*version 2.8*Contents

1 Introduction 3

1.1 Installation packages 3

2 Overview of EASY 4

3 Standard Software Components 5

3.1 Redhat 6 or CentOS 6 5

3.2 Oracle Java SE 7 SDK 5

3.3 Tomcat 6 5

3.4 Apache HTTP Server 2.2.15 6

3.5 PostGreSQL 8.4 6

3.6 OpenLDAP 2.4 7

4 EASY Back-end Services 7

4.1 EASY Fedora Commons Repository 7

4.2 EASY LDAP Directory 10

4.3 EASY Filesystem RDB 12

4.4 EASY PID RDB 12

4.5 EASY SOLR Search Index 13

5 EASY Frond-end Services 15

5.1 EASY Web-UI Application 15

5.2 EASY SWORD-based Ingest Service (Optional) 18

5.3 EASY Proai 18

TODO:

* EASY SWORD-based Ingest Service beschrijven
* Bij elke instructie een laatste stap toevoegen: “Test”
* Diagram bijwerken:
  + Proai toevoegen
* Verschillende Fedora-gebruikers voor verschillende clients (easy-webui, easy-sword, easy-proai)
* Security: de volgende bestanden bevatten wachtwoorden en mogen dus alleen leesbaar zijn voor de owner (tomcat):
  + /opt/easy-webui/cfg/application.properties
  + /opt/fedora/server/config/fedora.fcfg
  + /opt/fedora/server/config/fedora-users.xml

# 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.

## Installation packages

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

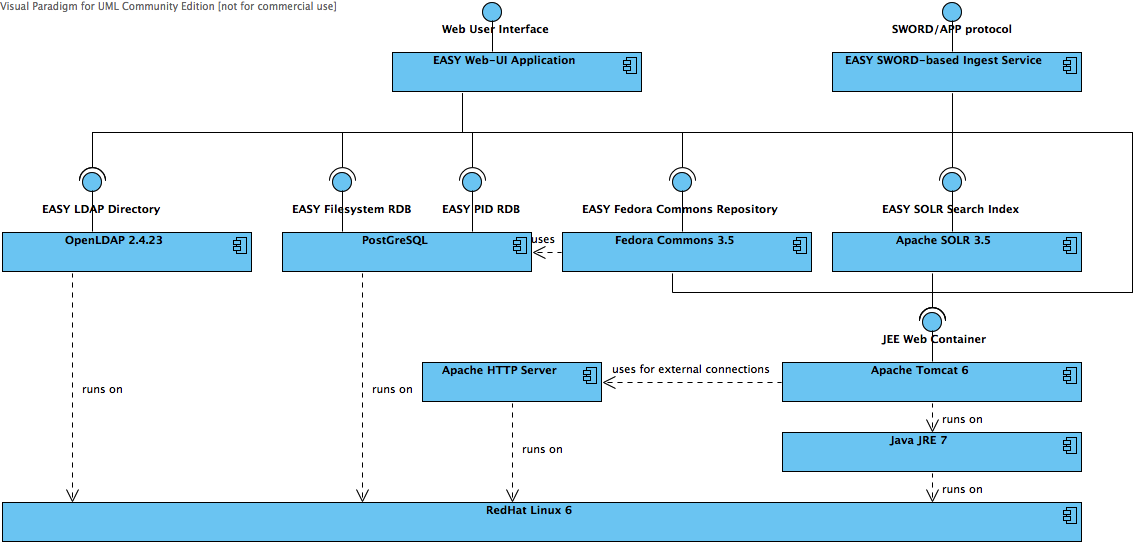
* *easy-backend-2.8.tar.gz* - contains the files for setting up the back-end services needed to run EASY, as well as a copy of this Guide;
* *easy-webui-2.8.tar.gz* - contains the Web-UI Application component;
* *easy-sword-2.8.tar.gz* (optional) - contains the SWORD-based Ingest Service component.

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

$EASY\_BACKEND, $EASY\_WEBUI and $EASY\_SWORD 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,* 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.”

# Overview of EASY

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



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.*

# 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

### 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

### 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/

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

### Add java to alternatives

As the Tomcat installation will automatically install OpenJDK and set it as the default Java installation, we add Oracle Java to alternatives, so we can restore it as the default later.

sudo alternatives --install /usr/bin/java java /usr/java/default/bin/java 1

### Notes

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

## Tomcat 6

### Install Tomcat 6

Execute the following command:

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

### Set java back to Oracle Java

Yum installs OpenJDK and makes it the current Java installation. With alternatives we can put Oracle Java back:

sudo alternatives --config java

When prompted for an alternative to activate choose the one that points to the executable “/usr/java/default/bin/java”.

