlled vocabularies***

As part of your overall data strategy you may want to consider how to leverage the use of controlled vocabularies to enforce consistent use of the proper terminology during data entry, while also facilitating more accurate search results. Arches manages controlled vocabularies via its [Reference Data Manager](#) (RDM). In order to take full advantage of the RDM, which works in tandem with the Arches Designer, you may consider investing some time to create controlled vocabularies for data entry fields.

### ***Migrating data***

Arches facilitates the bulk import of legacy data through three data formats: csv, json, and shapefile. Which data format you use for the purpose of import into Arches depends on how complex and what kind of legacy data you have. If your data isn't particularly complex, then it may be easiest to convert your existing files to csv. However, if you have heavily nested and highly structured data, you may need to convert your data to json to prevent data loss. And if you are importing spatial data, you may either convert your spatial information to WKT (for import through csv) or use a shapefile to import the data into Arches. Each data field in an import file is mapped to the appropriate data field in an Arches Resource Model through an intermediary mapping file. For this reason, the Resource Models must be defined before any

data import can occur. As with any data migration process, reformatting and cleaning your data will likely be an involved undertaking, so be sure to plan accordingly.

*Recommended skills/knowledge:* The following skills will be helpful in determining your overall data strategy including data import and cleaning, and defining your Arches database: data processing and database administration, command line familiarity, knowledge of the CIDOC CRM if you choose to apply the CRM as an ontology, experience with constructing controlled vocabularies. Also, a thorough knowledge of your legacy data is required.

## **Configuration Considerations**

Once you've installed Arches on a server and determined your organization's overall requirements, you can begin to configure Arches. This document defines configuration as any activity implementers take to set up Arches according to their own needs without changing the core Arches software code. Customization of the Arches code to serve your specific use case is covered in the Customizing and Extending Arches section.

### ***Localized settings and content***

There are a number of settings and some content that will need to be localized. For example: home page content and branding, including the name you are giving the system and your organization's logo, saved searches, default map extent coordinates, additional map overlays, configuration of your basemaps including historic maps or satellite imagery. You may also be using a non-English language for your installation's UI—Arches uses Transifex to manage its ever-growing number of translations, so you can set your deployment to use one or more languages for which a translation is available. Changes to the settings or content can generally be made at any time during the configuration process, depending on how the data is organized.

### ***Configuring Arches to handle your data***

Arches allows you to load readymade packages, which include Resource Models, with predefined data fields, data entry forms and reports, and permissions settings, along with the accompanying controlled vocabularies. You may edit package settings using the [Arches Designer](#) and the [Reference Data Manager](#). Alternatively, you may choose to create new Resource Models and controlled vocabularies from scratch using these interfaces, which would require some expertise in database design, semantic modeling, and vocabulary creation. Packages, as well as their constituent Resource Models and controlled vocabularies, can be shared between Arches implementations that have similar requirements. As a result, the Arches community is currently assembling a library of packages and individual Resource Models that can be used as is or adapted for more specific use cases.

### ***Users and Permissions***

Arches allows you to manage individual users and user groups, and to define how both users and groups as well as the general public can interact with your data through permissions settings. For example, based on your organization's data access policies, you may decide that only certain staff members can edit geospatial data for specific types of resources or that your

Arches administrator is the only person who can modify your controlled vocabularies. You may add new users and groups and modify permissions at any time, but you may want to create an initial plan regarding who has access to view, create, edit, and delete what types of data. This information will help you to fully define your Resource Model permissions settings using the Arches Designer.

*Recommended skills/knowledge:* The latest version of Arches allows you to do many of your configuration tasks using the Arches user interface. These include: establishing and applying the name of your Arches instance; defining your Google Analytics key, changing basic search and map settings; adding saved searches; styling map layers (e.g., color of icons, transparency of overlays); creating and editing data fields, data entry forms, reports, and permissions settings using the Arches Designer; and defining and adding new terminology using the Reference Data Manager. In addition, using the Arches Django administration panel, you can also manage users and add the map layers to be managed within the user interface. For these tasks, no software coding is required, and apart from learning the Arches interface, you will need some background knowledge on the concepts underlying each task. For example, the Reference Data Manager allows you to easily create controlled vocabularies, but you will need some background knowledge on best practices to do correctly.

Configuration tasks that require additional IT skills include: changing the Arches front-end —i.e. how Arches appears to the end user, adding custom map layers, and changing or adding to the languages that the interface displays. In order to change the look and feel of Arches, knowledge of HTML/CSS and Javascript (Knockout, RequireJS) will be necessary. For adding custom map layers such as creating and adding a historic map overlay, the skills involved include geospatial data analysis, PostGIS, georeferencing, and command line familiarity.

### **Customizing and Extending Arches**

The Arches platform source code is open, malleable and extensible, so your imagination is the limit of how your installation can be customized to support your needs. Here are some examples of customizing and extending Arches:

- Creating a unique user interface;
- Setting up e-mail reporting so admins get a daily summary of database activity;
- Integrating Arches with another system so that resources are synchronized between the two databases;
- Setting up the login page as the first thing that a user encounters;
- Incorporating 3D models or other types of viewers into resource reports;
- Implementing a draft → publish workflow for resource creation;
- Adding the ability to geocode addresses to create spatial coordinates.

Enhancements such as these can be seamlessly integrated into your Arches implementation. We only ask that you share your improvements with the rest of the Arches community if they are more broadly applicable! See the next section for more information on community participation.

*Skills required:* For most Arches enhancements it will be necessary to have a good working knowledge of the following: HTML/CSS and Javascript (Knockout, RequireJS), Python, Django, PostgreSQL/PostGIS, Mapbox GL JS, and Elasticsearch.

### **Ongoing Support and Community Participation**

As you are installing, configuring, and potentially customizing your Arches instance, you may come to the realization that the Arches platform is meant to support your ongoing work and the data that it produces. The following are some considerations to help you plan for the future of your Arches implementation:

#### ***Ongoing maintenance and server administration***

It will be necessary to establish a system administrator, potentially train that administrator as needed, and determine who will provide ongoing technical support. Ongoing support will consist of basic server updates, as well as Arches upgrades and enhancements.

#### ***Data updates***

In order to ensure that the data in your Arches instance is valid and authoritative, it is essential that a strategy for ongoing data updates is established. If new data will be entered using the Arches data entry interface or through Arches Collector, then new or existing staff must be trained in their use and training materials may need to be created. If data will be updated through a linkage with another system, then you may need to plan for a customization that enables that.

#### ***Community Participation***

The Arches open source community has many opportunities for involvement by Arches implementers, and these include, but are not limited to: sharing Arches Resource Models and packages; contributing new code as the result of customizing and extending Arches; helping other Arches implementers via the Discussion Forum and other channels; writing articles about your Arches experience or helping to improve Arches documentation; and taking part in the overall governance of the community and helping to determine the general developmental direction of the software. Being part of the Arches community ensures that you are up-to-date on the latest Arches news and developments, which help you to best maintain your Arches instance.

*Recommended skills/knowledge:* For ongoing maintenance, the recommended skills are the same as those under the Installation Considerations. For data updates, the skills are the same as those under Data Considerations. For community participation, please see the [Arches Community Code of Conduct](#) for some guidelines on general community expectations.

If you have questions or feedback regarding implementation considerations, please post on the Arches Discussion Forum (<http://www.archesproject.org/forum>)