



DSpace Installation and Systems Administration Procedures for the Edinburgh Research Archive (ERA)

<http://www.era.lib.ed.ac.uk/>

Version: 1.9

Author: Richard Jones (r.d.jones@ed.ac.uk)

DSpace Version: 1.1.1

Tapir Version: 0.3 (pre-release)

1. Introduction	3
1.1. General Information.....	3
1.2. Notation and Fonts for this Document	3
1.3. Short Hand for this Document	4
1.4 System Users	5
2. Full Installation Overview	6
3. PostgreSQL	8
3.1. Installation from Source.....	8
3.2. Configuration	9
3.3. Starting and Stopping PostgreSQL	9
3.4. Cron Jobs	9
4. DSpace	11
4.1. Installation from Source.....	11
4.2. Configuration File	13
4.3. Cron Jobs	15
5. Tomcat.....	16
5.1. Configuration for DSpace	16
5.2. Tomcat Configuration as Standalone Web-Server	16
5.3. Starting and Stopping Tomcat	18
5.4. Removing the Cache	18
5.5. Configuring SSL	19
6. SSL Certificates	20
6.1. Generating SSL Certificates with OpenSSL	20
6.2. Importing the Signed Certificate	20
6.3. Creating a Self-Signed Certificate with the Java Keytool	21
6.4. Creating a Self-Signed Certificate with OpenSSL	21
7. Handle Server	23
7.1. Configuration	23
7.2. Starting and Stopping the Handle Server	24
8. Localisation	25
8.1. Installing the Custom Design.....	25
8.2. Installing the New Log Reporter	25
8.3. System Wide Name Change	25
8. OAI-PMH	27
8.1. Configuring the OAI Interface	27
9. Web-Server Issues	28
9.1. Apache httpd	28
9.2. IPTables	28
9.3. xinetd.....	29
9.4. mod_jk.....	29
10. Tapir.....	30
10.1. Installation from Source.....	30
11. Data Import	32
11.1. Taking the Data from Source DSpace	32
11.2. Installing the Data in Target DSpace	33

1. Introduction

1.1. General Information

Required Ports

- 80 – HTTP
- 22 – SSH
- 443 – HTTPS
- 8080 – Tomcat HTTP
- 8443 – Tomcat HTTPS
- 2461 – Handle Server native
- 8000 – Handle Server HTTP

Operating System: Red Hat Enterprise System 3

Java Version: 1.4.2_04

Ant Version: 1.5.2

PostgreSQL Version: 7.4.2

DSpace Version: 1.1.1

IP and Server Name

129.215.166.124
agrona.lib.ed.ac.uk

1.2. Notation and Fonts for this Document

Normal Sans-Serif Font - The main body of the text of this document

Italic Sans-Serif Font - Notes that should be paid attention to.

`Fixed-Width Font` - Examples of commands or items that you might see on the computer screen. For example, directory and file names as well as installation commands.

Italic Fixed-Width Font - The user that you should be logged in as when performing tasks. You should always be logged in as the user most recently specified on the left-hand side of the page in the current section.

1.3. Short Hand for this Document

The following short-hand notations are used throughout this document to represent certain types of information. When these are encountered in the main text they should *always* be replaced with the values found in this section.

[postgres] =
The location of the PostgreSQL installation

[dspace] =
The location of the DSpace installation

[dspace-source] = /
The location of the source code used to create the DSpace installation

[dspace-home] =
The `dspace` user's home directory on the system

[database-pw] =
The `dspace` database password for the `dspace` user

[admin-email] =
The email address of the system administrator who will receive all administrative requests and emails from the site.

[admin-fn] =
The First Name of the administrator's account

[admin-ln] =
The Last Name of the administrator's account

[admin-pw] =
The DSpace administrator's login password

[era-url] =
The full URL of the instance of DSpace, including `http://`

[era-host] =
The full URL of the instance of DSpace without the `http://` or any trailing slashes

[handle] =
The Handle Prefix given to the organisation by CNRI

[tomcat] =
The installation directory of the Tomcat web-server

[handle-server] =
The installation directory of the Handle Server

[CNRI Contact] =
Your contact at the CNRI to whom to send support requests

[tapir-source] =
The pre-installation location of the Tapir source code

[machine-ip] =
The IP address of the machine on which your DSpace installation is going

1.4 System Users

Users that will have to exist on the machine onto which DSpace is to be installed are as follows:

- *root* - the super-user for the machine.
- *dspace* - the user who will be the default DSpace user, and will own the tomcat instance etc. This is the most used user.
- *postgres* - the user who will own the database software

The user you need to be logged in as when performing the actions laid out in this document will be given on the left hand side of the page before the set of actions which this user needs to perform.

