D.2.4.P2 SIP Creation and Ingest of Records (Full Scale Pilot 2)

DOI: 10.5281/zenodo.1171451

Grant Agreement Number:	620998
Project Title:	European Archival Records and Knowledge Preservation
Release Date:	12th February 2018
Contributors	
Name	Affiliation
Terje Pettersen-Dahl	National Archives of Norway
Arne-Kristian Groven	National Archives of Norway
István Alföldi	National Archives of Hungary
David Anderson	University of Brighton
Janet Anderson	University of Brighton

Table of Contents

1. Executive Summary

This document is part of the deliverable:

D2.4) Pilot documentation

Pilot documentation: This package of documentation will provide technical and end-user guidance to support not only the pilot sites but also possible future deployments thereafter. [month 33] (from DoW)

Structure of this deliverable

The deliverable is a package if linked documents.

This **Summary** contains the common information and short overview of the pilots, along with links to the final version of the Pilot Definition excel files and Pilot Documentation Packages. The **Pilot Definition** excel provides detailed information about scenarios, data sets and step-by-step preparation and process step instructions. The **Pilot Documentation Package** is created by the pilot staff responsible for the pilot execution. This package contains additional information along with screenshots (and videos in some cases) of the tools during the execution of the pilot.

Summary (this document) – Created by WP2

Pilot Package – Pilot 1
- Pilot Definition (Final version) – Created by WP2 and Pilot 1 responsible
- Pilot Documentation files – Created by Pilot 1

Pilot Package – Pilot 2
- Pilot Definition (Final version) – Created by WP2 and Pilot 2 responsible
- Pilot Documentation files – Created by Pilot 2

Pilot Package – Pilot 3
- Pilot Definition (Final version) – Created by WP2 and Pilot 3 responsible
- Pilot Documentation files – Created by Pilot 3

Pilot Package – Pilot 4
- Pilot Definition (Final version) – Created by WP2 and Pilot 4 responsible
- Pilot Documentation files – Created by Pilot 4

Pilot Package – Pilot 5
- Pilot Definition (Final version) – Created by WP2 and Pilot 5 responsible
- Pilot Documentation files – Created by Pilot 5

Pilot Package – Pilot 6
- Pilot Definition (Final version) – Created by WP2 and Pilot 1 responsible
- Pilot Documentation files – Created by Pilot 6

Pilot Package – Pilot 7
- Pilot Definition (Final version) – Created by WP2 and Pilot 7 responsible
- Pilot Documentation files – Created by Pilot 7

2. Pilot documentation

2.1 Scenarios

Scenario 1: SIP Creation and Ingest of unstructured records

Scenario 2: SIP Creation and Ingest of unstructured records

Scenario 3: SIP Creation and Ingest of structured records

2.2 About the pilot

Pilot goals

The initial goals have been:

1. To evaluate tools developed in the E-ARK project on real data sets, containing a substantial number of digital records
2. To specifically look into the ability to get a job done with a satisfactory result
3. Also to provide some quantitative measurements around time consumption
4. Preferably using the E-ARK package definitions

Deviations

When the pilot goals for our pilot were defined at the beginning of the project, the direction of the E-ARK work regarding OAIS package structure implementations was still not known to us.

We hoped it would imply minor changes compared to OAIS package structure we had defined at the National Archives of Norway a few years earlier. But the resulting E-ARK package structure represented more substantial differences than anticipated.

As it turned out there were two options available for us:

a) Either to apply the E-ARK package definitions on constructed data sets on test servers with low security level, or
b) Evaluate the E-ARK tools on the real world data set, but with another OAIS package structure

How we addressed the deviations

Our chosen solution to this challenge is:

- We run the three original scenarios on real world data sets, using tools developed in E-ARK, but not on the E-ARK package structure. (From which we are reporting here.)
- In addition we run (in November 2016) an additional test scenario on constructed data sets in a test environment, using the E-ARK package[1] structure.

By doing it this way we can also evaluate the tools ability to handle different OAIS package structures.

Summary of our pilot experience

[1] http://www.eark-project.com/resources/project-deliverables/51-d33pilotspec/file

To summarize our experience:

- The tools we evaluated were able to produce satisfactory results (output).
- But we would like to evaluate on even larger data sets to conclude about scalability

The time consumption measurements can be found later in this report. The subjective experience from (some of) the personnel performing the pilots is that they would like the automated actions to take less time. But if that is realizable has to be seen? One should also compare with time consumption in comparable tools, which is beyond the scope of this pilot.

