*version 2.8*Table of Contents

1 Introduction 3

1.1 Overview of EASY 3

1.2 Installation packages 4

1.3 Passwords 4

1.4 Conventions 5

2 Standard Software Components 6

2.1 Redhat 6 or CentOS 6 6

2.2 Oracle Java SE 7 SDK (CentOS) 6

2.3 Oracle Java SE 7 SDK (RedHat) 7

2.4 Tomcat 6 7

2.5 Apache HTTP Server 2.2.15 8

2.6 PostGreSQL 8.4 8

2.7 OpenLDAP 2.4 9

3 EASY Back-end Modules 11

3.1 EASY Fedora Commons Repository 11

3.2 EASY LDAP Directory 17

3.3 EASY Filesystem RDB 20

3.4 EASY PID RDB 21

3.5 EASY SOLR Search Index 22

4 EASY Frond-end Modules 25

4.1 EASY Web-UI Application 25

4.2 EASY SWORD-based Ingest Module (Optional) 26

4.3 EASY Experimental RESTful Module (Optional) 28

4.4 EASY Customized PrOAI Module (Optional) 29

5 EASY Tools For Data Managers 31

5.1 EASY Batch Ingest & Update (EBIU) 31

# Introduction

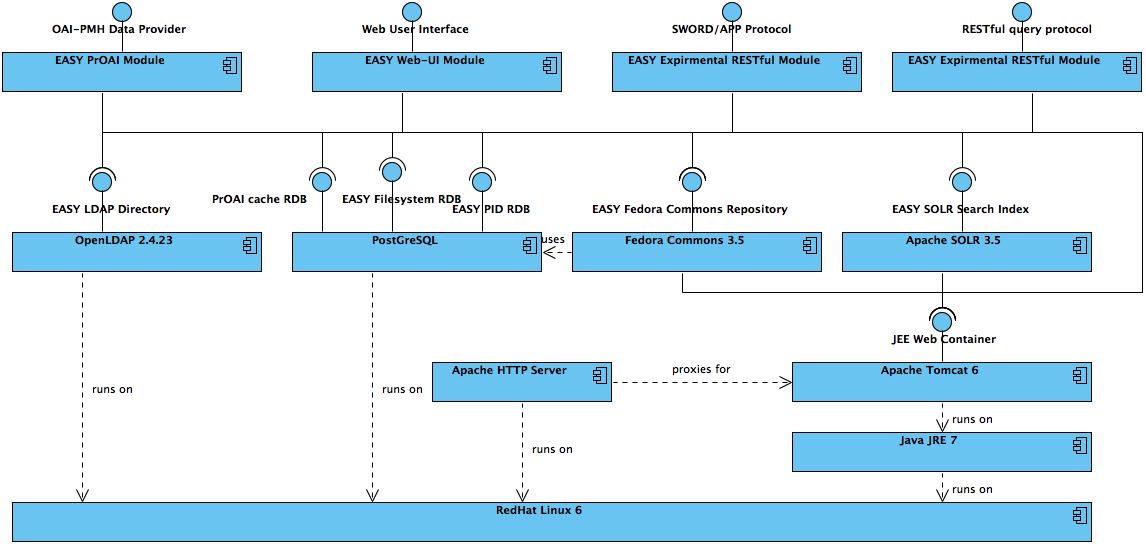
The Electronic Archiving SYstem (EASY) is the DANS[[1]](#footnote-1) Fedora Commons based repository system for the long term preservation of scientific research data. It includes a web-based user interface that lets users find and download data as well as submit information packages for ingest into the repository.

This document will guide you through the steps of installing EASY on a server. As described below, EASY is built on several open source software components. Several configurations on different platforms should therefore be possible. However, this Guide describes a simple one-server set-up, on a CentOS 6.3 or a RedHat Linux 6 server, the configuration currently in use at DANS. So far, no other configurations have been tested.

## Overview of EASY

Refer to the diagram below for an overview of the components that make up an EASY installation.

Figure EASY Components



Some of these components could *in principle* be replaced by different components. If only a standard protocol is mentioned in the interface, a different implementation of that protocol could possibly be used:

* EASY LDAP Directory - another LDAP implementation could be used;
* EASY Filesystem RDB - another rdbms could be used;
* EASY PID RDB - another rdbms could be used;
* EASY Fedora Commons Repository - needs to be a version of Fedora Commons;
* EASY SOLR Search Index - needs to be a version of Apache SOLR.

*However, it is important to remember that only the configuration discussed in this document has been tested.*

## Installation packages

Before you continue, please make sure you have the following required installations packages:

* *easy-backend-2.8.tar.gz* - contains the files for setting up the back-end services needed to run EASY, as well as a copy of this Guide;
* *easy-webui-2.8.tar.gz* - contains the Web-UI Application;
* *easy-sword-2.8.tar.gz* (optional) - contains the SWORD-based Ingest Module;
* *easy-rest-2.8.tar.gz* (optional) - contains the Experimental RESTful Module;
* *easy-proai-1.1.tar.gz* (optional) - contains the Customized PrOAI Module;

When referring to files in these packages we will use the following conventions:

$EASY\_BACKEND, $EASY\_WEBUI, $EASY\_SWORD, $EASY\_REST and $EASY\_PROAI refer to the directory’s created by unzipping *easy-backend-2.8.tar.gz, easy-webui-2.8.tar.gz*, *easy-sword-2.8.tar.gz, easy-rest-2.8.tar.gz* and *easy-proai-1.1.tar.gz* respectively. So, to look for the file $EASY\_BACKEND/util/java.sh, you should unzip the file *easy-backend-2.8.tar.gz*, open the resulting directory, look for a subdirectory called “util” and look there for the file “java.sh.”

## Passwords

During the installation you will be asked several times to provide a password. Please, ensure that you create safe passwords. Prefer randomly generated passwords over human readable ones. Store your passwords in a central, encrypted database that you secure with a passphrase you can remember.

The passwords you generate have to be specified later in the instruction. For your convenience we provide the table below that you can copy and fill in before you start the installation. Where in the text it says “fill in password:fedora\_db\_admin” look up the corresponding password here.

Table Passwords

|  |  |
| --- | --- |
| **Name** | **Password** |
| fedora\_db\_admin |  |
| fedoraAdmin |  |
| easy\_db\_admin |  |
| easy\_webui |  |
| easy\_sword |  |
| easy\_proai |  |
| easy\_rest |  |
| easy\_ebiu |  |
| fedoraIntCallUser |  |
| ldapadmin |  |
| easyadmin |  |
| proai\_db\_admin |  |

## Conventions

The remainder of this Guide consists of a step-by-step instruction. We use the following conventions:

* To indicate input or output on the command line the courier font is used;
* input is preceded by a prompt, like this:

$ sudo vi example.txt

The prompt must *not* be typed on the command line (it, or a different prompt such as #, should already be there);

* what follows a line with a prompt is expected output. Note however that the output might be slightly different on your system;
* if the contents of a configuration file must be changed the relevant section is displayed in the courier font with the changed parts in **bold**;
* some commands are included in order to check the results of previous commands (e.g., sudo chkconfig --list slapd); it should be obvious which ones are.

# Standard Software Components

The following industry standard software components need to be installed first. See subsections for comments about alternatives and additional configuration. The items in this section can typically be performed by the IT department.

## Redhat 6 or CentOS 6

We recommend that you run the operation system in SELinux “protected mode.”

## Oracle Java SE 7 SDK (CentOS)

If you are working on RedHat, skip to 2.3 Oracle Java SE 7 SDK (RedHat) for an easier installation.