2. Full Installation Overview

This section details the order in which things ought to be done in order to have a fully working DSpace with Tapir in the form of the Edinburgh Research Archive.

1. Before starting the installation ensure that the machine has the prerequisite software for DSpace installed:
 - a. Red Hat Enterprise Server 3.0
 - b. Tomcat 4.0.30
 - c. Java 1.4.2_04
 - d. Ant 1.5.2

The installation of this software is outside the scope of this document.

2. Install PostgreSQL as per the section **PostgreSQL: Installation from Source**. Once this has been done you should configure PostgreSQL as per the section **PostgreSQL: PostgreSQL Configuration** before starting the database server as per the section **PostgreSQL: Starting and Stopping PostgreSQL**.
3. Set up the PostgreSQL cron job as per the section **PostgreSQL: Cron Jobs**.
4. Install DSpace as per the section **DSpace: Installation from Source**.
5. Generate and install a Self-Signed SSL Certificate with OpenSSL as per the section **SSL Certificates: Creating a Self-Signed Certificate with OpenSSL**.
6. Configure Tomcat to use the SSL Certificate as per the section **Tomcat: Configuring SSL**
7. Configure the pre-installed handle server as per the section **Handle-Server: Configuration** then start it as per the section **Handle Server: Starting and Stopping**.
8. Make the localisation updates to integrate the system with local design decisions as per the section **Localisation: Installing the Custom Design**. Update the default JSPs to have the new system name as per the section **Localisation: System Wide Name Change**
9. To replace a broken perl script to do log analysis follow the section **Localisation: Installing the New Log Reporter**.

10. Test that the new OAI interface is working correctly as per the section **OAI-PMH: Configuring the OAI Interface**.
11. Install Tapir as per the section **Tapir: Installation from Source**.
12. Import the data from another instance of DSpace as per the section **Data Import**.

3. PostgreSQL

3.1. Installation from Source

1. Download postgresql-7.4.2.tar.gz from www.postgres.org

postgres

2. Insert postgresql-7.4.2.tar.gz into [postgres]

3. Unzip and Untar postgresql-7.4.2.tar.gz

```
gunzip postgresql-7.4.2.tar.gz
tar -xvf postgresql.7.4.2.tar
```

4. Configure PostgreSQL for installation:

```
In [postgres]/postgres-7.4.2:
```

```
./configure --prefix=[postgres]
```

5. Build the PostgreSQL source code. In [postgres]/postgresql-7.4.2:

```
gmake
```

6. Check that this build has worked

```
gmake check
```

7. Install PostgreSQL

```
gmake install
```

8. Add PostgreSQL to the PATH variable for users *postgres* and *dspace*. In *.bash_profile* in each user's home directory, add the line

```
PATH = PATH:[postgres]/bin
```

9. Initialise the database

```
[postgres]/bin/initdb -D [postgres]/data
```

10. Download the PostgreSQL Java drivers from:
<http://jdbc.postgresql.org/download.html>. The required file is *pg74.213.jdbc3.jar*, and should be placed in *[dspace]/lib*

3.2. Configuration

The following settings need to be entered in the file
`[postgres]/data/postgresql.conf`:

1. `tcpip_socket = true`
To allow PostgreSQL to be accessed via TCP/IP
2. `max_connections = 400`
To allow 400 connections to be opened to the database
3. `shared_buffers = 3000`
To provide normal usage in a live environment

Note that if these values have to go any larger then the value of the system variable `SHMMAX` which limits segment size must be increased.

3.3. Starting and Stopping PostgreSQL

postgres

- Start the database server:
`[postgres]/bin/postmaster -D [postgres]/data`
- Stop the database server:
`pg_ctl stop -m fast -D [postgres]/data`
or (only if above fails):
`kill `cat [postgres]/data/postmaster.pid``

3.4. Cron Jobs

dspace

1. Check that the file `mycron` exists in the directory `[dspace-home]` to contain all the cron information for the `dspace` user. If not, create it.
2. Add the following lines into the `mycron` file:
`#vacuum full analyze the database every night`

```
0 2 * * * [postgres]/bin/vacuumdb -f -z
```