If the tools (easily) can be reconfigured regarding OAIS package structure, as we assume but yet has to experience ourselves, it would be a nice trait of such tools.

The evaluated tools have implemented the METS and PREMIS standards as XML structures, which is a good thing. The ability to export/import (even) more standards like EAD or DC would also be positive.

2.2.1 Organisations involved

The organizations involved in this pilot are: The National Archives of Norway and ES Solutions AB (Sweden, also partner in the project).

The data used are real production data, but the data providers have not been directly involved in the pilot.

2.2.2 Software components

Software components developed as part of the E-ARK project

The main software products involved in this pilot are:

- ETA (ESSArch Tools Archive), v0.93.1
- EPP (ESSArch Preservation Platform), v2.7.3

ETA is the depots front-end to EPP, the preservation platform.

ETP (ESSArch Tools Producer) is a front-end tool meant for the producer side. We originally planned to simulate producer activities using the newest version of ETP (v0.93.1). But it was not possible, due to the fact that we only had Windows platforms available when the pilots were run and ETP is only available on Linux. Instead a preliminary version of ETP called ET was used[2].

The software is open source software, available from:

- http://epp.essarch.org/ (EPP)
- http://eta.essarch.org/ (ETA)
- http://etp.essarch.org/ (ETP)

[2] In our additional test scenario we will use the latest version of ETP.

In the text below the term ESSArch is used instead of mentioning all three components, ETP, ETA, and EPP. specifially.

ESSArch is an Open Source solution and consist of several modules and engines. The system has an extended policy and I/O engine besides a quality check and validation function. It is storage independent and continuously tracks (all) events related to the digital depot.

The tools implements of OAIS (Reference Model for an Open Archival Information System) and includes PreIngest, Ingest, Data Management, Archival Storage.

Access and PreAccess. ESSArch is managed by Archive Management and Management Regulations. But ESSArch is not a tool for presenting material for access and search.

The tools are using METS (Metadata Encoding and Transmission Standard) descriptors which can be included in the ETP or ETA tools. ESSArch uses PREMIS (Preservation Metadata and Maintenance Activity) together with METS to describe events and technical metadata for the content.

ESSArch uses HTTP (HyperText Transfer Protocol) as one interface to communicate with thru a traditionell web reader. XMLRPC (XML Remote Procedure Call) as another interface to communicate with ESSArch. W3C SOAP (Simple Object Access Protocol) as third way.

ESSArch is built upon MySQL databases for managing information packages. ESSArch can natively write to a Unix filesystem or a Tape library.

Locally developed software

In all three scenarios locally developed software has been used related to either construction of content or validation of content. They are mentioned here for completeness:

- Arkadukt 2.0: Create XML files validating with respect to ADDML (locally developed), from CSV files
- ArkN4: Validates content with respect to the Noark-4 standard.
- Arkade 4.0: Validates content with respect to the Noark-5 standard or the ADDML standard.

2.3 Data characterization

Scenario 1

Data used in this pilot originated from an archival producer called Direktoratet for utviklingssamarbeid (Norad). Norad is the Norwegian Agency for Development Cooperation. The period of data creation was 2003-04-01 to 2005-12-31.

The data is the standardized output from an EDRMS system, generated according to the national NOARK-4 standard. The output consisting of a generated Noark-4 XML file containing (meta-) data descriptions of the records created in the EDRMS, in addition to the documents (records) represented as (mostly) PDF/A files. The overall size is 20 GB.

Scenario 2

Data used in this pilot originated from an archival producer called Folketrygdfondet, which is a professional investment manager whose main task is to manage the Government Pension Fund Norway on behalf of the Ministry of Finance. The period of data creation was 2005-01-01 to 2014-03-31.

The data is the standardized output from an EDRMS system, generated automatically according to the national NOARK-5 standard. The output consisting of a generated Noark-5 XML file containing (meta-) data descriptions of the records created in the EDRMS, in addition to the 8194 documents (records) represented as (mostly) PDF/A files. The overall size is 5 GB.

Scenario 3

Data used in this pilot originated from an archival producer called Direktoratet for Naturforvaltning, a no longer existing directorate related to environmental administration. The period of data creation was 1985-01-01 to 1999-12-30.

