E-Soter Web Portal Cookbook

D.Simms, H.I. Reuter, S. Hallett, P.Tempel, Y.Yigini, D. Leibovici, A. Pourabdollah

March, 2013
Executive Summary

This document presents a ‘Cookbook’ description for installing the software tools necessary to develop and configure an e-SOTER web portal. The portal developed may provide the basis for project, national or regional datasets held in the SOTER format. The work presented here is a direct output and deliverable of the EU Framework 7 project ‘e-SOTER – Regional Pilot Platform as EU Contribution to a Global Soil Observing System’, Grant Agreement Number 211578 and was conducted under Work Package 6 of the project.

The steps below describe the process of preparing a server to host the tools necessary to operate and run the e-SOTER web Portal and associated components. Initial steps are required to install each of the essential elements, followed by more detailed configuration of the setup and environment.

Example datasets are provided in SoTerML format to allow the new e-SOTER Web Portal to be set up to serve SOTER data across the Internet.
Contents
Executive Summary
Table of Figures
Glossary
Overview
SOTER in Context
Platform
  Accessing the server
Initial Installation
  Java
  Apache
  Tomcat
  Postgres / PostGIS
  Python
  Perl
  Wrap-up
Configuration of Installed Software
  Postgres Configuration – Preparing for the database
  Postgres Configuration - Communications
Installing GeoServer
  Preparing the server for GeoServer
  Obtaining and installing GeoServer
  Obtaining and installing GDAL
Configuring the Apache Web Server
  Changing the Apache redirects for Proxy Handling
  Restarting Apache and Tomcat servers
  Confirming the operation of GeoServer
Uploading Sample Data to the Portal
  Using ftp to upload the sample data
    Installing FTP clients
    Installing an FTP Server
    Copying over the Sample SoTerML data
Creation of the e-SOTER Database
  Using PGAdmin – the graphical database administration tool
Configuring GeoServer for Simple Objects
   Installing the SoTerML processing tools
      Editing Loadref.Java and Loadsample.Java
      Compiling Loadref.Java and Loadsample.Java
   Configuring GeoServer for a Simple Objects web Service
      Testing the Server WMS Capabilities
Configuring GeoServer for Complex Objects
   Installing the GeoServer Application schema extension
   Configuring the Geoplan workspace folder
      Copying the mapping files for workspaces
      Editing the mapping files for workspaces
      Copying the mapping files for schemas
      Editing the mapping files for schemas
   Testing the Installation
      Testing the Server WFS Capabilities
Debugging
   Installing Geo-network
   Installing Algorithm Database
Installation and Configuration of the e-Soter Web Services Portal
Appendix One: Digital Media
   Filesset One: Database files
   Filesset Two: Client Data files
   Filesset Three: GeoServer files
Appendix Two: SoTerML Parser
   eSoterParser utility – Manual and Instructions
      Introduction
      Prerequisites
      Instructions on use
      Data outputs
      Data validation
      Code documentation
      Source Code
      Further Queries
      Digital attachment
Table of Figures

Figure 1 Overview of the installation and configuration process for the e-SOTER web portal
Figure 2 Overview architecture of the e-SOTER web portal
Figure 3 Running nmap on the server
Figure 4 The Tomcat installation success webpage
Figure 5 The GeoServer home webpage
Figure 6 Logging in to GeoServer
Figure 7 Changing the GeoServer admin account password
Figure 8 GeoServer Sample Data mapping, sf:streams
Figure 9 Connecting PGAdmin to the server
Figure 10 The PGAdmin Console
Figure 11 Selecting the SQL editor in PGAdmin
Figure 12 Using the SQL editor in PGAdmin
Figure 13 Executing queries using the SQL editor in PGAdmin
Figure 14 Refreshing the table view in PGAdmin
Figure 15 Deleting tables in PGAdmin
Figure 16 Loading SQL code to the SQL editor in PGAdmin
Figure 17 The Function view in PGAdmin
Figure 18 Adding a new data store in GeoServer
Figure 19 Configuring a new data store in GeoServer
Figure 20 Viewing the test layer in OpenLayers
Figure 21 WFS capabilities of the server
Figure 22 Administrator panel for GeoNetwork
Figure 23 Adding a user groups
Figure 24 Adding Templates
Figure 25 Login bar for accessing GeoNetwork administrator interface
Figure 26 Main Page of Administration panel
Figure 27 Metadata Creation Options
Figure 28 Metadata Creation Page User Group Selection
Figure 29 Metadata Creation Screen from a template
Figure 30 Metadata insert - Import metadata record in XML or MEF format
Figure 31 Metadata in GeoNetwork
Figure 32 Metadata record of the E-Soter Central European Window
Figure 33 TRAC Home Screen
Figure 34 New Ticket Screen in TRAC
Figure 35 Main Configuration page during the Joomla installation
Figure 36 Database configuration page
Figure 37 Finalisation page to the Joomla installation
Figure 38 Installation overview during the Joomla installation
Figure 39 Success page of the Joomla installation
Figure 40 Starting screen of the eSoterParser utility
Figure 41 Selecting an Access database
Figure 42 Status page for database selection
Figure 43 Selecting a Shapefile database
Figure 44 Shapefile selection
Figure 45 Selecting the country field
Figure 46 Country field selected
Figure 47 Selecting the country field
Figure 48 Soter Unit identifier field selected
Glossary

GEOSS - Global Earth Observation System of Systems
SOTER – World SOil and TERrain Digital Database
SQL – Structured Query Language
XML – Extensible Markup Language
WFS – Web Feature Service
WMS – Web Mapping Service
This report is submitted as a formal deliverable for the EU Framework 7 project ‘e-SOTER – Regional Pilot Platform as EU Contribution to a Global Soil Observing System’, Grant Agreement Number 211578. The work described here was conducted under Work Package 6 of this project.