### Download the JDK

Download “jdk-7u*XX*-linux-x64.rpm” from the Oracle website (where *XX* is the latest update number):

<http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html>

### Run installer

Upload the rpm-file to your server with scp or sftp and run the installer:

$ sudo rpm -i jdk-7u*XX*-linux-x64.rpm

Unpacking JAR files...

rt.jar...

jsse.jar...

charsets.jar...

tools.jar...

localedata.jar...

jfxrt.jar...

### Add the JAVA\_HOME environment variable

Copy the file $EASY\_BACKEND/util/java.sh to /etc/profile.d and run it:

$ sudo cp java.sh /etc/profile.d/

$ exit

Now, log off and on to add the JAVA\_HOME variable to your environment.

$ echo $JAVA\_HOME

/usr/java/default/

### Add java to alternatives

CentOS comes default with OpenJDK. Add Oracle JDK to alternatives and activate it:

$ sudo alternatives --install /usr/bin/java java /usr/java/default/bin/java 2

$ sudo alternatives --config java

Er zijn 3 programma's die 'java' leveren.

Selectie Commando

-----------------------------------------------

\*+ 1 /usr/lib/jvm/jre-1.6.0-openjdk.x86\_64/bin/java

2 /usr/java/default/bin/java

3 /usr/lib/jvm/jre-1.5.0-gcj/bin/java

<enter> om de huidige selectie te bewaren[+], of type een selectie nummer: 2

$ java -version

java version "1.7.0\_51"

Java(TM) SE Runtime Environment (build 1.7.0\_51-b13)

Java HotSpot(TM) 64-Bit Server VM (build 24.51-b03, mixed mode)

Make sure the output does not mention “OpenJDK”.

### Notes

* Version 6 will work as well;
* OpenJDK might work as well, but has not been tested.

## Oracle Java SE 7 SDK (RedHat)

<installatie via yum>

## Tomcat 6

### Install Tomcat 6

Execute the following command:

$ sudo yum install tomcat6 tomcat6-webapps tomcat6-admin-webapps

Loaded plugins: fastestmirror, security

base | 3.7 kB 00:00

base/primary\_db | 4.4 MB 00:01

extras | 3.4 kB 00:00

# .. more output, respond with y to prompts

Complete!

### Give the Tomcat 6 jvm more memory to work with

Add the following line to /etc/tomcat6/tomcat6.conf (just below the line that starts with “JAVA\_OPTS=”):

$ sudo vi /etc/tomcat6/tomcat6.conf

JAVA\_OPTS="${JAVA\_OPTS} -Xmx2048m -Xms2048m -server -XX:PermSize=256m \

-XX:MaxPermSize=256m -XX:+AggressiveHeap"

*Look out when copy-pasting the above, the backslash seems to confuse Tomcat, so you had better put everything on one line.*

### Configure Tomcat 6 to expect UTF-8 in percent-encoded bytes

Configure all the connectors you specify in /etc/tomcat6/server.xml to use the UTF-8 encoding, by means of the attribute: URIEncoding="UTF-8". When adding an AJP-connector to connect Tomcat to Apache HTTP Server (see next step) don’t forget to also configure it.

$ sudo vi /etc/tomcat6/server.xml

…

<Connector port="8080" protocol="HTTP/1.1"

connectionTimeout="20000"

redirectPort="8443"

**URIEncoding="UTF-8"**/>

…

<Connector port="8009" protocol="AJP/1.3" redirectPort="8443" **URIEncoding="UTF-8"**/>

### Configure the Tomcat daemon to start automatically

Configure the Tomcat daemon to start automatically at system startup :

$ sudo chkconfig tomcat6 on

$ sudo chkconfig --list tomcat6

tomcat6 0:uit 1:uit 2:**aan** 3:**aan** 4:**aan** 5:**aan** 6:uit

*Do not start the Tomcat daemon yet. We need to configure our web applications before they are deployed.*

## Apache HTTP Server 2.2.15

### Install Apache HTTP Server (?)

Of is dat standaard geïnstalleerd?

### Set up Apache HTTP Server to as Tomcat proxy

Configure to … with Tomcat 6. (invullen door Arnoud)

## PostGreSQL 8.4

### Install PostGreSQL

Execute the following command:

$ sudo yum install postgresql-server.x86\_64

Loaded plugins: fastestmirror, security

Determining fastest mirrors

# .. more output, respond with y to prompts

Complete!

### Initialize the database

Initialize the database after installation:

$ sudo service postgresql initdb

Initializing database: [ OK ]

### Configure auto-vacuum (optional)

PostGreSQL by default doesn’t automatically garbage collect deleted rows. A DBA can start a garbage collect session (known as “vacuum”) manually. However, it is also possible to have PostGreSQL do this automatically.

Open the file /var/lib/pgsql/data/postgresql.conf and change the corresponding lines to look like below:

$ sudo vi /var/lib/pgsql/data/postgresql.conf

# ..

# - Query/Index Statistics Collector -

#track\_activities = on

**track\_counts = on**

#track\_functions = none # none, pl, all

#track\_activity\_query\_size = 1024

#update\_process\_title = on

#stats\_temp\_directory = 'pg\_stat\_tmp'

# - Statistics Monitoring -

#log\_parser\_stats = off

#log\_planner\_stats = off

#log\_executor\_stats = off

#log\_statement\_stats = off

#------------------------------------------------------------------------------

# AUTOVACUUM PARAMETERS

#------------------------------------------------------------------------------

**autovacuum = on # Enable autovacuum subprocess? 'on'**

**# requires track\_counts to also be on.**

#log\_autovacuum\_min\_duration = -1 # -1 disables, 0 logs all actions and

# their durations, > 0 logs only

# actions running at least this number

# of milliseconds.

#autovacuum\_max\_workers = 3 # max number of autovacuum subprocesses

#autovacuum\_naptime = 1min # time between autovacuum runs

#autovacuum\_vacuum\_threshold = 50 # min number of row updates before

# vacuum

### Configure database to accept user/password credentials

Configure the database to accept local connections based on username/password credentials by editing the file /var/lib/pgsql/data/pg\_hba.conf. The “postgres” user (super user) will keep using the “ident” method for Unix domain sockets which means that the requesting process must be run by the “postgres” operating system user.

$ sudo vi /var/lib/pgsql/data/pg\_hba.conf

Change the lines at the bottom of the file to look like this:

# TYPE DATABASE USER CIDR-ADDRESS METHOD

# "local" is for Unix domain socket connections only

**local all postgres ident**

local all all **md5**

# IPv4 local connections:

host all all 127.0.0.1/32 **md5**

# IPv6 local connections:

host all all ::1/128 **md5**

### Start the daemon

Make the PostGreSQL daemon start by default:

$ sudo chkconfig postgresql on

$ sudo chkconfig --list postgresql

postgresql 0:uit 1:uit 2:**aan** 3:**aan** 4:**aan** 5:**aan** 6:uit

Start the daemon now:

$ sudo service postgresql start

Starting postgresql service: [ OK ]

## OpenLDAP 2.4

### Install OpenLDAP servers and clients

Execute the following command:

$ sudo yum install openldap-servers openldap-clients

Loaded plugins: fastestmirror, security

Loading mirror speeds from cached hostfile

# .. more output, respond with y to prompts

Complete!

### Remove the “default” database (optional)

The OpenLDAP installer configures a default user database. Since we are not going to use it, we will remove it. There does not seem to be a clean way (i.e. through the LDAP protocol) to do this yet, so we will remove the appropriate file from the config directory:

$ sudo rm /etc/openldap/slapd.d/cn\=config/olcDatabase\=\{2\}bdb.ldif

### Start the daemon

Make the OpenLDAP daemon start by default:

$ sudo chkconfig slapd on

$ sudo chkconfig --list slapd

slapd 0:uit 1:uit 2:**aan** 3:**aan** 4:**aan** 5:**aan** 6:uit

Start the daemon now:

$ sudo service slapd start

Starting slapd: [ OK ]

# EASY Back-end Modules

Now that we have the standard software in place we turn to the set-up and configuration of the back-end modules that support EASY. The items in this section should typically be performed by the technical support staff for your repository.