The data set is a register, originating from an old database made in the 1980s. The data is represented as a set of 10 CSV files, containing 338.500 registrations. The size is 105 MB.

2.4 Pilot workflow

The software components described in the previous section were orchestrated according the the following sequence:

1. Construction of ADDML files (not involving E-ARK tools)
2. Validate content (not involving E-ARK tools)
3. Metadata preparation, create/submit SIP
4. Receive/validate SIP

Each of these stages are described on the following sections:

Data extraction has not been part of the different scenarios. But data has been taken from the line of production at the National Archives. (See data characterization above.)

2.4.1 Construction of ADDML files (Scenario 3)

For scenario 3 an XML file in accordance with our (locally developed) ADDML standard was first created in the (locally developed) Arkadukt 2.0, based on the CSV output from the old database source, and (other) textual descriptions.

The first few lines of the ADDML file looks as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<addml name="Jegerregisteret" xmlns="http://www.arkivverket.no/standarder/addml">
  <dataset name="Jegerregisteret 1998">
    <reference/>
    <flatFiles>
      <flatFile name="jeger" definitionReference="ffd_3">
        <properties>
          <property name="fileName">
            <value>jeger.dat</value>
          </property>
        </properties>
      </flatFile>
      <flatFile name="ut_jeger" definitionReference="ffd_4">
        <properties>
          <property name="fileName">
            <value>ut_jeger.dat</value>
          </property>
        </properties>
      </flatFile>
      <flatFile name="ikkejeg" definitionReference="ffd_5">
        <properties>
          <property name="fileName">
            <value>ikkejeg.dat</value>
          </property>
        </properties>
      </flatFile>
      <flatFile name="jegerkort" definitionReference="ffd_6">
        <properties>
          <property name="fileName">
            <value>jegerK.dat</value>
          </property>
        </properties>
```

2.4.2 Validate content

Content was then validated, also using locally developed software tools:

- ArkN4 for Noark 4 content
- Arkade 4.0 for Noark 5 and ADDML content.

2.4.3 Metadata preparation, create/submit SIP

