*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 6

4 EASY Back-end Services 7

4.1 EASY Fedora Commons Repository 7

4.2 EASY LDAP Directory 9

4.3 EASY Filesystem RDB 9

4.4 EASY PID RDB 9

4.5 EASY SOLR Search Index 10

5 EASY Frond-end Services 10

5.1 EASY Web-UI Application 10

5.2 EASY SWORD-based Ingest Service (Optional) 10

# 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 RedHat Linux 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:

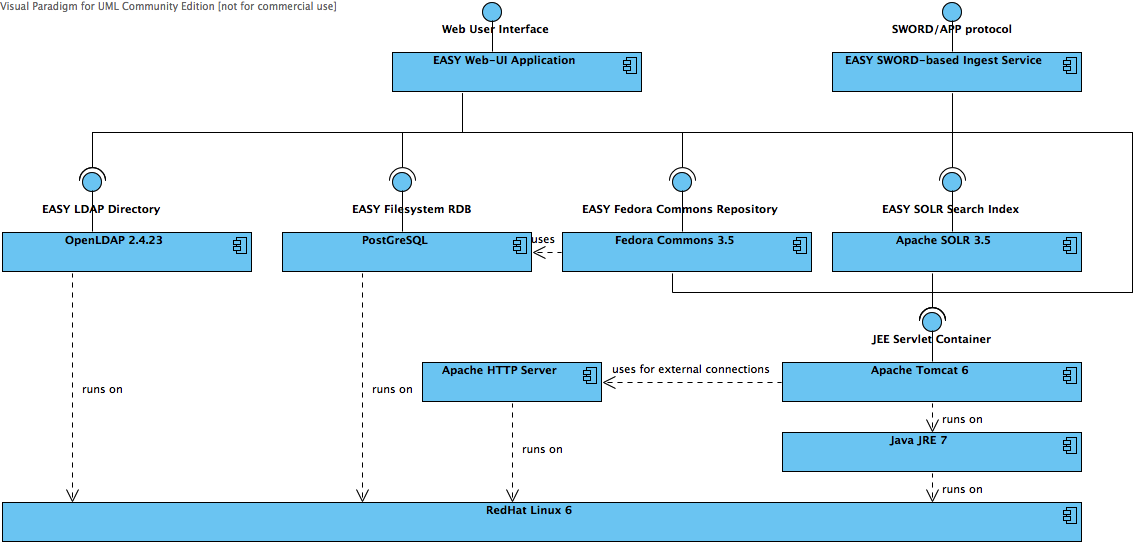
* *easy-backend-2.8.tar.gz* - contains the files for setting up the back-end services needed to run EASY, as well as a copy of this Guide.
* *easy-webui-2.8.tar.gz* - contains the Web-UI Application component
* *easy-sword-2.8.tar.gz* (optional) - contains the SWORD-based Ingest Service component

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 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.

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:

# 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"

### 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.

## 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**

## OpenLDAP 2.4

### Install OpenLDAP servers and clients

Execute the following command:

yum install openldap-servers openldap-clients

# 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

Edit the file

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

and replace the string “replace-with-save-password” with exactly that: a safe password. Consider using a randomly generated password.

On the command line execute the following command:

sudo -u postgres psql -U postgres < create\_fedora\_db.sql

using of course your edited version of “create\_fedora\_db.sql”

(Note: you may get a warning ‘could not change directory to “root”’ but this does not seem to prevent the database from being created.)

### Set the FEDORA\_HOME environment variable

Copy the file $EASY\_BACKEND/easy-fedora-commons-repository/fedora.sh to /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 fedoraDbAdmin 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:

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

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

Edit the file $FEDORA\_HOME/server/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”

### 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/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:

sudo fedora-batch-ingest.sh …

## EASY LDAP Directory

The EASY LDAP Directory component requires some EASY-specific schema’s and a few basic entry’s to work. 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

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

The OpenLDAP 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

We will add the configuration through the LDAP protocol, so we first need a running server:

service slapd start

### 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:

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**

The, 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**

## EASY Filesystem RDB

## EASY PID RDB

## EASY SOLR Search Index

# EASY Frond-end Services

## EASY Web-UI Application

## EASY SWORD-based Ingest Service (Optional)

1. Data Archiving and Networked Services, an institute of the Netherlands Academy for the Arts and Sciences (KNAW) and the Netherlands Research Organisation (NWO). [↑](#footnote-ref-1)
2. See: <https://wiki.duraspace.org/display/FEDORA35/Installation+and+Configuration> [↑](#footnote-ref-2)