## EASY Fedora Commons Repository

The core component of EASY is the respository that stores the actual scientific research datasets. The repository is implemented using the Fedora Commons repository software. There are no standard (yum- or rpm-based) installation packages for Fedora Commons. The following steps are based on the instructions on the Fedora Commons website.[[2]](#footnote-2)

### Create a database for Fedora Commons in PostGreSQL

Use the file

$EASY\_BACKEND/easy-fedora-commons-repository/create-fedora-db.sql

On the command line execute the following command:

$ sudo -u postgres psql -U postgres < create-fedora-db.sql

CREATE ROLE

CREATE DATABASE

(Note: if you are in a directory that is inaccessible to the postgres user you may get a warning ‘could not change directory to “…”’ but this does not seem to prevent the database from being created.)

### Set the fedora\_db\_admin password

Set the password of the fedora\_db\_admin[[3]](#footnote-3) postgres user:

$ sudo -u postgres psql -U postgres

And then in postgres:

postgres# \password fedora\_db\_admin

Enter new password:

Enter it again:

postgres# \q

and fill in password:fedora\_db\_admin from Table 1 Passwords.

### Set the FEDORA\_HOME environment variable

Copy the file $EASY\_BACKEND/easy-fedora-commons-repository/fedora.sh to /etc/profile.d

$ sudo cp fedora.sh /etc/profile.d

and log off and on again. The FEDORA\_HOME environment variable should now point to /opt/fedora.

$ echo $FEDORA\_HOME

/opt/fedora

### Run the Fedora Commons installer

Download the Fedora Commons installer (fcrepo-installer-3.5.jar) from the Fedora Commons website at:

<https://wiki.duraspace.org/display/FEDORA35/Downloads>

Edit the file at:

$EASY\_BACKEND/easy-fedora-commons-repository/install.properties

$ sudo vi install.properties

* for database.password fill in password:fedora\_db\_admin
* for fedora.admin.pass fill in password:fedoraAdmin

Then execute the following command:

$ sudo java -jar fcrepo-installer-3.5.jar install.properties

WARNING: The environment variable, CATALINA\_HOME, is not defined

WARNING: Remember to define the CATALINA\_HOME environment variable

WARNING: before starting Fedora.

WARNING: The environment variable, FEDORA\_HOME, is not defined

WARNING: Remember to define the FEDORA\_HOME environment variable

WARNING: before starting Fedora.

Preparing FEDORA\_HOME...

Configuring fedora.fcfg

Installing beSecurity

Will not overwrite existing /usr/share/tomcat6/conf/server.xml.

Wrote example server.xml to:

/opt/fedora-3.5/install/server.xml

Preparing fedora.war...

Deploying fedora.war...

Installation complete.

----------------------------------------------------------------------

Before starting Fedora, please ensure that any required environment

variables are correctly defined

(e.g. FEDORA\_HOME, JAVA\_HOME, JAVA\_OPTS, CATALINA\_HOME).

For more information, please consult the Installation & Configuration

Guide in the online documentation.

----------------------------------------------------------------------

where “install.properties” is your edited copy of the install.properties files mentioned above.

You can safely ignore the warnings above.

After the installation change the ownership of installation directory to tomcat:

$ sudo chown -R tomcat:tomcat /opt/fedora-3.5

### Create a symbolic link to the fedora installation

Create a symbolic link to the /opt/fedora-3.5:

$ sudo ln -s /opt/fedora-3.5 /opt/fedora

$ ls -l /opt/

totaal 8

lrwxrwxrwx. 1 root root 15 mrt 2 01:43 fedora -> /opt/fedora-3.5

drwxr-xr-x. 7 tomcat tomcat 4096 mrt 2 01:24 fedora-3.5

Now, if you want to switch to another installed version of Fedora Commons you will only need to point this link to the appropriate directory; the FEDORA\_HOME environment variable will automatically point to the same directory.

### Create and configure location of data store and resource index

In this example we will assume that the Fedora objects and datastreams will be located in /data/fedora/objects and /data/fedora/datastreams respectively and that the resoure index will store its data in /data/fedora/resourceIndex.

First, make sure the target locations exist, if they don’t, create them and change ownership to the tomcat user:

$ sudo mkdir -p /data/fedora/objects /data/fedora/datastreams /data/fedora/fedora-xacml-policies/repository-policies/default /data/fedora/resourceIndex

$ sudo chown -R tomcat:tomcat /data/fedora

$ ls -l /data/fedora/

totaal 16

drwxr-xr-x. 2 tomcat tomcat 4096 mrt 2 01:45 datastreams

drwxr-xr-x. 3 tomcat tomcat 4096 mrt 2 01:45 fedora-xacml-policies

drwxr-xr-x. 2 tomcat tomcat 4096 mrt 2 01:45 objects

drwxr-xr-x. 2 tomcat tomcat 4096 mrt 2 01:45 resourceIndex

Note that the policies directory does need to exist, even though we don’t customize the policy mechanism.

Edit the file $FEDORA\_HOME/server/config/fedora.fcfg, and change the following items:

* In the module with the attribute role="org.fcrepo.server.storage.lowlevel.ILowlevelStorage", change the value of the “object\_store\_base” param to “/data/fedora/objects” and change the value of the param “datastream\_store\_base” to “/data/fedora/datastreams”
* In the datastore with the attribute id="localMulgaraTriplestore", change the value of the “path” param to “/data/fedora/resourceIndex”

$ sudo vi /opt/fedora/server/config/fedora.fcfg

<module role="org.fcrepo.server.storage.lowlevel.ILowlevelStorage"

class="org.fcrepo.server.storage.lowlevel.DefaultLowlevelStorageModule">

<param name="path\_algorithm"

value="org.fcrepo.server.storage.lowlevel.TimestampPathAlgorithm">

<comment>The java class used to determine the path algorithm;

default is org.fcrepo.server.storage.lowlevel.

TimestampPathAlgorithm.</comment>

</param>

<param name="object\_store\_base" value="**/data/fedora/objects**" isFilePath="true">

<comment>The root directory for the internal storage of Fedora objects.

This value should be adjusted based on your installation

environment. This value should not point to the same location as

datastream\_store\_base.</comment>

</param>

<param name="backslash\_is\_escape" value="true">

<comment>Whether the escape character (i.e. (the token beginning an

escape sequence) for the backing database (which includes

registry tables) is the backslash character. This is needed to

correctly store and retrieve filepaths from the registry

tables, if running under Windows/DOS. (Set to true for MySQL and

Postgresql, false for Derby and Oracle)</comment>

</param>

<param name="datastream\_store\_base" value="**/data/fedora/datastreams**"

isFilePath="true">

<comment>The root directory for the internal storage of Managed

Content datastreams. This value should be adjusted based on your

installation environment. This value should not point to the same

location as object\_store\_base.</comment>

…

<datastore id="localMulgaraTriplestore">

<comment>local Mulgara Triplestore used by the Resource Index</comment>

<param name="poolInitialSize" value="3">

<comment>The initial size of the session pool used for queries.

Note: A value of 0 will cause the Resource Index to operate in

synchronized mode: concurrent read/write requests are put in a queue

and handled in FIFO order; this will severely impair performance and

is only intended for debugging.</comment>

</param>

# more params

<param name="path" value="**/data/fedora/resourceIndex**" isFilePath="true">

<comment>The local path to the main triplestore directory.</comment>

</param>

### Add Fedora Commons users

So far we only have fedoraAdmin user. We will use different users for different services connecting to Fedora Commons. Edit the file

$FEDORA\_HOME/server/config/fedora-users.xml and add user elements for users easy\_webui, easy\_sword, easy\_rest, easy\_proai, eays\_ebiu. Give them the role administrator and fill in the password from Table 1 Passwords.