The initial plan was to use the newest version of the ETP, but a preliminary version called ET was used instead.

Creates log information as PREMIS objects and fixity information using SHA-256.

Uses METS schema as a standard for encoding descriptive, administrative, and structural metadata regarding the objects involved

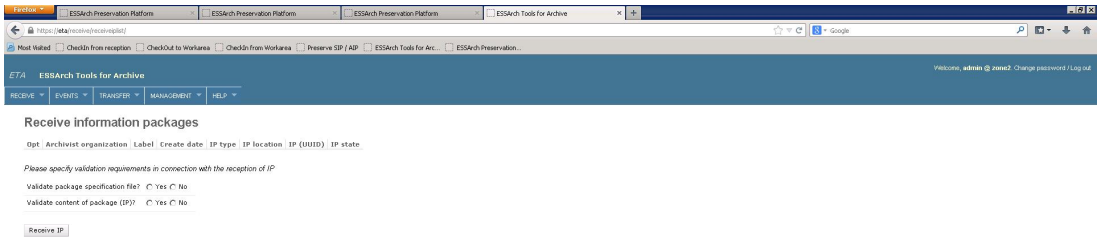
Create packages according to SIP structure.

The time used on the automated parts of these operations, are as follows:

- Scenario 1 (20 GB)
    - 1 h 50 min
- Scenario 2 (5 GB)
    - 1 h
- Scenario 3 (105 MB)
    - (?)

2.4.4 Receive/validate SIP

The E-ARK tool used in this step was ETA (ESSArch Tools Archive), v0.93.1:



The functionality includes

- Receiving SIP
- Validation of submission, e.g.:
    - Integrity/fixity validation
    - Syntax check, of the IP
- Review and approval of content

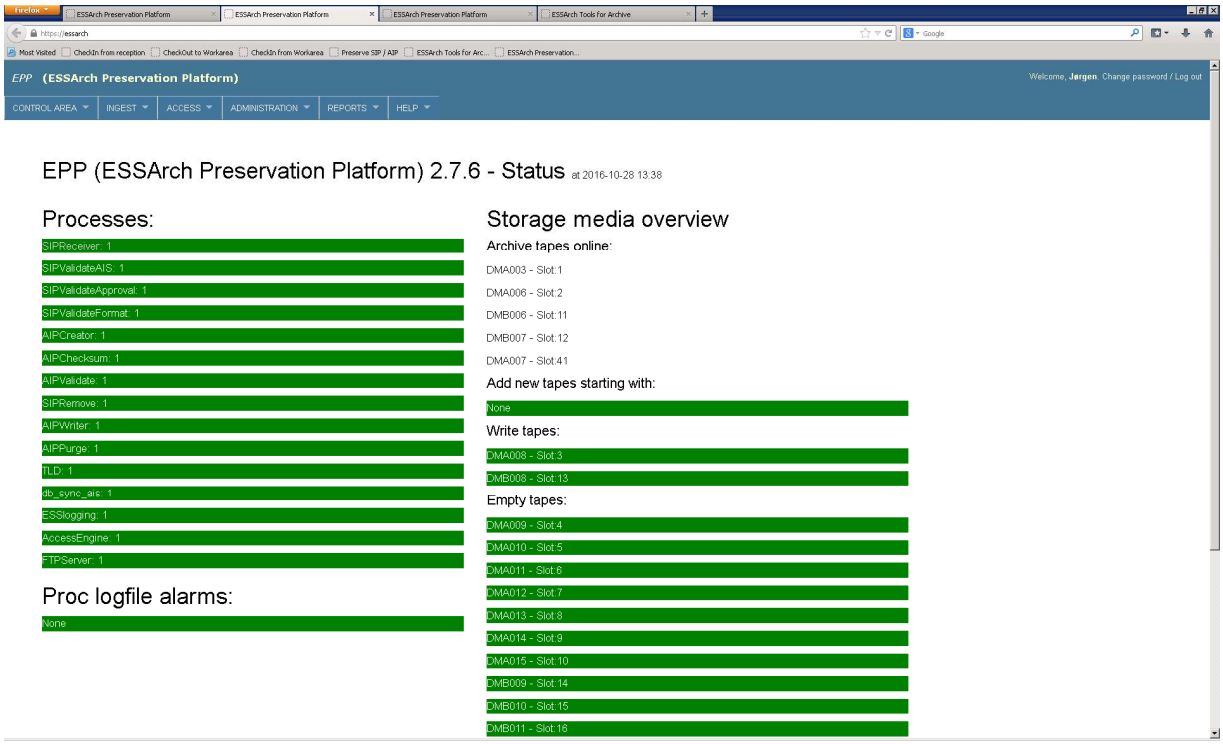
The automated part used 50 min on the largest data set.

2.4.5 AIP creation and archival storage

The E-ARK tool used in this step was EPP (ESSArch Preservation Platform), v2.7.3.

An AIP is generated and stored together with the original SIP. In addition, some validation activities take place in EPP and PREMIS files are updated.

The picture below shows the ETA user interface (version 2.7.6 upgrade):



.

2.5 Infrastructure

The infrastructure used in this pilot was composed of clients and servers:

Characteristic	Client side	Server side
Type	Physical	Physical
CPU	2 cores, 4 threads @ 3.10GHz	16 threads @ 2,4 GHz
RAM	20 GB	72 GB
Storage	128 GB systemdisk, 2 TB storage disk	Local 1,7 TB, Iscsi 3 x 8,0 TB, IBM TS 3500 - LTO5
Software	Win 7 Pro x64	Centos 6.4 ESSArch ETP ESSArch EPP MySql …

2.6 Installation instructions

2.6.1 EPP (ESSArch Preservation Platform)

Installation instructions for EPP (ESSArch Preservation Platform) are available at the following web site: http://epp.essarch.org/. Here are the main points, just to illustrate:

Environment preparation

Create user and group

Don't forget to create /home/arch

```
Please run the following command as user root.
# groupadd arch
# useradd -c "ESSArch System Account" -m -g arch arch
```

Set password for arch user

```
Please run the following command as user root.
# passwd arch
Changing password for user arch.
New UNIX password: password
Retype new UNIX password: password
```

Customize user environment for arch user

Add the following rows to /home/arch/.bash_profile:

```
### ESSArch start
##
export PATH=/ESSArch/pd/python/bin:$PATH:/usr/sbin
export LANG=en_US.UTF-8
export LD_LIBRARY_PATH=/ESSArch/pd/python/lib:/ESSArch/pd/libxslt/lib:/ESSAr
export EPP=/ESSArch/pd/python/lib/python2.7/site-packages/ESSArch_EPP
export PYTHONPATH=$EPP:$EPP/workers:/ESSArch/config
export DJANGO_SETTINGS_MODULE=config.settings
alias bin='cd /ESSArch/bin'
alias log='cd /ESSArch/log'
##
### ESSArch end
```

Create installation directory

```
Please run the following command as user root.
# mkdir /ESSArch
# chown -R arch:arch /ESSArch
```

Installation

EPP installation script

```
Change to user "arch" with the following command.
# su - arch

Download and extract ESSArch_EPP_install tarfile.
[arch@server ~]$ wget http://xxx.xxx.xxx.xxx/ESSArch_EPP_installer_xxxxxxxxx
[arch@server ~]$ tar xvf ESSArch_EPP_installer_xxxxxxxxxxxx.tar.gz
[arch@server ~]$ cd ESSArch_EPP_installer
[arch@server ~]$ ./install

The installation of ESSArch is now running and dependent on hardware configu
[arch@server ~]$ tail -f /ESSArch/install.log

When installation is finished, search in the log file /ESSArch/install.log f
```

Installation of Advanced Message Queuing Protocol

ESSArch is designed to be AMQP (Advanced Message Queuing Protocol) independent. However the installation package is prepared for RabbitMQ and the following instructions assume that you use RabbitMQ.

Follow the instructions below in order to install RabbitMQ required by ESSArch.

```
Please run the following commands as root user.
# rpm -i /ESSArch/install/packages/rabbitmq-server.rpm
# chkconfig rabbitmq-server on
# service rabbitmq-server start
```

If startup failed and you see an error message in /var/log/rabbitmq/startup_log after a minute or so like:

ERROR: epmd error for host "yourhostname": timeout (timed out)_

Then you need to update your /etc/hosts file to add your hostname to the list of localhost:

127.0.0.1 yourhostname

Configuration

Apache httpd configuration

```
Edit file /ESSArch/config/httpd-epp.conf and change the configuration entry

For test purpose you can use the existing configuration for SSL certificate,
```

ESSArch configuration

```
In /ESSArch/config you will find all the configuration files for ESSArch. Th

For ESSArch you will find the configuration in the local database tables. To
```

Database

ESSArch is designed to be RDBMS-independent. However the installation package is prepared for MySQL and the following instructions assume that you use MySQL.

MySQL

Please note that the MySQL JDBC driver provided by the installer requires MySQL version 5.x or higher.

Follow the instructions below in order to create the user and tables required by EPP installation.

```
To enable MySQL on CentOS 5.5 use the following commands.
/sbin/chkconfig mysqld on
/sbin/service mysqld start
/usr/bin/mysql_secure_installation

The MySQL commands listed below can be run within the mysql program, which m
# mysql -u root -p

Create the database. For example, to create a database named "essarch", ente
mysql> CREATE DATABASE essarch DEFAULT CHARACTER SET utf8;

Set username, password and permissions for the database. For example, to set
mysql> GRANT ALL ON essarch.* TO arkiv@localhost IDENTIFIED BY 'password';
```

Create default tables in database

```
Please run the following command as user: arch
[arch@server ~]$ python $EPP/manage.py migrate
```

Add default configuration data to database

Use only this default configuration for test purpose, do not install this default configuration in production.

For production environment you should first make a copy of this configuration file and update for example site_profile, site_name. After you done all your updates you install it.

```
Please run the following command as user: arch
[arch@server ~]$ python $EPP/extra/install_config.py
```

2.6.2 ETA (ESSArch Tools Archive)

Installation instructions for ETA (ESSArch Tools Archive) are available at the following web site: http://eta.essarch.org/. Here are the main points, just to illustrate:

Environment preparation

Create user and group

Don't forget to create /home/arch

```
Please run the following command as user root.
# groupadd arch
# useradd -c "ESSArch System Account" -m -g arch arch
```

Set password for arch user

```
Please run the following command as user root.
# passwd arch
Changing password for user arch.
New UNIX password: password
Retype new UNIX password: password
```

Customize user environment for arch user

Add the following rows to /home/arch/.bash_profile:

```
### ESSArch start
##
export PATH=/ESSArch/pd/python/bin:$PATH:/usr/sbin
export LANG=en_US.UTF-8
export LD_LIBRARY_PATH=/ESSArch/pd/python/lib:/ESSArch/pd/libxslt/lib:/ESSAr
export ETA=/ESSArch/pd/python/lib/python2.7/site-packages/ESSArch_TA
export PYTHONPATH=$ETA:/ESSArch/config
export DJANGO_SETTINGS_MODULE=config.settings
alias bin='cd /ESSArch/bin'
alias log='cd /ESSArch/log'
##
### ESSArch end
```

Create installation directory

```
Please run the following command as user root.
# mkdir /ESSArch
# chown -R arch:arch /ESSArch
```

Installation

ETA installation script

```
Change to user "arch" with the following command.
# su - arch

Download and extract ESSArch_TA_install tarfile.
[arch@server ~]$ wget http://xxx.xxx.xxx.xxx/ESSArch_TA_installer_xxxxxxxxxx
[arch@server ~]$ tar xvf ESSArch_TA_installer_xxxxxxxxxxx.tar.gz
[arch@server ~]$ cd ESSArch_TA_installer
[arch@server ~]$ ./install

The installation of ESSArch is now running and dependent on hardware configu
[arch@server ~]$ tail -f /ESSArch/install.log

When installation is finished, search in the log file /ESSArch/install.log fo
```

Installation of Advanced Message Queuing Protocol

ESSArch is designed to be AMQP (Advanced Message Queuing Protocol) independent. However the installation package is prepared for RabbitMQ and the following instructions assume that you use RabbitMQ.

Follow the instructions below in order to install RabbitMQ required by ESSArch.

```
Please run the following commands as root user.
# rpm -i /ESSArch/install/packages/rabbitmq-server.rpm
# chkconfig rabbitmq-server on
# service rabbitmq-server start
```

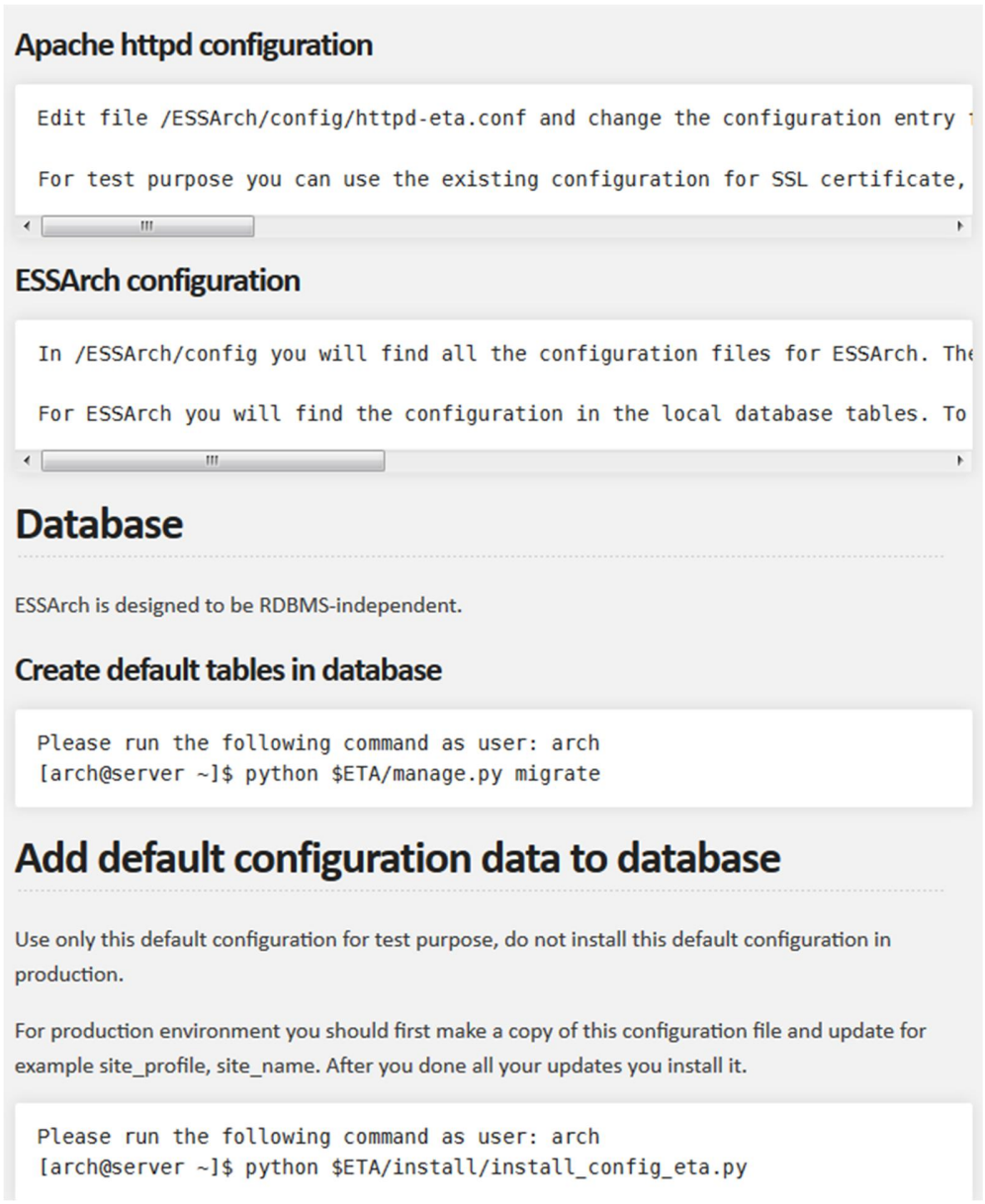
If startup failed and you see an error message in /var/log/rabbitmq/startup_log after a minute or so like:

ERROR: epmd error for host "yourhostname": timeout (timed out)_

Then you need to update your /etc/hosts file to add your hostname to the list of localhost:

127.0.0.1 yourhostname

Configuration



2.6.3 ETP (ESSArch Tools Producer)

Installation instructions for ETP (ESSArch Tools Producer) are available at the following web site: http://etp.essarch.org/. Here are the main points, just to illustrate:

Environment preparation

Create user and group

Don't forget to create /home/arch