### 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=”):

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.

### Configure the Tomcat daemon to start automatically

We will configure the Tomcat daemon to start automatically at reboot time:

sudo chkconfig tomcat6 on

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

## Apache HTTP Server 2.2.15

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

## PostGreSQL 8.4

### Install PostGreSQL

Execute the following command:

sudo yum install postgresql-server.x86\_64

### Initialize the database

Initialize the database after installation:

sudo service postgresql initdb

### 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.

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

Start the daemon now:

sudo service postgresql start

## OpenLDAP 2.4

### Install OpenLDAP servers and clients

Execute the following command:

yum install openldap-servers openldap-clients

### 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

Start the daemon now:

sudo service slapd start

# EASY Back-end Services

Now that we have the standard software in place we turn to the set-up and configuration of the back-end services 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

(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:

\password fedora\_db\_admin

and provide a safe password when prompted.

### 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.

### 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

* Fill in the same password for fedora\_db\_admin as you did in step 1 in create-fedora-db.sql.
* Fill in a safe password for fedoraAdmin

Then execute the following command:

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

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

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

sudo mkdir -p /data/fedora/datastreams

sudo mkdir -p /data/fedora/fedora-xacml-policies/repository-policies/default

sudo mkdir -p /data/fedora/resourceIndex

sudo chown -R tomcat:tomcat /data/fedora

Note that the policies directory does need to exist, even though we don’t use 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”

### 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: 5.3 EASY Customized Proai Service) you need to also deploy 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.

Then copy the Tomcat context container to /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

### 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 <fedoraAdminPassword> with the password you filled in after “fedora.admin.pass=” in install.properties.

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

*Look out: the password must be in single quotes, and any dollar signs in it must be escaped with a backslash.*

## 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

### 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

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

### 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

### 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.

Then execute the following command:

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

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 your safe password when prompted to do so. Copy the resulting hash and replace the hash in the file “change-ldapadmin-pw.ldif” (the part in bold):

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

### 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.

Generate a new safe password and calculate its hash with:

slappasswd -h {SSHA}

Edit the file “change-easyadmin-user-pw.ldif” and replace the password hash with the one calculated by slappasswd:

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

Don’t forget that you have to user 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 < create-easy-db.sql

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

### 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

and assign a safe password for each user when prompted. The easy\_admin user is intented to be used for adminstrative 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.

## 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

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

### Assigning passwords

The admin of this database is set to easy\_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 password.

## 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 -xzvf 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

### 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

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

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

sudo cp solrconfig.xml /data/solr/cores/datasets/conf

sudo cp stopwords.txt /data/solr/cores/datasets/conf

sudo cp synonyms.txt /data/solr/cores/datasets/conf

sudo cp 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

### 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

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

# EASY Frond-end Services

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 -xzvf 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:

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

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

sudo service tomcat6 force-reload

### 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-editable-webui-home, 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-home

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

The following is the contents of the file. Fill in the bold, underlined parts as specified. Where the bold, underlined part is in angle brackets, fill in an appropriate value.