$ sudo vi /opt/fedora/server/config/fedora-users.xml

<?xml version='1.0' ?>

<users>

<user name="fedoraAdmin" password="**password:fedoraAdmin**">

<attribute name="fedoraRole">

<value>administrator</value>

</attribute>

</user>

**<user name="easy\_webui" password="password:easy\_webui">**

**<attribute name="fedoraRole">**

**<value>administrator</value>**

**</attribute>**

**</user>**

**<user name="easy\_sword" password="password:easy\_sword">**

**<attribute name="fedoraRole">**

**<value>administrator</value>**

**</attribute>**

**</user>**

**<user name="easy\_rest" password="password:easy\_rest">**

**<attribute name="fedoraRole">**

**<value>administrator</value>**

**</attribute>**

**</user>**

**<user name="easy\_proai" password="password:easy\_proai">**

**<attribute name="fedoraRole">**

**<value>administrator</value>**

**</attribute>**

**</user>**

**<user name="easy\_ebiu" password="password:easy\_ebiu">**

**<attribute name="fedoraRole">**

**<value>administrator</value>**

**</attribute>**

**</user>**

<user name="fedoraIntCallUser" password="**password:fedoraIntCallUser**">

<attribute name="fedoraRole">

<value>fedoraInternalCall-1</value>

<value>fedoraInternalCall-2</value>

</attribute>

</user>

</users>

It may seem useless to create extra users if they are all going to be admins anyway. However, in the future we assign different roles to these users with more restricted privileges.

### Change password of fedoraIntCallUser

The fedoraIntCallUser is a user that Fedora Commons uses internally to make calls to itself. By default it has the unsafe password “changeme”. We will change it to a safe password.

We have already edited the file $FEDORA\_HOME/server/config/fedora-users.xml.

For fedoraIntCallUser we also need to edit $FEDORA\_HOME/server/config/beSecurity.xml to assign the same password from Table 1 Passwords.

$ sudo vi /opt/fedora/server/config/beSecurity.xml

<?xml version="1.0" encoding="UTF-8"?>

<serviceSecurityDescription

xmlns="info:fedora/fedora-system:def/beSecurity#"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="info:fedora/fedora-system:def/beSecurity#

http://www.fedora.info/definitions/1/0/api/beSecurity.xsd"

role="default">

<serviceSecurityDescription

role="fedoraInternalCall-1"

callSSL="false"

callBasicAuth="false"

callUsername="fedoraIntCallUser"

callPassword="**password:fedoraIntCallUser**"

callbackSSL="false"

callbackBasicAuth="false"

iplist="127.0.0.1"/>

<serviceSecurityDescription

role="fedoraInternalCall-2"

callSSL="false"

callBasicAuth="false"

callbackSSL="false"

callbackBasicAuth="false"

iplist="127.0.0.1"/>

</serviceSecurityDescription>

### Limit access to passwords

Several configuration files contain passwords. We need to limit read rights for security:

$ sudo chmod 0600 $FEDORA\_HOME/server/config/fedora-users.xml

$ sudo chmod 0600 $FEDORA\_HOME/server/config/fedora.fcfg

$ sudo chmod 0600 $FEDORA\_HOME/server/config/beSecurity.xml

$ ls -l $FEDORA\_HOME/server/config/

totaal 124

-rw-r--r--. 1 tomcat tomcat 1403 mrt 2 07:24 activemq.xml

**-rw-------. 1 tomcat tomcat 805 mrt 2 07:24 beSecurity.xml**

-rw-r--r--. 1 tomcat tomcat 4757 mrt 2 07:24 config-melcoe-pep-mapping.xml

-rw-r--r--. 1 tomcat tomcat 16670 mrt 2 07:24 config-melcoe-pep.xml

**-rw-------. 1 tomcat tomcat 55854 mrt 2 2014 fedora.fcfg**

**-rw-------. 1 tomcat tomcat 1183 mrt 2 07:24 fedora-users.xml**

-rw-r--r--. 1 tomcat tomcat 1266 mrt 2 07:24 jaas.conf

-rw-r--r--. 1 tomcat tomcat 2163 mrt 2 07:24 logback.xml

-rw-r--r--. 1 tomcat tomcat 15082 mrt 2 07:24 mime-to-extensions.xml

drwxr-xr-x. 3 tomcat tomcat 4096 mrt 2 07:24 spring

It seems that chmod 0400 (only read-access to owner) is too restrictive. I am not sure why, but read/write access by the owner (0600) is safe enough.

### Deploy Saxon (Optional)

EASY datasets are declared in their RELS-EXT datastream to be OAI-PMH items. This means that they can be queried for metadata in several formats by an OAI-PMH data provider. If you are going to set up such a provider (see: 4.4 EASY Customized PrOAI Module (Optional) on page 33) you will also need the Saxon XSLT Processing Service. In this scenario we use the version provided in

$EASY\_BACKEND/easy-fedora-commons-repository/saxon

First create the installation directory:

$ sudo mkdir -p /opt/saxon/log

Then copy the file fcrepo-webapp-saxon-3.6.war to /opt/saxon.

$ sudo cp fcrepo-webapp-saxon-3.6.war /opt/saxon

Then copy the Tomcat context container saxon.xml to /etc/tomcat6/Catalina/localhost.

$ sudo cp saxon.xml /etc/tomcat6/Catalina/localhost

Finally, change ownership of /opt/saxon to the tomcat user:

$ sudo chown -R tomcat:tomcat /opt/saxon

### Start Tomcat 6

Finally we are ready to start up Tomcat 6 (we will tail the Tomcat log file to see if everything goes well):

$ sudo service tomcat6 start; tail -f /var/log/tomcat6/catalina.out

Starting tomcat6: [ OK ]

Mar 02, 2014 8:11:12 AM org.apache.catalina.startup.HostConfig deployDirectory

INFO: Deploying web application directory sample

Mar 02, 2014 8:11:12 AM org.apache.coyote.http11.Http11Protocol start

INFO: Starting Coyote HTTP/1.1 on http-8080

Mar 02, 2014 8:11:12 AM org.apache.jk.common.ChannelSocket init

INFO: JK: ajp13 listening on /0.0.0.0:8009

Mar 02, 2014 8:11:12 AM org.apache.jk.server.JkMain start

INFO: Jk running ID=0 time=0/63 config=null

Mar 02, 2014 8:11:12 AM org.apache.catalina.startup.Catalina start

INFO: Server startup in 12985 ms

### Add the basic EASY digital objects

In order to run, EASY needs a minimal set of Fedora Commons digital objects. These are provided in:

$EASY\_BACKEND/easy-fedora-commons-repository/basic-digital-objects

Change directory to this folder and execute the following command, replacing <password:fedoraAdmin> the corresponding entry from Table 1 Passwords.

*Look out: if the password contains dollar signs it must be in single quotes, and any dollar signs in it must be escaped with a backslash (this may also be true for other “special” characters in the password)*

$ fedora-batch-ingest.sh . ~/ingest.log text info:fedora/fedora-system:FOXML-1.1 localhost:8080 fedoraAdmin **<password:fedoraAdmin>** http

ingest succeeded for: easy-collection:1.xml

ingest succeeded for: easy-discipline:18.xml

# .. more output

ingest succeeded for: easy-discipline:27.xml

ingest succeeded for: easy-data:oai-repository1.xml

Batch Ingest Summary

74 files processed in this batch

74 objects successfully ingested into Fedora

0 objects failed

0 unexpected files in directory

0 files ignored after error

### Add EASY PrOAI Module support

For EASY PrOAI Module support we need to add some more digital objects. Go to the directory

$EASY\_BACKEND/easy-fedora-commons-repository/oai-pmh

Then execute the following command:

$ sudo ./add-oai-pmh-support.sh <password:fedoraAdmin>

Ingested pid: dans-xsl:1

# .. more output

# 5 POST messages with response 201 Created

## EASY LDAP Directory

The EASY LDAP Directory component, apart from an LDAP daemon, consists some EASY-specific schema’s and a few basic entry’s. We will add those here, using the standard LDAP client tools.