```
Please run the following command as user root.
# groupadd arch
# useradd -c "ESSArch System Account" -m -g arch arch
```

Set password for arch user

```
Please run the following command as user root.
# passwd arch
Changing password for user arch.
New UNIX password: password
Retype new UNIX password: password
```

Customize user environment for arch user

Add the following rows to /home/arch/.bash_profile:

```
### ESSArch start
##
export PATH=/ESSArch/pd/python/bin:$PATH:/usr/sbin
export LANG=en_US.UTF-8
export LD_LIBRARY_PATH=/ESSArch/pd/python/lib:/ESSArch/pd/libxslt/lib:/ESSAr
export ETP=/ESSArch/pd/python/lib/python2.7/site-packages/ESSArch_TP
export PYTHONPATH=$ETP:/ESSArch/config
export DJANGO_SETTINGS_MODULE=config.settings
alias bin='cd /ESSArch/bin'
alias log='cd /ESSArch/log'
##
### ESSArch end
```

Create installation directory

```
Please run the following command as user root.
# mkdir /ESSArch
# chown -R arch:arch /ESSArch
```

Installation

ETP installation script

```
Change to user "arch" with the following command.
# su  arch

Download and extract ESSArch_TP_install tarfile.
[arch@server ~]$ wget http://xxx.xxx.xxx.xxx/ESSArch_TP_installer_xxxxxxxxxx
[arch@server ~]$ tar xvf ESSArch_TP_installer_xxxxxxxxxxxx.tar.gz
[arch@server ~]$ cd ESSArch_TP_installer
[arch@server ~]$ ./install

The installation of ESSArch is now running and dependent on hardware configu
[arch@server ~]$ tail f /ESSArch/install.log

When installation is finished, search in the log file /ESSArch/install.log f
```

Installation of Advanced Message Queuing Protocol

ESSArch is designed to be AMQP (Advanced Message Queuing Protocol) independent. However the installation package is prepared for RabbitMQ and the following instructions assume that you use RabbitMQ.

Follow the instructions below in order to install RabbitMQ required by ESSArch.