3. Export the new cron job to the dspace user's crontab:

```
crontab mycron
```

4. DSpace

4.1. Installation from Source

1. Download the DSpace source from <http://sourceforge.net/projects/dspace>

root

2. Create the DSpace user and group (`dspace` and `dspace` respectively).

dspace

3. Create source directory: `[dspace-source]`, and insert the uncompressed source from <http://sourceforge.net/projects/dspace>
4. Create the target installation directory `[dspace]`
5. Place the additional required libraries into `[dspace-source]/lib`
 - o `activation.jar`, obtained from <http://java.sun.com/products/javabeans/glasgow/jaf.html>
 - o `mail.jar`, obtained from <http://java.sun.com/products/javamail>
 - o `servlet.jar`, obtained from <http://java.sun.com/products/jsp/download.html>
 - o `pg74.213.jdbc3.jar`, obtained from <http://jdbc.postgresql.org/download.html>
6. Create a `dspace` database user for PostgreSQL (the database server must be running in order for these commands to work). In the directory `[postgres]/bin`

```
./createuser -U postgres -d -A -P dspace
```

Password: `[database-pw]`

7. Create the database to be used by `dspace`

```
./createdb -U dspace dspace
```

8. Prepare the configuration files for build in `[dspace-source]/config`. See the section **DSpace: Configuration File** for more details.
9. Compile the code. In `[dspace-source]`

```
ant
```

10. Install the code

```
ant fresh_install
```

Note: If you wish to perform this step manually, then in [dspace-source]/bin:

```
./dsrun org.dspace.storage.rdbms.InitializeDatabase  
/u01/dspace-home/dspace-1.1.1  
/etc/database_schema.sql
```

```
./dsrun org.dspace.administer.RegistryLoader  
-bitstream  
/u01/dspace/config/registries  
/bitstream-formats.xml
```

```
./dsrun org.dspace.administer.RegistryLoader -dc  
/u01/dspace/config/registries  
/dublin-core-types.xml
```

11. Prepare the Tomcat configuration as per the section **Tomcat: Configuration for DSpace**

12. Install the config files into the dspace install. In [dspace]/bin

```
./install-configs
```

13. Create the dspace system administrator. In [dspace]/bin

```
./create-administrator
```

Using the values

```
email: [admin-email]  
First Name: [admin-fn]  
Last Name: [admin-ln]  
Password: [admin-pw]
```

14. Index the (empty) contents of the database. In [dspace]/bin

```
./index-all
```

15. Set up Tomcat as the web-server as per the section **Tomcat: Configuration as Standalone Web-Server**

16. Start Tomcat as per the section **Tomcat: Starting and Stopping Tomcat**

17. Set up the cron jobs associated with DSpace as per the section **DSpace: Cron Jobs**

4.2. Configuration File

The following settings are included in the DSpace config file `dspace.cfg` in `[dspace]/config` (or `[dspace-source]/config` prior to install). It is important to ensure that there are no extraneous spaces after any of the properties.

```
# DSpace installation directory
dspace.dir = [dspace]

# DSpace base URL
dspace.url = [era-url]

# DSpace host name - should match base URL
dspace.hostname = [era-host]

# Name of the site
dspace.name = Edinburgh Research Archive

config.template.log4j.properties =
    [dspace]/config/log4j.properties
config.template.log4j-handle-plugin.properties =
    [dspace]/config/log4j-handle-plugin.properties
config.template.oaicat.properties =
    [dspace]/config/oaicat.properties
config.template.oai-web.xml =
    [dspace]/oai/WEB-INF/web.xml

##### Database settings #####

# URL for connecting to database
db.url = jdbc:postgresql://localhost:5432/dspace

# JDBC Driver
db.driver = org.postgresql.Driver

# Database username and password
db.username = dspace
db.password = [database-pw]

##### Email settings #####

# SMTP mail server
mail.server=mailrelay.ed.ac.uk

# From address for mail
mail.from.address = [admin-email]

# Currently limited to one recipient!
feedback.recipient = [admin-email]
```

```

# General site administration (Webmaster) e-mail
mail.admin = [admin-email]

# Recipient for server errors and alerts
alert.recipient = [admin-email]

#### File Storage #####

# Asset (bitstream) store number 0 (zero)
assetstore.dir = [dspace]/assetstore

# Specify extra asset stores like this, counting from 1 upwards:
# assetstore.dir.1 = /second/assetstore
# assetstore.dir.2 = /third/assetstore

# Specify the number of the store to use for new bitstreams with this
# property. The default is 0 (zero) which corresponds to the
# 'assetstore.dir' above
# assetstore.incoming = 1

# Directory for history serializations
history.dir = [dspace]/history

# Where to put search index files
search.dir = [dspace]/search

# Where to put the logs
log.dir = [dspace]/log

# Where to temporarily store uploaded files
upload.temp.dir = /tmp

# Maximum size of uploaded files in bytes, must be positive
# 512Mb
upload.max = 536870912

##### Handle settings #####

# CNRI Handle prefix
handle.prefix = [handle]

# Directory for installing Handle server files
handle.dir = [handle-server]

##### Web UI Settings #####

# The site authenticator - must implement
# org.dspace.app.webui.SiteAuthenticator
webui.site.authenticator = org.dspace.app.webui.SimpleAuthenticator

# Certificate authority
webui.cert.ca = [dspace]/etc/certificate-ca.pem

# If a user presents a valid Web certificate, but does not have an
# e-person record, should they automatically be given a new e-person
# record?
webui.cert.autoregister = true