### Create a separate directory folder for EASY

To keep things neat and tidy, we will give EASY its own directory:

$ sudo mkdir /var/lib/ldap/easy; sudo chown ldap:ldap /var/lib/ldap/easy

$ ls /var/lib/ldap/ -l

totaal 4

drwxr-xr-x. 2 ldap ldap 4096 mrt 2 08:51 easy

### Add DANS and EASY schema’s

The schema’s are added using LDIF files that can be found in:

$EASY\_BACKEND/easy-ldap-directory.

Execute the following commands:

$ sudo ldapadd -v -Y EXTERNAL -H ldapi:/// -f dans-schema.ldif

ldap\_initialize( ldapi:///??base )

SASL/EXTERNAL authentication started

SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth

SASL SSF: 0

add objectClass:

olcSchemaConfig

add cn:

dans

add olcAttributeTypes:

{0}( 1.3.6.1.4.1.33188.0.1.1 NAME 'dansState' DESC 'The state of an entity'

# .. more simliar output

adding new entry "cn=dans,cn=schema,cn=config"

modify complete

$ sudo ldapadd -v -Y EXTERNAL -H ldapi:/// -f easy-schema.ldif

ldap\_initialize( ldapi:///??base )

SASL/EXTERNAL authentication started

SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth

SASL SSF: 0

add objectClass:

olcSchemaConfig

add cn:

easy

add olcAttributeTypes:

{0}( 1.3.6.1.4.1.33188.1.1.3 NAME 'easyAcceptConditionsOfUse' DESC 'accepts the Conditions of Use of DANS-EASY' EQUALITY booleanMatch SYNTAX 1.3.6.1.4.1.1466.115.121.1.7 SINGLE-VALUE )

# .. more simlar output

adding new entry "cn=easy,cn=schema,cn=config"

modify complete

### Add EASY database

First we add the EASY database configuration to the config directory:

$ sudo ldapadd -v -Y EXTERNAL -H ldapi:/// -f easy-db.ldif

ldap\_initialize( ldapi:///??base )

SASL/EXTERNAL authentication started

SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth

SASL SSF: 0

add objectClass:

olcDatabaseConfig

# .. more similar output

adding new entry "olcDatabase=bdb,cn=config"

modify complete

### Add basic entries to the EASY database

To run EASY needs a minimal set of entries in its LDAP directory. Those entries are provided in the easy-basis.ldif file.

Before running the following command replace the string “FILL.IN.YOUR@VALID-EMAIL.NL” with the e-mail address of the use that is going to be the administrator of the EASY installation, for example your own e-mail address.

$ sudo vi easy-basis.ldif

…

dn: uid=easyadmin,ou=users,ou=easy,dc=dans,dc=knaw,dc=nl

objectClass: top

objectClass: person

objectClass: organizationalPerson

objectClass: inetOrgPerson

objectClass: dansUser

objectClass: easyUser

cn: easyadmin

sn: Admin

uid: easyadmin

dansAcceptConditionsOfUse: TRUE

dansNewsletter: FALSE

dansState: ACTIVE

displayName: EASY Admin

easyHasConfirmedGeneralConditions: FALSE

easyLogMyActions: TRUE

easyRoles: ARCHIVIST

easyRoles: ADMIN

easyRoles: USER

givenName: EASY

initials: E A

l: Den Haag

mail: **FILL.IN.YOUR@VALID-EMAIL.NL**

o: DANS

postalAddress: Anna van Saksenlaan 51

postalCode: 2593 HT

# DEFAULT PASSWORD: root

userPassword: {SSHA}C6iqKQ2A76f0DYlpkNjTHKZ0c9RmYjTO

Then execute the following command:

$ sudo ldapadd -W -D cn=ldapadmin,dc=dans,dc=knaw,dc=nl -f easy-basis.ldif

Enter LDAP Password: secret

adding new entry "dc=dans,dc=knaw,dc=nl"

adding new entry "ou=easy,dc=dans,dc=knaw,dc=nl"

adding new entry "ou=groups,ou=easy,dc=dans,dc=knaw,dc=nl"

adding new entry "ou=users,ou=easy,dc=dans,dc=knaw,dc=nl"

adding new entry "ou=migration,ou=easy,dc=dans,dc=knaw,dc=nl"

adding new entry "ou=federation,ou=easy,dc=dans,dc=knaw,dc=nl"

adding new entry "ou=Archeology,ou=groups,ou=easy,dc=dans,dc=knaw,dc=nl"

adding new entry "ou=History,ou=groups,ou=easy,dc=dans,dc=knaw,dc=nl"

adding new entry "uid=easyadmin,ou=users,ou=easy,dc=dans,dc=knaw,dc=nl"

We are using the OpenLDAP user “cn=ldapadmin,dc=dans,dc=knaw,dc=nl”. This is the administrator of the EASY LDAP Directory. The default password of this user is “secret” (we will change that in a moment, but you need it to complete this command).

### Change the ldapadmin password

The default password of the ldapadmin user is of course complete non-self-describing, so we will change it here. First, generate a safe password, then execute the following command

$ slappasswd -h {SSHA}

and enter <password:ldapadmin> from Table 1 Passwords when prompted to do so. Copy the resulting hash and replace the hash in the file “change-ldapadmin-pw.ldif” (the part in bold):

$ sudo vi change-ldapadmin-pw.ldif

dn: olcDatabase={2}bdb,cn=config

changetype: modify

replace: olcRootPW

olcRootPW: **{SSHA}ZrVZQ66Y7qzCKGg1I5iX4Qq//s7oosHw**

Then, execute this command:

$ sudo ldapadd -v -Y EXTERNAL -H ldapi:/// -f change-ldapadmin-pw.ldif

ldap\_initialize( ldapi:///??base )

SASL/EXTERNAL authentication started

SASL username: gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth

SASL SSF: 0

replace olcRootPW:

{SSHA}9dQ07izka8farPzfFJHQg4YTSgjDAwVN

modifying entry "olcDatabase={2}bdb,cn=config"

modify complete

### Change the easyadmin user’s application password

The file “easy-basis.ldif,” which we added earlier, added the administrator user for the EASY application: easyadmin. The default password for this user is also

“easyadmin.” This needs to be replaced by a safe password.

Execute:

$ slappasswd -h {SSHA}

and enter <password:easyadmin> from Table 1 Passwords when prompted to do so. Edit the file “change-easyadmin-user-pw.ldif” and replace the password hash with the one calculated by slappasswd:

$ sudo vi change-easyadmin-user-pw.ldif

dn: uid=easyadmin,ou=users,ou=easy,dc=dans,dc=knaw,dc=nl

changetype: modify

replace: userPassword

userPassword: **{SSHA}VzBuoiJKS46ZIiTmvAHkj4C92qE749YR**

Then execute this command:

$ sudo ldapadd -W -D cn=ldapadmin,dc=dans,dc=knaw,dc=nl -f change-easyadmin-user-pw.ldif

Enter LDAP Password:

modifying entry "uid=easyadmin,ou=users,ou=easy,dc=dans,dc=knaw,dc=nl"

Don’t forget that you have to use your new ldapadmin-password now!

## EASY Filesystem RDB

EASY depends on a relational database to store a (redundant) model of the file and folder structure of each dataset. The same information is present in the digital objects that represent the parts of this structure. However, the Resource Index that indexes these statements was found to perform unsufficiently to supply the Web-UI with this information.