```
Please run the following commands as root user.
# rpm i /ESSArch/install/packages/rabbitmq-server.rpm
# chkconfig rabbitmq-server on
# service rabbitmq-server start
```

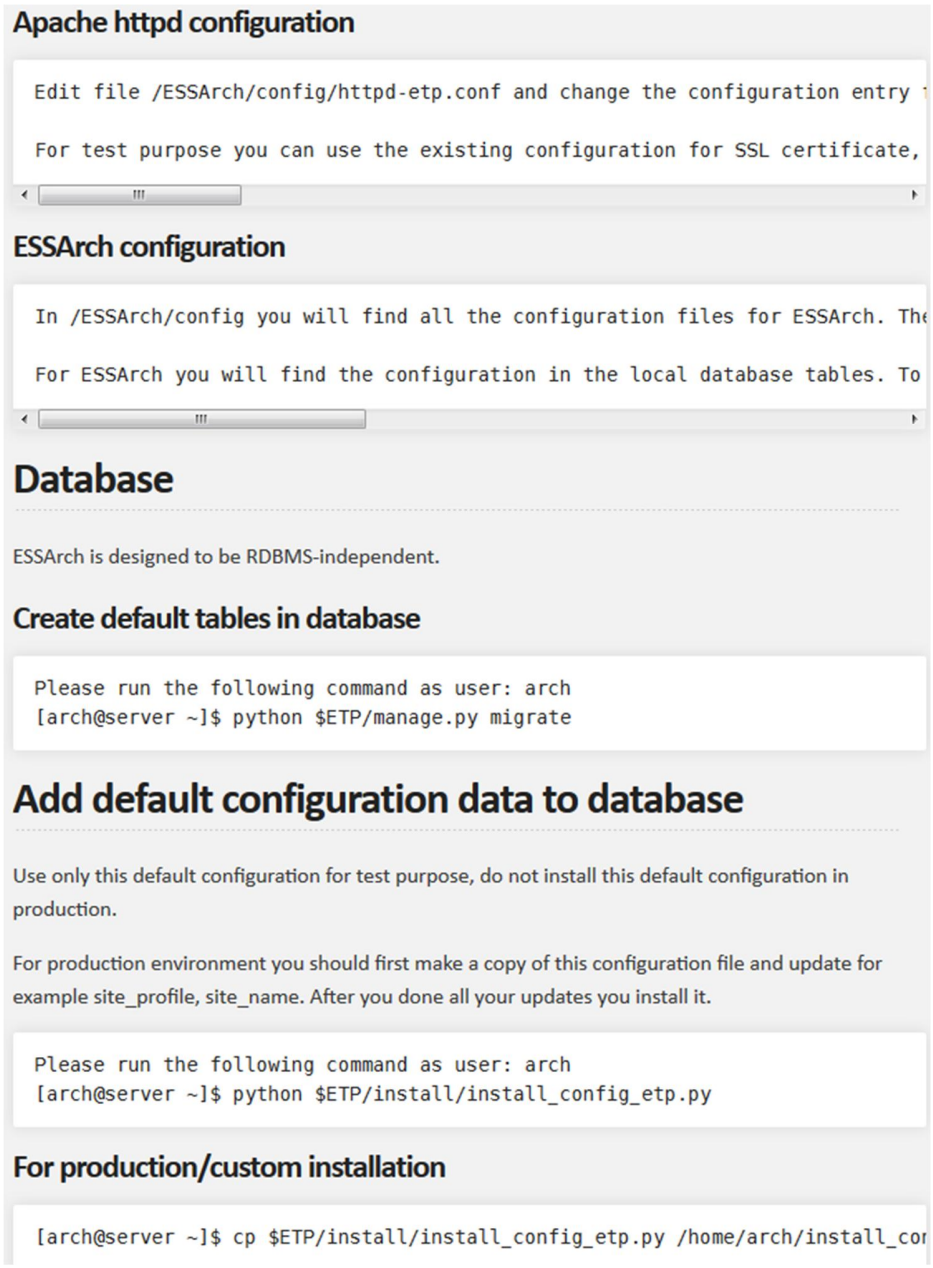
If startup failed and you see an error message in /var/log/rabbitmq/startup_log after a minute or so like:

ERROR: epmd error for host "yourhostname": timeout (timed out)_

Then you need to update your /etc/hosts file to add your hostname to the list of localhost:

127.0.0.1 yourhostname

Configuration



2.7 Other support sources

Service and support on ETP is regulated in maintenance contract with ES Solutions AB. A case is registered on the support portal http://projects.essolutions.se.