Citation. This report may be cited as follows: Simms, D., Reuter, H., Hallett, S., Tempel, P., Yigini, Y., Leibovici, D., and Pourabdollah, A. (2013) e-SOTER Portal Cookbook. 73pp. ISRIC, NL.

Disclaimer

E-Soter is furnishing this item "as is". E-Soter does not provide any warranty of the item whatsoever, whether express, implied, or statutory, including, but not limited to, any warranty of merchantability or fitness for a particular purpose or any warranty that the contents of the item will be error-free.

In no respect shall E-Soter incur any liability for any damages, including, but limited to, direct, indirect, special, or consequential damages arising out of, resulting from, or any way connected to the use of the item, whether or not based upon warranty, contract, tort, or otherwise; whether or not injury was sustained by persons or property or otherwise; and whether or not loss was sustained from, or arose out of, the results of, the item, or any services that may be provided by E-Soter.

Report © e-SOTER WP6, 2013
Overview
This document presents a ‘Cookbook’ description for installing the software tools necessary to develop and configure an e-SOTER web portal. The portal developed may provide the basis for project, national or regional datasets held in the SOTER format. The work presented here is a direct output and deliverable of the EU Framework 7 project ‘e-SOTER – Regional Pilot Platform as EU Contribution to a Global Soil Observing System’, Grant Agreement Number 211578. The work described here was conducted under Work Package 6 of this project.

The steps below describe the process of preparing a server to host the tools necessary to operate and run the e-SOTER web Portal and associated components. Initial steps are required to install each of the essential elements, followed by more detailed configuration of the setup and environment. In overview the steps are as follows:

- Server Preparation
  - Confirm server specification
  - Confirm root user and login; accessing server

- Key software installation
  - Install database management system
  - Install web services systems

- Software configuration
  - Database configuration
  - Integration of web services and schemas

- Sample data and services
  - SoTerML data
  - Simple Objects Services
  - Complex Object Services

Figure 1. Overview of the installation and configuration process for the e-SOTER web portal

Part of this cookbook provides information in loading in some example SoTerML datasets and publishing them as both simple and complex object web services. You can follow these examples for your interest, or adapt them for your own purposes.

SOTER in Context
Soil and land information is needed for policy-making, hands-on management of land resources, and for monitoring of the environmental impact of development. Lack of comprehensive information about land resources - globally, nationally or locally – means uninformed policies, continuing degradation of land and water resources, unnecessary carbon emissions to the atmosphere, and no likelihood of achieving the UN Millennium Development Goals. The viability and cost of vital infrastructure is affected just as much as food and water security and response to environmental change. In the case of
the European Soil Thematic Strategy, the operational measures laid down in the Framework Directive and Impact Assessment are hamstrung by lack of accessible, easy-to-use, consistent, harmonized and relevant soil data.

SOTER (World SOil and TERrain Digital Database) is a long-established and well-respected methodology for capturing and representing global soils information at small scales. SOTER is intended to support a series of applications such as “crop suitability, soil degradation, forest productivity, global soil change, irrigation suitability, agro-ecological zonation, and risk of drought”.

Development of the SOTER approach has culminated in the ‘e-SOTER’ project, which has sought to create harmonized SOTER database windows in western Europe between France and England; the Chemnitz region between the Czech and German borders; northern Morocco; and China (http://www.esoter.net/; http://eusoils.jrc.ec.europa.eu/projects/esoter/). Further to this the e-SOTER project has sought to develop new, improved methodologies to create better harmonized, more comprehensive and consistent data products using state of the art data processing, remote sensing and terrain modelling tools. As the European contribution to a Global Soil Observing System (part of GEO), a key element to these phases of activity has been the concurrent development of an e-SOTER web portal and related information spatial data infrastructure, designed to hold, manipulate and disseminate SOTER databases. Parallel with this has been the development of an XML-based markup language for representing the SOTER datasets, SoTerML (Pourabdollah et al, 2012).

The traditional SOTER implementation involves the development of a database (in MS Access) with a linked GIS dataset (as a Shapefile). These two ‘legacy’ data sources work together to hold geometric features and their linked spatial, properties. However, this approach is limited in a contemporary web-enabled information system where data should comply with the Open Geospatial Consortium (OGC) standards for web-based spatial services consistent with the guiding principles of the GEOSS initiative. In establishing such information systems there is a need to take advantage of the now growing suites of open-source software tools for each of the core components, comprising: database management; web services server; web server; geoprocessing tools and java servlet management. Such tools are open-source and so are free to use, providing a huge advantage for research teams and other stakeholders building such systems. However, open-source tools come with a high degree of complexity in their setup and configuration that can act as a counter to their comprehensive functionality. This manual is designed to guide the user through the process of establishing and configuring such a web portal, starting with a clean server installation.

As the architecture figure below shows, the portal components are designed to receive SOTER data in the form of a valid SoTerML ‘XML’ file, derived from the legacy data sources and produced by a SoTerML parser, described separately in Appendix Two. The data is loaded into a database built using the open source database, which is accessed by the GeoServer tool to provide spatial web-based services to external client applications, such as environmental models and web-mapping applications. The portal may be accessed via the Internet directly to access the data sources provided.
A typical Use Case scenario for the portal is that a SOTER project is undertaken which leads to the requirement for storage of the results in a database whose results can be shared. Also a legacy SOTER database may be converted to SoTerML to allow the widest access and usage of the data it holds, such that it can be uploaded to a SOTER web portal. The process for building this portal is described below.