### Creating the database and tables

We will now create the database to store this information. Use the files

$EASY\_BACKEND/easy-filesystem-rdb/create-easy-db.sql

$EASY\_BACKEND/easy-filesystem-rdb/create-easy-db-tables.sql

and execute the following command:

$ sudo -u postgres psql -U postgres < create-easy-db.sql

CREATE ROLE

CREATE DATABASE

$ sudo -u postgres psql easy\_db -U postgres < create-easy-db-tables.sql

SET

SET

SET

SET

SET

SET

CREATE TABLE

ALTER TABLE

CREATE TABLE

ALTER TABLE

CREATE TABLE

ALTER TABLE

CREATE TABLE

ALTER TABLE

CREATE TABLE

ALTER TABLE

ALTER TABLE

ALTER TABLE

ALTER TABLE

ALTER TABLE

ALTER TABLE

CREATE INDEX

CREATE INDEX

CREATE INDEX

CREATE INDEX

CREATE INDEX

CREATE INDEX

CREATE INDEX

CREATE INDEX

CREATE INDEX

CREATE INDEX

CREATE INDEX

CREATE INDEX

CREATE INDEX

ALTER TABLE

ALTER TABLE

ALTER TABLE

CREATE ROLE

CREATE ROLE

CREATE ROLE

CREATE ROLE

GRANT

GRANT

### Assigning passwords

The previous commands created the database and users. We will now assign passwords to the users. Execute the following commands:

$ sudo -u postgres psql -U postgres

# \password easy\_db\_admin

# \password easy\_webui

# \password easy\_sword

# \password easy\_rest

# \q

and fill in the corresponding passwords from Table 1 Passwords. The easy\_db\_admin user is intented to be used for administrative actions on the database through the psql command line client (which should in principle never be necessary). The other users are used by the application or service with the same name. Notice that we use the same passwords as for access to the EASY Fedora Commons Repository. This is not mandatory, but just to keep the number of passwords limited.

## EASY PID RDB

EASY assigns a unique persistent identifier (PID) to each submitted dataset. To keep track of the last assigned PID EASY uses a separate database.

### Creating the database and tables

We will now create the database to store this information. Use the files

$EASY\_BACKEND/easy-pid-rdb/create-pid-db.sql

$EASY\_BACKEND/easy-pid-db/create-pid-db-tables.sql

and execute the following command:

$ sudo -u postgres psql -U postgres < create-pid-db.sql

CREATE DATABASE

$ sudo -u postgres psql -U postgres pid\_db < create-pid-db-tables.sql

CREATE TABLE

ALTER TABLE

CREATE INDEX

INSERT 0 1

GRANT

### Assigning passwords

The admin of this database is set to easy\_db\_admin and the users easy\_webui and easy\_sword get the privileges they need to use this database. Therefore there is no need to set additional passwords.

## EASY SOLR Search Index

### Install Apache SOLR 3.5

Download Apache SOLR 3.5 from <http://archive.apache.org/dist/lucene/solr/3.5.0/apache-solr-3.5.0.tgz>

and unzip it to /opt:

$ sudo tar -xzf apache-solr-3.5.0.tgz -C /opt

This will create the directory /opt/apache-solr-3.5.0.

After the installation change the ownership of installation directory to tomcat:

$ sudo chown -R tomcat:tomcat /opt/apache-solr-3.5.0

**drwxr-xr-x. 7 tomcat tomcat 4096 mrt 2 09:40 apache-solr-3.5.0**

lrwxrwxrwx. 1 root root 15 mrt 2 07:43 fedora -> /opt/fedora-3.5

drwxr-xr-x. 8 tomcat tomcat 4096 mrt 2 08:11 fedora-3.5

drwxr-xr-x. 2 root root 4096 jun 22 2012 rh

drwxr-xr-x. 3 tomcat tomcat 4096 mrt 2 07:37 saxon

### Create a symbolic link to the SOLR installation and war

As we did for fedora, we will create a symbolic link to the current installation:

$ sudo ln -s /opt/apache-solr-3.5.0 /opt/apache-solr

$ ls -l /opt

totaal 16

**lrwxrwxrwx. 1 root root 22 mrt 2 09:42 apache-solr -> /opt/apache-solr-3.5.0**

drwxr-xr-x. 7 tomcat tomcat 4096 mrt 2 09:40 apache-solr-3.5.0

lrwxrwxrwx. 1 root root 15 mrt 2 07:43 fedora -> /opt/fedora-3.5

drwxr-xr-x. 8 tomcat tomcat 4096 mrt 2 08:11 fedora-3.5

drwxr-xr-x. 2 root root 4096 jun 22 2012 rh

drwxr-xr-x. 3 tomcat tomcat 4096 mrt 2 07:37 saxon

$ sudo ln -s /opt/apache-solr-3.5.0/dist/apache-solr-3.5.0.war /opt/apache-solr/solr.war

$ ls -l /opt/apache-solr/

totaal 284

-rw-r--r--. 1 tomcat tomcat 156267 nov 22 2011 CHANGES.txt

drwxr-xr-x. 3 tomcat tomcat 4096 mrt 2 09:40 client

drwxr-xr-x. 9 tomcat tomcat 4096 nov 22 2011 contrib

drwxr-xr-x. 3 tomcat tomcat 4096 mrt 2 09:40 dist

drwxr-xr-x. 5 tomcat tomcat 4096 mrt 2 09:40 docs

drwxr-xr-x. 11 tomcat tomcat 4096 mrt 2 09:40 example

-rw-r--r--. 1 tomcat tomcat 80058 nov 22 2011 LICENSE.txt

-rw-r--r--. 1 tomcat tomcat 17111 nov 22 2011 NOTICE.txt

-rw-r--r--. 1 tomcat tomcat 4917 nov 22 2011 README.txt

**lrwxrwxrwx. 1 root root 49 mrt 2 09:43 solr.war -> /opt/apache-solr-3.5.0/dist/apache-solr-3.5.0.war**

### Create and the solr.home-directory

Now we create the directory were SOLR will store its index:

$ sudo mkdir -p /data/solr/cores/datasets/data /data/solr/cores/datasets/conf

Copy the file solr.xml in

$EASY\_BACKEND/easy-solr-search-index/config-all to the /data/solr directory:

$ sudo cp solr.xml /data/solr

Copy the files schema.xml, solrconfig.xml, stopwords.txt, synonyms.txt, protwords.txt in

$EASY\_BACKEND/easy-solr-search-index/config-datasets to the /data/solr/cores/datasets/conf directory:

$ sudo cp schema.xml solrconfig.xml stopwords.txt synonyms.txt protwords.txt /data/solr/cores/datasets/conf

Now set ownerschip of the whole directory tree to the tomcat user:

$ sudo chown -R tomcat:tomcat /data/solr

$ ls -l /data/solr/

totaal 8

drwxr-xr-x. 3 tomcat tomcat 4096 mrt 2 09:43 cores

-rw-r--r--. 1 tomcat tomcat 1386 mrt 2 09:47 solr.xml

$ ls -l /data/solr/cores/datasets/

totaal 8

drwxr-xr-x. 2 tomcat tomcat 4096 mrt 2 09:48 conf

drwxr-xr-x. 2 tomcat tomcat 4096 mrt 2 09:43 data

$ ls -l /data/solr/cores/datasets/conf/

totaal 40

-rw-r--r--. 1 tomcat tomcat 873 mrt 2 09:49 protwords.txt

-rw-r--r--. 1 tomcat tomcat 15919 mrt 2 09:49 schema.xml

-rw-r--r--. 1 tomcat tomcat 10693 mrt 2 09:49 solrconfig.xml

-rw-r--r--. 1 tomcat tomcat 781 mrt 2 09:49 stopwords.txt

-rw-r--r--. 1 tomcat tomcat 1133 mrt 2 09:49 synonyms.txt

### Copy the Tomcat 6 context container

Copy the solr.xml in

$EASY\_BACKEND/easy-solr-search-index/**config-tomcat** (don’t confuse with the previous file of the same name) to the directory /etc/tomcat6/Catalina/localhost:

$ sudo cp solr.xml /etc/tomcat6/Catalina/localhost; tail -f /var/log/tomcat6/catalina.out

INFO: Deploying configuration descriptor solr.xml

Mar 02, 2014 9:52:03 AM org.apache.solr.core.SolrResourceLoader locateSolrHome

INFO: Using JNDI solr.home: /data/solr

Mar 02, 2014 9:52:03 AM org.apache.solr.core.SolrResourceLoader <init>

INFO: Solr home set to '/data/solr/'

# .. more output

Mar 02, 2014 9:52:28 AM org.apache.solr.servlet.SolrServlet init

INFO: SolrServlet.init() done

Again, we are tailing the Tomcat 6 log to see if the deployment goes well.