#

# EASY Web-UI configuration file

#

# Please, modify the configuration below to match your environment. At the very least,

# you will need to fill in the places marked with # FILL IN # below. Notice the

# additional comments about what values to fill in.

#

# If all supporting services are located on the same host (using default ports)

# fill in the hostname here (e.g. 'localhost' for the same host as easy-webui).

# Otherwise change the instances of ${easy.host} below appropriately.

easy.host=localhost

easy.web.server.port.http=80

easy.web.server.port.https=443

# OpenLDAP user database

ldap.providerURL=ldap://${easy.host}:389

ldap.securityPrincipal=cn=ldapadmin,dc=dans,dc=knaw,dc=nl

ldap.securityCredentials=**<password for ldapadmin, see 4.2.5>**

ldap.context.users=ou=users,ou=easy,dc=dans,dc=knaw,dc=nl

ldap.context.groups=ou=groups,ou=easy,dc=dans,dc=knaw,dc=nl

ldap.context.migration=ou=migration,ou=easy,dc=dans,dc=knaw,dc=nl

ldap.context.federation=ou=federation,ou=easy,dc=dans,dc=knaw,dc=nl

# Federation login support

easy.federationLogin.enabled=false

easy.federationLogin.federationUrl=**<to do, for now: any existing url>**

easy.federationLogin.propertyMapping.shibSessionId=Shib\_Session\_ID

# globally unique ID for the federation account, used to map the federation account to

# an EASY account. This should normally be the standard REMOTE\_USER CGI var.

easy.federationLogin.propertyMapping.remoteUser=Shib\_eduPersonPN

easy.federationLogin.propertyMapping.email=Shib\_email

easy.federationLogin.propertyMapping.firstName=Shib\_givenName

easy.federationLogin.propertyMapping.surname=Shib\_surName

easy.federationLogin.propertyMapping.organization=Shib\_HomeOrg

# Fedora Resource Index

fedora.ri.url=http://${easy.host}:8080/fedora/risearch

# Fedora server

easy.store.name=easy\_local

fedora.base.url=http://${easy.host}:8080/fedora

fedora.admin.username=fedoraAdmin

fedora.admin.userpass=**<fedoraAdmin password, see: 4.1.3>**

# Easy database

fedora.db.username=easy\_webui

fedora.db.password=**<easy\_webui password, see: 4.3.2>**

fedora.db.connectionUrl=jdbc:postgresql://${easy.host}:5432/easy\_db

fedora.db.hbnDriverClass=org.postgresql.Driver

fedora.db.hbnDialect=org.hibernate.dialect.PostgreSQLDialect

# Persistent Identifier database

pid.generator.username=easy\_webui

pid.generator.password=**<easy\_webui password, see: 4.3.2>**

pid.generator.connectionUrl=jdbc:postgresql://${easy.host}/pid\_db

pid.generator.driver=org.postgresql.Driver

# SORL search index

solr.url=http://${easy.host}:8080/solr

# Audit

business.audit.enabled=false

business.audit.file=/var/log/easy-webui/audit-log/audit

# Mail

mail.smtp.host=localhost

mail.fromAddress=**<fill in e-mail address of easyadmin user, see: 4.2.4>**

mail.bccs=

admin.mailer.sendOnStarting=**true**

admin.mailer.sendOnClosing=**true**

admin.mailer.addresses=**<fill in e-mail address of easyadmin user, see: 4.2.4>**

# Limits on MB and number of files to download

data.downloadLimit=1000

file.downloadLimit=100

# Location of the content that is editable through the Web UI. It is HIGHLY

# RECOMMENDED that you copy the example initial content in

# {EASY\_WEBUI\_HOME}/res/example/editable to

# a location outside the installation directory and modify below property to point to

# that direction. Notice also that this new location mus be writeable by the Linux

# user running Tomcat.

easy.editable.content.root=**/opt/easy-webui-editable-home**

#

# Location of the masterkey file containing the SHA-1 hash of the masterkey password.

# If not specified or present, the masterkey is disabled.

#

easy.admin.masterkeyfile=**</your/home/dir/masterkey>**

# Base URL of the deposit instruction files

easy.depositinstructions.baseUrl=

### Configure EASY Web-UI logging

The default logging settings should do for most purposes, however, you do need to create a directory for the log-files.

sudo mkdir -p /var/log/easy-webui/audit-log

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

### Deploy the Tomcat context container

Finally we are ready to deploy the EASY Web-UI application. A default Tomcat context container file is provided in $EASY\_WEBUI\_HOME/bin. If you have unzipped the easy-webui-.tar.gz to the /opt diretory you can just used the default file and copy it to the directory were Tomcat will use it to auto-deploy the war file (which is also in /opt/easy-webui/bin):

sudo cp /opt/easy-webui/bin/ui.xml /etc/tomcat6/Catalina/localhost; \

tail -f /var/log/tomcat6/catalina.out

## EASY SWORD-based Ingest Service (Optional)

<TO DO>

## EASY Customized Proai Service

EASY can function as a 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 -xzvf 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:

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

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

sudo service tomcat6 force-reload

### Create Proai cache and session directories

sudo mkdir -p /data/proai/cache

sudo mkdir -p /data/proai/sessions

sudo mkdir -p /data/proai/schemas

sudo chown -R tomcat:tomcat /data/proai

### Create Proai database

### Configure EASY Proai service settings

Invullen (geen spaties om het password heen!?)

* proai.db.password
* driver.fedora.pass

### Configure EASY Proai logging

### Deploy the Tomcat context container

TODO: waar zetten we dit neer: “static/oai-pmh.xsl” ?

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)