**Platform**

The process descriptions below concern the configuration of the e-SOTER Web Portal server, commencing with a completely clean Linux installation. The server is assumed to have the following or equivalent specification:

- Linux - Debian
- Memory should be 1 Gb ++
- Disk capacity should be 100 Gb ++
- Visible on the Internet (e.g. in a DMZ ‘demilitarised zone’)

**Server and Operating System**
The Debian flavour of Linux is selected here by preference as it is generally considered a stable platform for building web services and is in common, widespread use for hosting such applications. Contemporary information systems practice encourages the use of ‘virtual server’ configurations, whereby server images are created and maintained on a computer such as a blade server, capable of holding several such virtual images. The test environments for the e-SOTER web portal have involved using virtual servers and this approach is considered practical and pragmatic for the purposes. **Responsibilities for network security are not covered in this cookbook, but are considered a prudent and important part of managing servers visible on the Internet.**

A key initial step in the process described here is to determine the Internet Protocol address of the server to be used. In Linux this is achieved with the command ‘ifconfig’. In describing and then showing this command below we may also introduce the notation convention used throughout the cookbook. Log in to the server as the root ‘superuser’ and run the command, thus:

```
ifconfig
```

Take a note of your **Server IP address** below, for example:

```
111.222.333.444
```

For the purposes of this Cookbook, a fictional server IP address of ‘111.222.333.444’ is assumed as some commands require IP addresses to be entered and shown.

**Accessing the server**

You may choose to access the server at a direct terminal. However, more likely you may choose to use another computer with a terminal emulator to open a session on the server remotely. If you do this, it is recommended you use a terminal programme such as ‘PuTTY’, which provides a free and excellent implementation of Telnet and SSH for Windows and Unix platforms ([http://www.chiark.greenend.org.uk/~sgtatham/putty/](http://www.chiark.greenend.org.uk/~sgtatham/putty/)). Equally, for a more graphical remote interface to the server, the ‘Tight VNC’ Server may be used (the server software being installed on the Debian computer server, and the client software on the remote access computer).

**Linux Commands**

Learning Linux can be daunting – some documentation is here [http://www.debian.org/doc/#manuals](http://www.debian.org/doc/#manuals) best is the one page reference card here [http://www.debian.org/doc/#other](http://www.debian.org/doc/#other).

**Initial Installation**

As a first step it is necessary to ensure the server has all necessary updates. The examples below use the latest versions of the open source toolsets employed at the time of writing. The nature of the open source tools used here is such that when you are installing this on your server the versions will most likely have changed. You should therefore substitute the version numbers noted throughout the cookbook for the latest equivalents. Logging in as the ‘superuser’ root user, this is achieved as follows:
Java
Java is a core requirement of the web portal. The server will require the Java Development Kit (JDK). If this is not present, install as follows:

`apt-get install sun-java6-jdk`

This installs Java on the server. This installation commences as a command-line process. During the installation, a graphical window appears with Sun’s terms and conditions. The ‘OK’ option for this is activated by pressing the tab key, and then entering return. Further pages of options may then be shown, again one uses the tab key to switch between selections and the return key to select. The following installation process can take some time (c. 10 minutes).

Once Java is installed, updates to the installation should be undertaken to ensure you have the latest edition. This is achieved as follows:

```
update-alternatives --config java
update-alternatives --config javac
```

The above commands may quite legitimately report no actions are required. As time passes, these commands may also be used to keep Java updated on the server.

Apache
The next step is to install the web server ‘Apache’, as follows:

`apt-get install apache2 unzip nmap`

To ‘restart’, ‘stop’ or ‘start’ apache, use the following command (varying the last word)

`service apache2 restart`

Tomcat
Following this you must also install the Java Servlet engine ‘Tomcat’, as follows:

`apt-get install tomcat6`

Postgres / PostGIS
The next step is to install the Postgres database with the PostGIS extensions. This is achieved as follows:

```bash
apt-get install postgresql postgresql-server-dev-8.4 postgresql-8.4-postgis postgresql-client
apt-get install postgresql-contrib-8.4
apt-get install postgresql-8.4-postgis
apt-get install postgresql-contrib
```

The steps above install the basic filesets required for the Postgres installation; configuration steps are described below.

**Python**
The Python language is required also, this is installed as described:

First, an initial check should be conducted to establish if python is already installed, this is achieved as follows:

```bash
python
```

If already installed, python reports as follows:

Python 2.6.6 (r266:84292 ......etc... *(use Ctrl-D to quit)*

If Python is not installed, it should be as follows:

```bash
apt-get install python
```

Next, the python module for postgis should be installed. This is undertaken in two stages:

**Stage 1** involves installing the python dependency

```bash
apt-get install python-pygresql
```

**Stage 2** involves installing the python module itself

```bash
apt-get install postgresql-plpython-8.4
```

**Perl**
Finally, perl scripting support should be added as follows:

```bash
apt-get install postgresql-plperl-8.4
```
Wrap-up
Once the installations are concluded certain packages and dependencies that are no longer required can be removed with the command:

```
apt-get autoremove
```

Once all the steps above are finished, your server will have all the requisite software components for the e-SOTER web portal. All that remains now is to configure these installations.

Configuration of Installed Software
The next section involves progressing through the steps required to configure the essential software components.

Postgres Configuration – Preparing for the database
The steps below establish a working database in Postgresql with support for PostGIS.