```

```

# Should the submit UI block submissions marked as theses?
webui.submit.blocktheses = false

##### SFX Server #####

# SFX query is appended to this URL.  If this property is commented
# out or omitted, SFX support is switched off.
# sfx.server.url = http://sfx.myu.edu:8888/sfx?

##### Ingest settings #####

# Default language for content of submissions
default.language = en

```

4.3. Cron Jobs

dspace

1. Check that the file `mycron` exists in the directory `[dspace-home]` to contain all the cron information for the `dspace` user. If not, create it.

2. Add the following lines into the `mycron` file:

```

#send out subscription emails at 01.00 every day
0 1 * * * [dspace]/bin/sub-daily

```

3. Export the new cron job to the `dspace` user's crontab:

```

crontab mycron

```

4. To ensure that the crontab has access to the Java files that it needs to run we need to add some additional code to the file being run. In the file `[dspace]/bin/sub-daily` add the line (near the top):

```

export PATH=$PATH:/usr/java/j2sdk1.4.2_04/bin

```

5. Tomcat

5.1. Configuration for DSpace

dspace

1. Set up the OAI service symlink. In `[tomcat]/webapps`

```
ln -s [dspace]/oai dspace-oai
```

2. Backup the original `ROOT` web-app then replace with one for `dspace`

```
ln -s [dspace]/jsp ROOT
```

5.2. Tomcat Configuration as Standalone Web-Server

dspace

1. Disable the Tomcat Admin interface as this interferes with the DSpace administration system. In `[tomcat]/webapps` rename the file `admin.xml` to `admin_xml.bak`.

2. Edit the `server.xml` file in `[tomcat]/conf`. Under where it reads:

```
<!-- Define the Tomcat Stand-Alone Service -->
```

- a. Set up the port connectors as follows:

Under:

```
<!-- Define a non-SSL Coyote HTTP/1.1 Connector on port xx -->
```

ensure the next line reads (the port number is the important bit):

```
<Connector  
className="org.apache.coyote.tomcat4.CoyoteConnector"  
port="8080" ...
```

Then under:

```
<!-- Define a SSL Coyote HTTP/1.1 Connector on  
port xx -->
```

ensure the next line reads (the port number is the important bit), and has also been uncommented:

```
<Connector
className="org.apache.coyote.tomcat4.CoyoteConnector"
port="8443" ...
```

- b. Set up the contexts for DSpace and the DSpace OAI interfaces.

At the top of the Tomcat connector examples section, inside the virtualhost configuration for the Tomcat Standalone service, enter the following (the first section should replace the current Tomcat ROOT context):

```
<!-- Tomcat Root Context - DSpace -->
<Context path="" docBase="ROOT" debug="0"
  reloadable="true" crossContext="true">
  <Resources
    className=
      "org.apache.naming.resources.FileDirContext"
    allowLinking="true" />
  </Context>

<!-- DSpace OAI Interface -->
<Context path="/dspace-oai" docBase="dspace-oai"
  debug="0" reloadable="true" crossContext="true">
  <Resources
    className=
      "org.apache.naming.resources.FileDirContext"
    allowLinking="true" />
  </Context>
```

root

3. Since HTTP requests come in on port 80 by default we need to route all traffic for that port to port 8080 where Tomcat is listening. We cannot run tomcat on port 80 as it will only run as *root*, which is a security risk. We therefore use IPTables to perform port pre-routing to overcome this problem. See the section **Web-Server Issues** for more information.