# EASY Frond-end Modules

Now that we have the back-end services up and running, we can deploy the front-end services.

## EASY Web-UI Application

The principal service is the Web User Inferace (Web-UI) application.

### Unzip the installation package to /opt

First unzip the installation package to the directory /opt and set ownership of the newly created subdirectory to the tomcat user:

$ sudo tar -xzf easy-webui-2.8.tar.gz -C /opt

$ sudo chown -R tomcat:tomcat /opt/easy-webui-2.8

### Create a symbolic link to the installation directory

Execute the following command:

$ sudo ln -s /opt/easy-webui-2.8 /opt/easy-webui

### Set the EASY\_WEBUI\_HOME environment variable

To set the home directory of the EASY Web-UI application edit the file /etc/tomcat6/tomcat6.conf. Below the command “If you wish to further customize your tomcat environment (…)” add the following line:

$ sudo vi /etc/tomcat6/tomcat6.conf

EASY\_WEBUI\_HOME=/opt/easy-webui

Now reload the Tomcat environment (i.e. stop and start it):

$ sudo service tomcat6 force-reload

Stopping tomcat6: [ OK ]

Starting tomcat6: [ OK ]

### Create a custom “editable” home directory

EASY Web-UI is a web application that has several texts (for pages and e-mails) that can be edited by a user with special rights. It is highly recommendable that you put these resources outside the EASY\_WEBUI\_HOME directory. Examples of these editable resources with the correct directory structure can be found in

$EASY\_WEBUI\_HOME/res/example/editable

To relocate these examples copy or move them to a different location, for example /opt/easy-webui-editable, set tomcat as the user and later configure EASY Web-UI to look in the new location (see next section):

$ sudo mv /opt/easy-webui/res/example/editable /opt/easy-webui-editable

$ sudo chown -R tomcat:tomcat /opt/easy-webui-editable-home

### Configure EASY Web-UI application settings

EASY Web-UI needs to be configured, among other things to provided the correct passwords when trying to connect to the back-end services. Edit the file

/opt/easy-webui/cfg/application.properties

Replacing the patterns ###<placeholder>### with the appropriate value. If the placeholder starts with “password:”, then look for the value in Table 1 Passwords, otherwise find a sensible value using the description in the comment above the property.

### Configure logging

EASY Web-UI uses Logback to log messages to output destinations of your choice. Edit the file $EASY\_WEBUI\_HOME/cfg/logback.xml to change the default settings.

By default a rolling file log is created in /var/log/easy-webui, so unless you change the location (by editing the property LOGDIR at the top of the file) you should make sure the directory /var/log/easy-webui exists and is writable for tomcat:

$ sudo mkdir /var/log/easy-webui

$ sudo chown -R tomcat:tomcat /var/log/easy-webui

### Limit access to passwords

As the application.properties file contains password you should make sure that access is limited to root and tomcat:

$ sudo chmod 0600 application.properties

### Deploy the webapp

The EASY Web-UI application is deployed using an Apache Tomcat context container. Copy this file to the directory /etc/tomcat6/Catalina/localhost:

$ sudo cp /opt/easy-webui/bin/ui.xml /etc/tomcat6/Catalina/localhost; tail -f /var/log/tomcat6/catalina.out

INFO: Deploying configuration descriptor ui.xml

# .., more output

INFO [ContainerBackgroundProcessor[StandardEngine[Catalina]]]: Sending admin mail on startup

INFO [ContainerBackgroundProcessor[StandardEngine[Catalina]]]: Application event ====> org.springframework.context.event.ContextRefreshedEvent[source=Root WebApplicationContext: startup date [Tue Mar 04 10:47:30 CET 2014]; root of context hierarchy]

## EASY SWORD-based Ingest Module (Optional)

The EASY SWORD-base Ingest Module or “EASY SWORD” for short is a machine-machine interface to ingest new datasets into the archive.

### Unzip the installation package to /opt

First unzip the installation package to the directory /opt and set ownership of the newly created subdirectory to the tomcat user:

$ sudo tar -xzf easy-sword-2.8.tar.gz -C /opt

$ sudo chown -R tomcat:tomcat /opt/easy-sword-2.8

### Create a symbolic link to the installation directory

Execute the following command:

$ sudo ln -s /opt/easy-sword-2.8 /opt/easy-sword

### Set the EASY\_SWORD\_HOME environment variable

To set the home directory of the EASY Web-UI application edit the file /etc/tomcat6/tomcat6.conf. Below the command “If you wish to further customize your tomcat environment (…)” add the following line:

$ sudo vi /etc/tomcat6/tomcat6.conf

EASY\_SWORD\_HOME=/opt/easy-sword

Now reload the Tomcat environment (i.e. stop and start it):

$ sudo service tomcat6 force-reload

Stopping tomcat6: [ OK ]

Starting tomcat6: [ OK ]

### Configure EASY SWORD-based Ingest Module application settings

EASY SWORD needs to be configured, among other things to provided the correct passwords when trying to connect to the back-end services. Edit the file

/opt/easy-sword/cfg/application.properties

Replacing the patterns ###<placeholder>### with the appropriate value. If the placeholder starts with “password:”, then look for the value in Table 1 Passwords, otherwise find a sensible value using the description in the comment above the property.

### Configure logging

EASY SWORD uses Logback to log messages to output destinations of your choice. Edit the file $EASY\_SWORD\_HOME/cfg/logback.xml to change the default settings.

By default a rolling file log is created in /var/log/easy-sword, so unless you change the location (by editing the property LOGDIR at the top of the file) you should make sure the directory /var/log/easy-sword exists and is writable for tomcat:

$ sudo mkdir /var/log/easy-sword

$ sudo chown -R tomcat:tomcat /var/log/easy-sword

### Limit access to passwords

As the application.properties file contains password you should make sure that access is limited to root and tomcat:

$ sudo chmod 0600 application.properties

### Deploy the webapp

The EASY SWORD module is deployed using an Apache Tomcat context container. Copy this file to the directory /etc/tomcat6/Catalina/localhost:

$ sudo cp /opt/easy-sword/bin/sword.xml /etc/tomcat6/Catalina/localhost; tail -f /var/log/tomcat6/catalina.out

INFO: Deploying configuration descriptor sword.xml

# .., more output

INFO [ContainerBackgroundProcessor[StandardEngine[Catalina]]]: Sending admin mail on startup

INFO [ContainerBackgroundProcessor[StandardEngine[Catalina]]]: Application event ====> org.springframework.context.event.ContextRefreshedEvent[source=Root WebApplicationContext: startup date [Tue Mar 04 10:47:30 CET 2014]; root of context hierarchy]

## EASY Experimental RESTful Module (Optional)

The EASY Experimental RESTful Module, “EASY REST” for short, is a machine-machine interface to retrieve metadata from EASY.