To undertake the next few steps, you will need to login as the database superuser by using su command as follows:

```
su - postgres
```

Note that this user has been created automatically during the initial Postgres installation. Now, logged in as the ‘postgres’ user, you can create a new database template with the Postgis components installed, thus:

```
createdb template_postgis15 -U postgres
```

Next use the psql command to add the language configuration as follows:

```
psql -d template_postgis15 -U postgres -c "CREATE LANGUAGE plpgsql"
```

Next, move to the correct working folder for the postgis installation scripts, for example:

```
cd /usr/share/postgresql/8.4/contrib/postgis-1.5
```

and run the setup SQL scripts, thus:

```
psql -d template_postgis15 -U postgres -f postgis.sql
psql -d template_postgis15 -U postgres -f spatial_ref_sys.sql
```

Finally, you can install the help description for each of these functions such that it can be searched via psql or pgadmin. To do this first change folder and then run the command, thus:
Postgres Configuration - Communications

The next step enables the database to make external communications. To do this we need to edit the Postgres configuration file directly. This can be achieved in the Debian GUI, if installed, or at the command line with an editor such as Pico, Nano or Vi. Note you should still be logged in as the user ‘postgres’ to do this.

```bash
cd /etc/postgresql/8.4/main
pico postgresql.conf or vi postgresql.conf
```

In the editor, edit the ‘listen_addresses’ line to read as follows:

```sql
listen_addresses = '*'
```

**Note:** Entering a ‘*’ in this way opens up communications to the Postgres server from all outside remote hosts. You may wish to take some advice from your network security specialist concerning this. For example this server may be located behind an institutional firewall and not be in the ‘DMZ (demilitarised zone)’ on your network.

Next you edit the client authentication configuration file, named ‘pg_hba.conf’.

```bash
cd /etc/postgresql/8.4/main
pico pg_hba.conf or vi pg_hba.conf
```

Edit the file as required to allow ssl connections. For instance, to allow such ssl connections from ISRIC’s servers the following line is added:

Under the section IPv4

```sql
host all all 137.224.252.10 255.255.255.0 md5
```

Once these edits are made, the configuration files should be reloaded, thus:

```bash
/etc/init.d/postgresql reload
```

It could be that a more authoritative restart is required to achieve this, thus:

```bash
/etc/init.d/postgresql force -reload
```

Or even

```bash
psql -d template_postgis15 -U postgres -f postgis_comments.sql
```
It is good practice to change the password for the Postgres database from the installation default, as follows *all on one command line*:

```
psql -d template_postgis15 -U postgres -c "ALTER USER postgres WITH PASSWORD 'xxxxxxx';
```

Here the password is being set to *xxxxxxx* – you should choose your own password. The result ‘ALTER ROLE’ is then reported back to indicate success.

**Make a careful note of the database user/password!**

This process changes the password for the Postgres database. Finally it is good practice also to change the password for the Debian Linux user ‘postgres’ to the same password as the Postgres database user.

**Note:** Remember the system user account ‘postgres’ is not the same as the Postgres database user account ‘postgres’. However, best practice is to ensure both accounts share the same password.

You should currently be logged in as the user ‘postgres’, log back out from this user to revert to the superuser account.

```
postgres > exit
> whoami
root
```

Use the `passwd` command to change the password of the postgres system account, thus:

```
passwd postgres
```

Follow the instructions onscreen to alter the password.

**Make a careful note of the postgres user password!**

```
Postgres Password: _____________
```

**Installing GeoServer**

The GeoServer tool provides published web data services for datasets held in the postgis database.

The software needs to be installed on the server, as the root superuser, thus:

**Preparing the server for GeoServer**

The first requirement is that there are created the appropriate folder structures to hold geoserver, thus:
mkdir /opt/cache/geoserver
chown -R tomcat6:nogroup /opt/cache/geoserver
chmod -R u=rwX /opt/cache/geoserver
chmod -R g=rwX /opt/cache/geoserver
chmod -R o=rX /opt/cache/geoserver
mkdir /opt/geoserver_data/
chown tomcat6:nogroup /opt/geoserver_data/
chmod -R u=rwX /opt/geoserver_data/
chmod -R g=rwX /opt/geoserver_data/
chmod -R o=rX /opt/geoserver_data/

Next run the nmap command to provide an overview of the network environment; nmap is a utility used to inspect a network or for security auditing.

nmap localhost

**Note:** the ports 80 (http); 5432 (postgres) and 8080 (http proxy) should be open on the server.

The ‘nmap’ command provides a range of other useful information concerning the server communication settings.

![nmap output]

**Figure 3 Running nmap on the server**

One must then confirm that the java servlet engine ‘tomcat’ is running correctly. Tomcat runs on port ‘8080’ by default, thus for server ‘111.222.333.444’ by example, one should try to open the webpage
on a browser on a separate computer at http://111.222.333.444:8080/. Hopefully you will see the page below.

![Tomcat installation success webpage](image)

**Figure 4 The Tomcat installation success webpage**

Once Tomcat is operational, the next step is to move on to the GeoServer installation.