To set up IP tables to route 80 to 8080 and 443 to 8443:

```
iptables -t nat -A PREROUTING -p tcp -i eth0 -d
[machine-ip] --dport 80 -j DNAT --to
[machine-ip]:8080
```

```
iptables -A FORWARD -p tcp -i eth0 -d
[machine-ip] --dport 8080 -j ACCEPT
```

```
iptables -t nat -A PREROUTING -p tcp -i eth0 -d
[machine-ip] --dport 443 -j DNAT --to
[machine-ip]:8443
```

```
iptables -A FORWARD -p tcp -i eth0 -d
[machine-ip] --dport 8443 -j ACCEPT
```

The web server will now route communication from ports 80 and 443 to 8080 and 8443 respectively.

5.3. Starting and Stopping Tomcat

dspace

Before starting or stopping tomcat for the first time, create shell scripts to do the job for you, as this will be much quicker.

1. In the directory `[dspace-home]/bin` create a file called `starttomcat` containing the line:

```
[tomcat]/bin/startup.sh
```

2. In the directory `[dspace-home]/bin` create a file called `stoptomcat` containing the line:

```
[tomcat]/bin/shutdown.sh
```

3. In the directory `[dspace-home]/bin` you need to make the files `starttomcat` and `stoptomcat` executable. To do this use:

```
chmod 764 starttomcat stoptomcat
```

4. Place this directory in the `dspace` user's `PATH` environment variable so that they can be run from anywhere on the system. In the file `[dspace-home]/.bash_profile` add the line:

```
PATH=$PATH:$HOME/bin
```

You can now start and stop tomcat from anywhere on the system by issuing one of the commands:

```
starttomcat
```

```
stoptomcat
```

Note that it is worth restarting tomcat after every system change as otherwise updates may not be reflected.

5.4. Removing the Cache

To force Tomcat to update changed files it can be necessary to remove its cached files. We create a shell script to do this job for us, as this will be much quicker.

dspace

1. In the directory `[dspace-home]/bin` create a file called `removecache` containing the line:

```
rm -r [tomcat]/work/Standalone/localhost/_
```

2. We need to make the file `removecache` executable, so use the command:

```
chmod 764 removecache
```

5.5. Configuring SSL

dspace

1. Generate the self-signed certificate for tomcat as per the section **SSL: Certificates: Creating a Self-Signed Certificate with OpenSSL**
2. Set up an SSL connector for Tomcat.

In `[tomcat]/conf/server.xml` set up an HTTP connector on port 8443:

```
<!-- Define a SSL Coyote HTTP/1.1 Connector on port 8443 -->
<Connector className="org.apache.coyote.tomcat4.CoyoteConnector"
    port="8443" minProcessors="5" maxProcessors="75"
    enableLookups="true" acceptCount="100" debug="0"
    scheme="https" secure="true"
    useURIVValidationHack="false"
    disableUploadTimeout="true">
    <Factory className=
        "org.apache.coyote.tomcat4.CoyoteServerSocketFactory"
        clientAuth="false" protocol="TLS"
        keystoreFile="[dspace-home]/keystore.tomcat"
        keystorePass="changeit" />
</Connector>
```

6. SSL Certificates

6.1. Generating SSL Certificates with OpenSSL

dspace

1. Generate the key in [dspace-home]/certificates:

```
openssl genrsa -out [era-host] 1024
```

2. Generate the certificate request:

```
openssl req -new -key [era-host].key -out  
[era-host].csr
```

Using the following values:

```
Country Name: GB  
State or Province Name: Mid-Lothian  
Locality Name: Edinburgh  
Organisation Name: Edinburgh University Library  
Organisational Unit Name: Library Systems  
Common Name: Edinburgh Research Archive  
Email Address: [admin-email]  
Password: <none>  
Company: <none>
```

3. Paste the contents of the file [era-host].csr into the form on the following web-page:

http://www.ucs.ed.ac.uk/fmd/unix/docs/certificates/sign_cert.html

6.2. Importing the Signed Certificate

dspace

1. Import the .pem file using the command

```
keytool -import -alias dspace -file  
[dspace-home]/certificates/dspace.pem
```