### Unzip the installation package to /opt

First unzip the installation package to the directory /opt and set ownership of the newly created subdirectory to the tomcat user:

$ sudo tar -xzf easy-rest-2.8.tar.gz -C /opt

$ sudo chown -R tomcat:tomcat /opt/easy-rest-2.8

### Create a symbolic link to the installation directory

Execute the following command:

$ sudo ln -s /opt/easy-rest-2.8 /opt/easy-rest

### Set the EASY\_REST\_HOME environment variable

To set the home directory of EASY REST edit the file /etc/tomcat6/tomcat6.conf. Below the command “If you wish to further customize your tomcat environment (…)” add the following line:

$ sudo vi /etc/tomcat6/tomcat6.conf

EASY\_REST\_HOME=/opt/easy-rest

Now reload the Tomcat environment (i.e. stop and start it):

$ sudo service tomcat6 force-reload

Stopping tomcat6: [ OK ]

Starting tomcat6: [ OK ]

### Configure EASY REST

EASY REST needs to be configured, among other things to provided the correct passwords when trying to connect to the back-end services. Edit the file

/opt/easy-rest/cfg/application.properties

Replacing the patterns ###<placeholder>### with the appropriate value. If the placeholder starts with “password:”, then look for the value in Table 1 Passwords, otherwise find a sensible value using the description in the comment above the property.

### Configure logging

EASY REST uses Logback to log messages to output destinations of your choice. Edit the file $EASY\_REST\_HOME/cfg/logback.xml to change the default settings.

By default a rolling file log is created in /var/log/easy-rest, so unless you change the location (by editing the property LOGDIR at the top of the file) you should make sure the directory /var/log/easy-rest exists and is writable for tomcat:

$ sudo mkdir /var/log/easy-rest

$ sudo chown -R tomcat:tomcat /var/log/easy-rest

### Limit access to passwords

As the application.properties file contains password you should make sure that access is limited to root and tomcat:

$ sudo chmod 0600 /opt/easy-rest/cfg/application.properties

## EASY Customized PrOAI Module (Optional)

EASY can function as an OAI-PMH data provider. In order for it to do so you need to install the PrOAI service. Because of several issues with the “official” version[[4]](#footnote-4) we are using a modified version.

### Unzip the installation package to /opt

First unzip the installation package to the directory /opt and set ownership of the newly created subdirectory to the tomcat user:

$ sudo tar -xzf easy-proai-1.0.tar.gz -C /opt

$ sudo chown -R tomcat:tomcat /opt/easy-proai-1.0

### Create a symbolic link to the installation directory

Execute the following command:

$ sudo ln -s /opt/easy-proai-1.0/ /opt/easy-proai

### Set the EASY\_PROAI\_HOME environment variable

To set the home directory of the EASY Web-UI application edit the file /etc/tomcat6/tomcat6.conf. Below the command “If you wish to further customize your tomcat environment (…)” add the following line:

$ sudo vi /etc/tomcat6/tomcat6.conf

EASY\_PROAI\_HOME=/opt/easy-proai

Now reload the Tomcat environment (i.e. stop and start it):

$ sudo service tomcat6

Stopping tomcat6: [ OK ]

Starting tomcat6: [ OK ]

### Create PrOAI data directory

$ sudo mkdir -p /data/proai

$ sudo chown -R tomcat:tomcat /data/proai

### Create PrOAI database

PrOAI uses a relational database to keep track of files in its cache. The database and the database user must exist before startup, but the tables are created dynamically. To create the database use the file:

$EASY\_PROAI\_HOME/install/create-proai-db.sql

Execute the following command:

$ sudo -u postgres psql -U postgres < create-proai-db.sql

Then set the password of the owner of this database (proai\_db\_admin):

$ sudo -u postgres psql -U postgres

and when logged in:

# \password proai\_db\_admin

and fill in password:proai\_db\_admin when prompted.

### Configure EASY Proai service settings

Edit the file $EASY\_PROAI\_HOME/cfg/proai.settings to specify the passwords for the database and Fedora Commons users:

* proai.db.password: fill in password:proai\_db\_admin
* driver.fedora.pass: fill in password:easy\_proai

Be sure to provide the password directly after the equals sign with no spaces around it!

### Configure EASY PrOAI logging

The logging settings can be specified in $EASY\_PROAI\_HOME/cfg/logback.xml. The default settings will write to files in /var/log/easy-proai, so let’s create that directory:

$ sudo mkdir -p /var/log/easy-proai

$ sudo chown -R tomcat:tomcat /var/log/easy-proai

### Deploy the Tomcat context container

Now we are ready to deploy EASY PrOAI to Tomcat. Go to the directory $EASY\_PROAI\_HOME/bin and execute the following command:

$ sudo cp oai.xml /etc/tomcat6/Catalina/localhost; tail -f /var/log/tomcat6/catalina.out

INFO: Deploying configuration descriptor oai.xml

It may take several minutes before EASY PrOAI starts filling its cache. Until then a “cannotDisseminateFormat” error is returned when trying to retrieve records from the service.

### Limit access to passwords

The configuration file proai.properties contains passwords. We need to limit read access to it to tomcat:

$ sudo chmod 0600 $EASY\_PROAI\_HOME/cfg/proai.properties

# EASY Tools For Data Managers

While the EASY front-end modules are intended primarily for use by customers of the archive the tools discussed in this chapter are meant to be used by data managers (archivists) working at the archive.

## EASY Batch Ingest & Update (EBIU)

The EASY Batch Ingest & Update tool can import large quantities of data and meta-data into the archive unattendedly.

### Unzip the installation package to /opt

First unzip the installation package to the directory /opt and set ownership of the newly created subdirectory to the root user:

$ sudo tar -xzvf easy-ebiu-2.8.tar.gz -C /opt

$ sudo chown -R root:root /opt/easy-ebiu-2.8

### Create a symbolic link to the installation directory

Execute the following command:

$ sudo ln -s /opt/easy-ebiu-2.8 /opt/easy-ebiu

### Configure EASY EBIU

EASY EBIU needs to be configured, among other things to provided the correct passwords when trying to connect to the back-end services. Edit the file

/opt/easy-ebiu/cfg/application.properties

Replacing the patterns ###<placeholder>### with the appropriate value. If the placeholder starts with “password:”, then look for the value in Table 1 Passwords, otherwise find a sensible value using the description in the comment above the property.

### Limit access to passwords

<to do: gebruik setuid?>

### Set up the tools for the users

You need to prepare the Linux accounts of the users who are going to use the EBIU tool. In order to doe so for a user “jdoe” execute the following command:

$ sudo /opt/easy-ebiu/setup/setup-for-user.sh /opt/easy-ebiu jdoe

/home/jdoe /home/jdoe

/home/jdoe

The first parameter is the EBIU installation directory, the second the Linux account name of the user. The script will terminate with an error if the input directory tree is already present or the path of the user has already been set up to include the EBIU tool.

1. Data Archiving and Networked Services, an institute of the Netherlands Academy for the Arts and Sciences (KNAW) and the Netherlands Research Organisation (NWO). (<http://www.dans.knaw.nl>) [↑](#footnote-ref-1)
2. See: <https://wiki.duraspace.org/display/FEDORA35/Installation+and+Configuration> [↑](#footnote-ref-2)
3. We have changed the default database admin name from fedoraAdmin to fedora\_db\_admin, as PostGreSQL seems to have problems with mixed case user names in some cases. [↑](#footnote-ref-3)
4. One of them being that it is not clear what is the “official” version. On SourceForge there is a 1.1.1 version (<http://proai.sourceforge.net/>) but the Fedora Commons project uses a different version (<https://github.com/fcrepo/proai>) [↑](#footnote-ref-4)