**Obtaining and installing GeoServer**

The GeoServer fileset may be downloaded into a temporary folder somewhere convenient (for example /home/) on the server, thus:

```
mkdir install
cd install
wget http://downloads.sourceforge.net/geoserver/geoserver-2.2.4-war.zip
unzip geoserver-2.2.4-war.zip
```

The files so unpacked should then be copied to their destination folder location, thus:

```
 cp geoserver.war /var/lib/tomcat6/webapps/
```

**Obtaining and installing GDAL**

GDAL, or Geospatial Data Abstraction Library is a translator library for raster geospatial data formats (see [http://www.gdal.org/](http://www.gdal.org/)). One must then ensure that GDAL is correctly installed. First ensure it is not already installed, thus:

```
apt-cache search gdal*
```

Assuming GDAL is not already so, it may be installed thus:

```
apt-get install libgdal1-1.6.0 gdal-bin
```
Configuring the Apache Web Server
The Apache webserver needs to be configured to work with the other tools in the e-SOTER web portal, thus:

Changing the Apache redirects for Proxy Handling
It is necessary to change the redirects in Apache, to do this use an editor such as Pico or Vi to edit the default file, thus:

```
cd /etc/apache2/sites-available/
pico default or vi default
```

_In the file, add a new clause to manage proxy handling:_

```
ProxyRequests Off
<Proxy *>
  Order deny,allow
  Allow from 137.224.252.10 [or whatever your server IP is …]
</Proxy>
ProxyPass /geoserver http://localhost:8180/geoserver
ProxyPassReverse /geoserver http://localhost:8180/geoserver
ProxyPreserveHost On
```

Proxy Request configuration is carried out to ensure Apache handles geoserver traffic correctly.

Next, one can use the ‘a2enmod’ command to enable the apache2 modules, creating the proper links to the module .load and .conf files, thus:

```
a2enmod proxy
a2enmod proxy_http
```

Restarting Apache and Tomcat servers
Once the configuration edits are made as described above, the Apache and Tomcat services should be restarted to pick up these edits (as the root superuser), thus:

```
/etc/init.d/apache2 restart
/etc/init.d/tomcat6 restart
```

If you receive an error, check the ‘default’ configuration file settings carefully to ensure they are as in the section above.

Next you will need to edit the tomcat security and properties settings in the file ‘tomcat6’ with an editor such as Pico or Vi, thus:
cd /etc/default/
pico tomcat6 or vi tomcat6

The settings in the file should be set thus:

```
```

It is suggested the tomcat service now be restarted again to ensure the settings were entered correctly, thus:

```
/etc/init.d/tomcat6 restart
```

**Confirming the operation of GeoServer**

Finally, one should confirm the correct installation and operation of GeoServer. Run this by opening a web browser session to the server from a separate networked computer. Enter the web address of the server, with the subfolder ‘/geoserver’. For instance for the server ‘111.222.333.444’, enter the following (noting the port number is ‘8080’):

```
http://111.222.333.444:8080/geoserver
```

Hopefully you will now be able to see the GeoServer page thus:

![GeoServer home webpage](image)

**Figure 5 The GeoServer home webpage**

Note the default GeoServer username and password, respectively is ‘admin’ and ‘geoserver’. Ensure you can log in correctly to GeoServer, thus (don’t tick ‘remember me’ until you change password):
It is a good idea now to change the password of the geoserver administration account. To do this, use the web gui interface and locate the ‘Users’ category in the menu in the left margin. Click then on the user admin in the next dialogue to show the settings for the account. From here the password can be changed.

**Figure 6 Logging in to GeoServer**

Once logged in successfully, locate the ‘Server Status’ and ‘Logs’ pages to ensure all is running well. Next open the ‘Layer preview’ to review some of the sample datasets. Open one or two layers in the formats ‘OpenLayers’, ‘KML’ and ‘GML’ (for example SF:Streams).
Uploading Sample Data to the Portal

As a part of this cookbook, we have provided some sample SoTerML data to enable a SOTER application to be established. There are several themes to this data which all need installing and configuring. This section takes you through these steps.

In the accompanying Digital media (Appendix One), there are provided a range of sample files to enable a demonstration of the eSOTER Web Portal. Sample data files are also available from the e-SOTER project website. These sample files may be loaded onto the server to allow their loading and activation in the system.

Traditionally, SOTER databases were held in dBASE and then MS Access formats. A ‘parser’ has been written to process legacy SOTER datasets held in the Access format into the new SoTerML format. Appendix Two presents the user guide to the parser and the digital source code files for it. The process steps typically involved are as follows:
Using ftp to upload the sample data
Sample datasets are provided and these need to be moved to the new Debian server.

Installing FTP clients
To upload the sample data files to the Debian server you can use WGET or FTP. To achieve this, one may install a copy of the ftp client software. To achieve this, ensure you are the root super user and type:

```bash
apt-get install ftp
```

To connect to the server from a PC to copy files using ftp, you need a local client, the open source tool ‘FileZilla’ is highly recommended for this task for PCs. To obtain this programme, see http://filezilla-project.org/. Another popular free ftp tool by contrast is WinSCP, see http://winscp.net/eng/index.php. Either tool will suffice.

Installing an FTP Server
As a separate procedure to this, you may also wish to install an ftp server on the debian computer. If you choose to do this, it is recommended to use the ‘proftp’ server tool, installed thus:

```bash
apt-get install proftpd
```

As with many tools in Linux, there are often more than one options available for a given software functionality. So the VSFTP ftp server may also be preferred over proftp.

VSFTP Installation

The vsftpd may be installed thus:

```bash
sudo apt-get install vsftpd
```

Edit the vsftp configuration file
pico /etc/vsftpd.conf

Search through the file and change the following lines:

```
anonymous_enable=YES Change To anonymous_enable=NO #local_enable=YES
Change To local_enable=YES #write_enable=YES Change To
write_enable=YES
```

Also, add a line to the bottom of the file:

```
force_dot_files=YES
```

Quit the editor and restart the vsftpd server

```
service vsftpd restart
```

You cannot ftp to a server using the root account so you may also need to create a new user on the debian server using the ‘adduser’ command thus (to create a new user called ‘newuser’):

```
adduser newuser
```

[Choose your own name. Follow the instructions for adding passwords and user account details]

Once you can connect to the debian server you can copy over the sample data.