When a password is requested use: `changeit`

The resulting keystore file is located here:

```
[dspace-home]/.keystore
```

6.3. Creating a Self-Signed Certificate with the Java Keytool

dspace

1. Generate the certificate:

```
keytool -genkey -alias tomcat -keyalg RSA
```

2. Self-sign the certificate:

```
keytool -selfcert -alias tomcat
```

3. If you wish to sign the certificate via a certificate authority at a later date, a certificate for signing can be exported thus:

```
keytool -certreq -keyalg RSA -alias tomcat -file  
    <place to put certificate request file>
```

6.4. Creating a Self-Signed Certificate with OpenSSL

This is the method that has been used to generate the certificate that is currently in use in ERA.

dspace

1. Generate the key:

```
openssl genrsa -des3 -out pass.key 1024
```

2. Generate the server key:

```
openssl rsa -in pass.key -out server.key
```

3. Sign the certificate yourself (valid for 999 days):

```
openssl req -new -key server.key -x509 -out  
    server.crt -days 999
```

4. Generate the DER key file:

```
openssl pkcs8 -topk8 -nocrypt -in server.key -out  
    server.key.der -outform der
```

5. Generate the DER certificate file:

```
openssl x509 -in server.crt -out server.crt.der  
    -outform der
```

6. Use the ImportKey utility in Java to import the key into the keystore:

```
java -cp importkey.jar comu.ImportKey  
server.key.der server.crt
```

7. Handle Server

7.1. Configuration

dspace

1. Make the handle server with the DSpace native commands. In [dspace]/bin:

```
./make-handle-config
```

Note: This may fail with an error like "Warning: data not encrypted". In this case it is necessary to do the installation directly:

```
./dsrun net.handle.server.SimpleSetup  
[handle-server]
```

2. The following are the answers to the questions that should be given during handle-server setup:

```
Caching or Regular: 1  
Primary Server: y  
IP Address: [era-host]  
Port Number: 2641  
HTTP Port: 8000  
Log all access: y  
Version/Serial No: 1  
Description: Edinburgh Research Archive  
Disable UDP: y  
Server Key: n  
Admin Key: n
```

3. This generates a number of files, including a setup file that needs to be sent to CNRI. Find the file:

```
[handle-server]/sitebndl.zip
```

and email to CNRI:

```
[cnri-contact]
```

quoting the identifier number registered for ERA: **[handle]**

4. While waiting for the site bundle to be applied we need to perform the local handle server configuration. In the file [handle-server]/config.dct

Find the section `server_config` and do the following updates:

```
"storage_type" = "CUSTOM"  
"storage_class" = "org.dspace.handle.HandlePlugin"
```

In addition, replace all instances of `YOUR_NAMING_AUTHORITY` with `[handle]` for the whole file.

5. Now start the handle server as per the section **Handle Server: Starting and Stopping the Handle Server**
6. Note that the handle server caches previous settings for a while, so it is necessary to wait for 24 hours or so before changes noticeably take effect.

7.2. Starting and Stopping the Handle Server

dspace

To start the handle server, in `[dspace]/bin` use the command:

```
./start-handle-server
```

To stop the handle server it is necessary to use the unix command `kill`. Get the process ID for the line starting:

```
/bin/sh /u01/dspace/bin/dsrun -Dlog4j.configuration  
=log4j-handle-plugin.properties  
net.handle.server.Main ...
```

and use:

```
kill <process id>
```

8. Localisation

8.1. Installing the Custom Design

dspace

1. Copy all of the design JSPs into `[dspace]/jsp/local`
2. Copy the custom image directory `eul-image` into `[dspace]/jsp`
3. Replace the file `[dspace]/jsp/styles.css.jsp` with the custom version of the same name.
4. Remove the file `[dspace]/jsp/local/styles.css.jsp`
5. Replace the files in `[dspace]/config/emails` with the custom version of each file.
6. Replace the file `[dspace]/jsp/favicon.ico` with the custom version of this file
7. Replace the files in `[dspace]/jsp/image/submit` with the custom version of each file
8. Replace the file `[dspace]/config/default.licence` with the custom version of this file.

8.2. Installing the New Log Reporter

dspace

1. Insert the following line near the end of the file `[dspace-home]/.bash_profile`:

```
export LC_ALL=C
```
2. Replace the file `[dspace]/bin/log-reporter` with the custom version of this file
3. Ensure the permission to run this file are set by running:

```
chmod 764 log-reporter
```

8.3. System Wide Name Change

To replace all instances of "Dspace" with instances of "ERA" we perform a recursive find and replace on the source code.

dspace

1. In [dspace]/jsp:

```
grep -l " Dspace" `find | grep jsp$` | xargs perl  
-pi -e 's/ Dspace/ ERA/g'
```

```
grep -l "&nbsp;Dspace" `find | grep jsp$` | xargs  
perl -pi -e 's/&nbsp;Dspace/&nbsp;ERA/g'
```

All instances of Dspace and Dspace have now been replaced with ERA and ERA.