**Note:** As a security precaution, it is suggested you may not wish to leave the ftp server running all the time, but to start it only when required and stopping the service at other times.

Once the data is copied, you may wish not to leave the ftp server running. In this case you can stop the server with the following command:

```
/etc/init.d/proftpd stop
```

To start (restart) the process, type:

```
/etc/init.d/proftpd start
```

**Copying over the Sample SoTerML data**

Using the ftp tools, copy over the filesets identified in Appendix One. It is suggested that a temporary location be made available to upload the files to, from where they can be copied as required:

For example, here we create a folder in /home to receive the files:

```
mkdir /home/portal_data
```
Next use ftp to upload all the files identified in Appendix One to this location. When completed one should have three sub-folders, namely:

<table>
<thead>
<tr>
<th>Client</th>
<th>Sample SoTerML file and loading utility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Db</td>
<td>Database configuration files</td>
</tr>
<tr>
<td>GeoServer</td>
<td>Webservices configuration filesets</td>
</tr>
</tbody>
</table>

**Creation of the e-SOTER Database**

The next step is to create a database to hold the e-SOTER datasets. This database will be called ‘esoter’ and will be created from the postgis database template created earlier, thus:

```bash
su postgres
createdb -T template_postgis15 esoter
```

**Using PGAdmin – the graphical database administration tool**

The best means to administer a Postgres database is with the ‘PGAdmin’ tool. pgAdmin is a programme independent of the database and can be downloaded from [http://www.pgadmin.org/index.php](http://www.pgadmin.org/index.php) compiled for a range of computer architectures (e.g. PC, Mac and Linux flavours).

pgAdmin has a graphical user interface and allows visual, fine control over the various elements of the database. PGAdmin can be installed on a Windows PC as a standalone programme and used to access the Debian server database ‘esoter’ created above:

Once PGAdmin is running you can make a connection to the Debian server. Use the ‘File’ -> ‘Add Server’ menu option and complete the dialog box as follows:

![Figure 9 Connecting PGAdmin to the server](image-url)
Once connected the new database is displayed in the tree hierarchy of PGAdmin, thus:

**Figure 10 The PGAdmin Console**

The next step is to load in the database definitions for the eSOTER database. These definitions are held in the folder ‘db’ of the downloaded sample datasets (see Appendix One). Locate the file ‘soter_v3.sql’.

In PGAdmin, place the cursor on the new database name (‘esoter’) and then open the PGAdmin SQL editor. This ensures that the tables this script creates will be written into the eSOTER database itself, thus:

**Figure 11 Selecting the SQL editor in PGAdmin**

Then load into the editor the file ‘soter_v3.sql’, first discarding any contents already in the editor, thus:
The script can be run with the ‘Execute Query’ command, thus:

A number of messages will appear in the feedback window indicating the progress of the script. Once completed you can view the newly created tables in the hierarchical folder view.

**Note:** you MUST ‘refresh’ the folder view to show these new tables!
If by mistake you find you have created the new tables in a different database, for instance the template database, don’t worry! The tables will all need to be deleted however and this is achieved by deleting all of the tables one by one from the database, except the `spatial_ref` and `geometry_columns` tables, via right mouse click option DROP CASCADED, thus:

![Figure 15 Deleting tables in PGAdmin](image)

**Note:** remember to refresh the view afterwards if you do this! This is a right mouse button option.

Having loaded the file ‘`soter_v3.sql`’, the same procedure can be followed to load and execute the SQL that writes the soter stored procedures to the database, also located in the ‘db’ folder (see Appendix One), ‘`pg_soter_upload3.sql`’, thus:

![Figure 16 Loading SQL code to the SQL editor in PGAdmin](image)
Once this has run correctly, use PGAdmin to ensure that the new functions have been created successfully. Remember to expand and refresh the functions view!

![The Function view in PGAdmin](image)

**Figure 17 The Function view in PGAdmin**

Once the basic database installation is complete, the sample datasets can be loaded.

### Installing the SoTerML processing tools

Within the sample datasets provided (see Appendix One) are the ‘Client’ files required for processing SoTerML ‘XML’ datasets. The files can be left in the uploaded location for now. Ultimately one may wish to relocate these files elsewhere on the server. The ‘soterUploader’ tool loads SoTerML XML into the Postgres database. Obtaining a SoTerML XML file from a traditional e-SOTER format database is described in Appendix Two.

The tool is called ‘soterUploader.jar’ written in Java and has the following arguments:

```
-d  The postgres driver class (org.postgresql.Driver )
-url The path to the database jdbc:postgresql://STELLAR/upload
-usr The database user name
-pwd The database user password
-SoTerML The path to the SoTerML to upload (eg C:/SoTerML.xml)
-pj   The name of the project being uploaded (eg KenyaSoterDB)
```

To run the uploader, enter the following command, substituting your database parameters:

```
```

As the process is undertaken, messages may appear onscreen informing the user of the progress of the operation. SoTerML files can contain a lot of data and is loaded in chunks over the connection to the database. Once complete, the Postgres database will contain the data sourced from the example SoTerML ‘XML’ datafile.
Optimising and managing your PostgreSQL database is beyond the scope of this cookbook but instructions and guides can be found in the online PostgreSQL user community.

Configuring GeoServer for a Simple Objects web Service
The SoTerML data source contains a large array of nested data types, all interlinked. Presenting all these interlinked data ‘objects’ at once requires a so-called ‘complex object’ web service from GeoServer. However, many software client tools cannot handle complex objects. Instead these applications require so-called ‘simple objects’, where by a geometric point, line or polygon is accompanied by attribute data that relate specifically to that feature. Simple object web services do not therefore present interrelationships between object types and their geometry. Complex objects are described below.

As a test, one may now set up and publish a simple object web service for one of the geometric feature types in the newly loaded SoTerML data file to ensure the installation so far is working correctly. In GeoServer, click on ‘Stores’ in the left menu bar:

Add the data store configuration details. Remember that for ‘User’ and ‘password’ you enter the Postgres database account details. Enter your own Postgres database name also. You are then presented a list of the feature types holding geometric data from the SoTerML test file. Select, for example, ‘Profile’. The ‘Edit Layer’ dialogue appears allowing you to enter in basic layer details.

**The specification for the EPSG:4326 spatial reference in GeoServer layer properties is** ‘urn:ogc:def:crs:EPSG:6.6:4326’ (For more on EPSG, see http://www.epsg.org/main.html). This value ensures that data loaded in lat/long axis order is displayed correctly in the resulting data services. For more details see:

http://docs.geoserver.org/latest/en/user/services/wfs/basics.html

Bounding boxes are required, and you may use the ‘Compute from data’ and ‘Compute from native bounds’ options to enter these details automatically. When finished save the layer.

Finally, select the left bar menu option ‘Layer Preview’ to confirm all worked correctly. Locate the new layer you have just added and use OpenLayers to view the result, thus:
Simple feature services can be created for any table in the database - including views - using SQL. For example, running the following SQL from within an eSOTER database will create a dataset of all units that were uploaded under the project title ‘myProject’:

```
CREATE VIEW soter_units_mp AS SELECT soter_unit.soter_unit_id, soter_unit.geom FROM soter_unit
LEFT JOIN project on project.project_id = soter_unit.project where project_name = 'myProject';
```

Note the inclusion of the geometry column, which allows the data to be visualised. Using GeoServer you can now add the table to your server in the same way as the ‘Profile’ table and check the results in ‘Layer Preview’.

Tables and views can also be created dynamically using procedures stored within the database. Sample SQL is included with the upload scripts for building content that can be published as simple feature services.

**Testing the Server WFS Capabilities**

The Web Feature Service (WFS) capabilities of the server may be tested by accessing the web services from the web browser of a separate computer, thus, for example if the server IP address is ‘111.222.333.444’, enter the URL:

```
http://111.222.333.444:8080/geoserver/wfs?request=getcapabilities
```

The resulting XML page display can be searched to locate the service you have just created, e.g. Chemnitz (one of the example e-soter datasets) thus for part of the file:
Testing the Server WMS Capabilities
The Web Map Service (WMS) capabilities of the server may be tested by accessing the web services from the web browser of a separate computer, thus, for example if the server IP address is ‘111.222.333.444’, enter the URL:
http://111.222.333.444:8080/geoserver/wms?request=getcapabilities

The resulting XML page display can be searched to locate the service you have just created, e.g. Chemnitz (one of the example e-soter datasets) thus for part of the file:

Configuring GeoServer for Complex Objects
As well as presenting subsets of the whole SoTerML data source in the form of Simple Objects web services, it is also useful to be able to serve up web services that offer complex objects, whereby the entirety of interrelated objects are presented together in the same XML datastream served up by Geoserver. To do this requires further configuration of the GeoServer installation, the first step of which is to install the GeoServer Application Schema extension. It is important to ensure the Schema Extension version matches your version of the GeoServer installation.

**Installing the GeoServer Application Schema Extension**

GeoServer provides support for a broad selection of simple feature data stores, including property files, shapefiles, and JDBC data stores such as PostGIS. The app-schema module takes one or more of these simple feature data stores and applies a mapping to convert the simple feature types into one or more complex feature types conforming to a GML application schema. e-SOTER uses such a complex schema and so the application schema extension for GeoServer is required and should be installed. First locate a convenient folder to download the files to, for example ‘/home/install/geoserver’:

```bash
mkdir /home/install/geoserver
cd /home/install/geoserver/
wget http://downloads.sourceforge.net/geoserver/geoserver-2.1.1-app-schema-plugin.zip
unzip geoserver-2.1.1-app-schema-plugin.zip
cp *.jar /var/lib/tomcat6/webapps/geoserver/WEB-INF/lib/
```

**Configuring the GeoServer workspace folder**

Sample datasets are provided to help configure the GeoServer services for SoTerML, see Appendix One.

**Copying the mapping files for workspaces**

Defining complex objects from simple objects is achieved my mapping files in GeoServer. These files define the nesting simple objects that make up the complex objects. These files can be copied to the workspace subfolder of the stml folder of the geoserver data directory. Presuming the sample datafiles are located in /home/portal_data/downloaded/Geoserver/workspaces/ the files can be copied across thus:

```bash
cp –R /home/portal_data/downloaded/Geoserver/workspaces/stml /var/lib/tomcat6/webapps/geoserver/data/workspaces
```

This copies the sample workspace data into place. The folders must now be assigned the correct privileges, thus:

*Note the commands below are each entered all on one line*
Editing the mapping files for workspaces
The mapping files require editing to reflect your own computing environment for the Soter database server details.