2. To reverse this process use the following commands in [dspace]/jsp:

```
grep -l " ERA" `find | grep jsp$` | xargs perl  
-pi -e 's/ ERA/ Dspace/g'
```

```
grep -l "&nbsp;ERA" `find | grep jsp$` | xargs  
perl -pi -e 's/&nbsp;ERA/&nbsp;Dspace/g'
```

All instances of Dspace and ERA have now been replaced with ERA and Dspace.

8. OAI-PMH

The ERA OAI-PMH interface is available at the address:

<http://www.era.lib.ed.ac.uk/dspace-oai>

8.1. Configuring the OAI Interface

Almost no configuration should be required to get the OAI interface working. A couple of tests should suffice to ensure that things are behaving as they should:

`http://www.era.lib.ed.ac.uk/dspace-oai?verb=Identify`

`http://www.era.lib.ed.ac.uk/dspace-oai?verb=ListRecords&from=<insert start date>&until=<insert end date>&metadataPrefix=oai_dc`

If the results of these queries come back looking OK then the OAI interface is working correctly.

9. Web-Server Issues

Since Tomcat can only be started as root on any privileged port and does not come with the facility to fork to another user it is necessary to find a solution that allows Tomcat to run on port 8080 while all communications to port 80 (the default HTTP port) get routed to port 8080 without causing any security problems; likewise for 443 and 8443. Possible solutions to this problem are as follows:

1. Use Apache *httpd* on ports 80 and 443 to redirect requests to ports 8080 and 8443
2. Use *IPTables* to do port pre-forwarding from 80 to 8080 and 443 to 8443
3. Use *xinetd* to start Tomcat bound to port 80 and 443 but not running as root.
4. Use *mod_jk* to connect Apache *httpd* to Tomcat.

9.1. Apache *httpd*

Pros:

- Known to be secure
- Well supported within the university

Cons:

- `:8080` inserted in URL at redirect, which does not look good, and may cause problems with page bookmarking and redirects.
- Doesn't work with IE in XP (unknown reason)
- Doesn't work for OAI Interface (redirect issue)

9.2. *IPTables*

Pros:

- Port forwarding happens invisibly, so changes in setup do not affect the user
- Minimal overhead as packet handling goes on in the kernel anyway

Cons:

- Not the officially supported solution supported by the university

9.3. xinetd

Pros:

- Allows Tomcat to run on port 80

Cons:

- Large overhead as xinetd forks at each request for the application.

9.4. mod_jk

Pros:

- Allows for use of Apache httpd (see section **Web Server Issues: Apache httpd**)

Cons:

- May be difficult/impossible to configure to provide the behaviour we desire.

10. Tapir

Tapir is the Edinburgh University Library's customised add-on to the DSpace system to provide E-Theses functionalit.

10.1. Installation from Source

For ERA v1.0 we install the Tapir version that lies between 0.2.1 and 0.3 and is not an official release.

dspace

1. Upload the Tapir source to the directory [tapir-source]
2. In the directory [tapir-source]/java run

```
ant -Ddslib[dspace]/lib install
```

This generates two warnings that can be ignored.
3. In the directory [tapir-source]/java run

```
ant -Dconfig[dspace]/config/dspace.cfg database
```

Note: to do this manually, in [dspace]/bin:

```
./dsrun org.dspace.rdbms.InitializeDatabase  
[tapir-source]/java/etc/workspace.sql
```

```
./dsrun org.dspace.rdbms.InitializeDatabase  
[tapir-source]/java/etc/submissions.sql
```
4. Insert the contents of [tapir-source]/jsp/WEB-INF/web.xml into [dspace]/jsp/WEB-INF/web.xml. **Note that all the sections of this file must be in the correct order or Tomcat will not start; it is therefore necessary to manually move each of the relevant parts in the Tapir file to the relevant part in the DSpace file.**
5. Insert the contents of [tapir-source]/jsp/WEB-INF/dspace-tags/tld into [dspace]/jsp/WEB-INF/dspace-tags.tld
6. At this stage there exists one local JSP which has been customised in the section **Localisation** and is also present in the Tapir source. The file [tapir-source]/jsp/layout/navbar-admin.jsp must be merged with the file [dspace]/jsp/local/layout/navbar-admin.jsp
7. Copy all of the files in [tapir-source]/jsp/ into [dspace]/jsp/local (except navbar-admin.jsp):

```
cp -r [tapir-source]/jsp/* [dspace]/jsp/local
```

8. Remove the directory [dspace]/jsp/local/WEB-INF
9. Update the DSpace config file to contain additional Tapir information. In [dspace]/config/dspace.cfg insert the lines of text contained in the file [tapir-source]/config/dspace.cfg:

```
cat [tapir-source]/config/dspace.cfg >>  
[dspace]/config/dspace.cfg
```

10. Insert all of the new licences into [dspace]/config (in [tapir-source]/config):

```
cp default.licence licence.* [dspace]/config/
```

11. Add the Tapir specific styles in [dspace]/jsp/local/styles.css.jsp to the base style sheet for DSpace at [dspace]/jsp/styles.css.jsp

12. Remove the file [dspace]/jsp/local/styles.css.jsp

13. Ensure the copies of all JSPs provided by Tapir that are new files to the system and not updated versions of DSpace native files are duplicated in [dspace]/jsp as well as appearing in the local directory. The directory structuring of these files should also be reflected.

To see a list of the new files look at the file [tapir-source]/docs/FILE_LIST.txt.

11. Data Import

This section describes how we move data from one instance of DSpace on one machine to another instance on another machine. This process should be adaptable to most import or upgrade tasks.

This procedure assumes similar structure on each machine, and notation is relative to the machine in question.

11.1. Taking the Data from Source DSpace

dspace

1. Stop tomcat.
2. Shut down Tomcat to prevent any further changes to the data or the database.
3. In `[dspace]` bundle up the `assetstore` for moving:

```
tar -cvf assetstore.tar assetstore/  
  
gzip -v9 assetstore.tar
```

This produces a file called `assetstore.tar.gz` which contains all of the archived files. This file can be very large (e.g. 1.7 Gb for ~ 380 items)

postgres

4. Back up the contents and structure of the database. In `[dspace]` run:

```
pg_dump dspace > era-data
```

This produces a file called `era-data` which contains all of the database. This file is generally of reasonable size (e.g. 3.2Mb for ~ 380 items).

dspace

5. Compress the `era-data` file for convenience:

```
gzip -v9 era-data
```

This generates a file called `era-data.gz` which could be as little as 30% the size of the original `era-data` file.

6. Backup `assetstore.tar.gz` and `era-data.gz` to another machine (ideally not the source or the target machines).
7. You can now restart Tomcat. Changes made after this will not be reflected in the target machine, and this procedure must be performed again for new updates.

11.2. Installing the Data in Target DSpace

dspace

1. Stop Tomcat.
2. Upload the files `assetstore.tar.gz` and `era-data.gz` created in the previous section to the directory `[dspace-home]/era-data/`

3. In `[dspace]` rename the old assetstore:

```
mv assetstore assetstore_old
```

4. Copy `[dspace-home]/era-data/assetstore.tar.gz` to `[dspace]`

5. Unzip and untar the new assetstore:

```
gunzip asseststore.tar.gz
```

```
tar -xvf assetstore.tar
```

This will create a new directory called `assetstore` which is an exact copy of the one in the source DSpace installation.

postgres

6. Backup the old data in the database. In `[dspace]` run:

```
pg_dump dspace > era-data-old
```

This generates a file called `era-data-old` which contains the old database contents.

7. Prepare to drop the old `dspace` database by stopping and starting `postgres` to kill all latent connections to the database (still logged in as `postgres`):

```
pg_ctl stop -m fast -D [postgres]/data
```

```
postmaster -i -D [postgres]/data
```

8. Drop the old DSpace database, and (still logged in as `postgres`):

```
dropdb dspace
```

dspace

9. Now log in as `dspace` and reinitialise the DSpace database as you would do in a normal installation. In `[postgres]/bin` run:

```
./createdb -U dspace dspace
```

postgres

10. Prepare to import the data into the DSpace database. Copy `[dspace-home]/era-data/era-data.gz` to `[postgres]`

11. Unzip the `era-data.gz` file:

```
gunzip era-data.gz
```

12. Import the data into the DSpace database:

```
psql dspace < era-data
```

This will create the database as an exact replica of the source installation's database.

13. You can now restart Tomcat.