First, at the top level workspaces folder, confirm the setting for the file ‘namespace.xml’:

Ensure the namespace URI is correct, the current version is http://www.esoter.org/stml/5.2

Located in the subfolder ‘stml_SoTerUnit, there are a number of mapping files that require editing:

Open each xml file in turn and replace the ‘sourceDataStores’ settings within as required, see below:
Also ensure that the target type reference is correct for the version of SoTerML you are using: version 5.2 at the time of writing:

```
<targetTypes>
  <FeatureType>
    <schemaUri>http://www.schema.esoter.net/stml/5.2/stml.xsd</schemaUri>
  </FeatureType>
</targetTypes>
```

Note the file ‘datastore.xml’ should also be checked for the correct namespace.

```
<entry key="namespace">http://www.esoter.org/stml/5.2</entry>
```

**Editing the mapping files for schemas**

The schema files copied over should **not require editing**. However, inspect the files to familiarise yourself with them.

**Testing the Installation**

One should now be in a position to see if the server serves up data correctly. First as a precaution it is suggested the Tomcat service be restarted to ensure it is using all the edited configuration files. As the root user this is achieved thus:
This may take longer than any previous restarts as the software will automatically download the SoTerML schema documents. You should make sure you are connected to the internet. The schema documents can also be loaded manually, refer to the GeoServer documentation for further details (http://docs.geoserver.org/latest/en/user/data/app-schema/app-schema-resolution.html).

**Testing the Server WFS Capabilities**

The Web Feature Services (WFS) capabilities of the server may be tested by accessing the web services from the web browser of a separate computer, thus, for example if the server IP address is ‘111.222.333.444’, enter the URL:

http://111.222.333.444:8080/geoserver/wfs?request=getcapabilities

Figure 21. WFS capabilities of the server

**Note that if there is no stml wfsoutput here, something is WRONG**

Further to consulting the capabilities of the service, try to test features can be received, thus, for example if the server IP address is ‘111.222.333.444’, enter the URL:

http://111.222.333.444:8080/geoserver/wfs?request=GetFeature&typeName=stml:SoTerUnit&maxFeatures=1
Debugging

It is possible that you may encounter some errors when undertaking the steps noted above. Inspect the system error logs to try and track down the problem. The error messages are held in a log file, found at

/var/log/tomcat6/catalina.out

If exceptions occur they are reported, thus: java.io.IOException: Schema ‘stml:attribute’ do not exist.

As data is written to this logfile, it may be useful to view it for debugging purposes. To watch data as it is written to a file the command ‘tail’ can be used (being the opposite to the ‘more’ command), thus:

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

Running GeoServer in a production environment

See the online documentation for optimising your GeoServer installation:

Installing Geo-network

GeoNetwork is a catalog application to manage spatially referenced resources. It provides powerful metadata editing and search functions as well as an embedded interactive web map viewer. It is currently used in numerous Spatial Data Infrastructure initiatives across the world.

```
# install geonetwork
# edit the apache
/* change apache redirects
pico /etc/apache2/sites-available/default
# add
ProxyPass /geonetwork http://localhost:8080/geonetwork
  ProxyPassReverse /geonetwork http://localhost:8080/geonetwork
/etc/init.d/apache2 restart
/etc/init.d/tomcat6 restart
```

Post Installation Configurations

**Geonetwork Username and Password:**

New installations of GeoNetwork use admin for both username and password. It is advisable, indeed important, to change this once installed. This is undertaken from the Administration page once logged on.

The example below shows a ‘walkthrough’ of the demonstration configuration. First login to GeoServer.
Figure 22. Administrator panel for Geonetwork

-data directory

<uploadDir>data/tmp</uploadDir>

[17:53:45] Jose García: <param name="dataDir" value="../../data" />

mkdir /var/lib/tomcat6/webapps/geonetwork/data
mkdir /var/lib/tomcat6/webapps/geonetwork/data/tmp
chown tomcat6:tomcat6 /var/lib/tomcat6/webapps/geonetwork/data
chown tomcat6:tomcat6 /var/lib/tomcat6/webapps/geonetwork/data/tmp

-background

blue background from demis.nl
geonetwork/WEB-INF/config-gui.xml

-for map search

uncomment localhost and add

<layer server="http://wms3.isric.org/geoserver/gwc/service/wms?LAYERS=bluemarble"
tocName="Map" params="{layers: 'Earth Image,Borders,Coastlines', format: 'image/jpeg'}" options="{isBaseLayer: true}" />
- for viewer

uncomment localhost and add

```html
<layer server="http://www2.demis.nl/wms/wms.ashx?WMS=BlueMarble"
tocName="Borders"  params="{layers: 'Borders,Coastlines', transparent: 'true', format: 'image/png'}"  options="{}" />

<layer server="http://www2.demis.nl/wms/wms.ashx?WMS=BlueMarble"
tocName="Ortophoto"  params="{layers: 'Earth Image', format: 'image/png'}"
options="{isBaseLayer: true}" />
```

- adjust image at top header

```bash
wget http://www.esoter.net/sites/all/themes/esoter/images/header.gif into /var/lib/tomcat6/webapps/geonetwork/images/
change root to tomcat ownership
#edit to adjust for the isric image
web\geonetwork\xsl\banner.xsl

```xml
    <img src="{/root/gui/url}/images/ISRIC_right.png" alt="ISRIC Logo " align="top"/>
    
    </td>
    
    <td align="right" class="banner">
        <img src="{/root/gui/url}/images/ISRIC_left.png" alt="WDC logo" align="top" />
```

#edit web\geonetwork\geonetwork.css

```css
td.banner
    { font-size: 8pt;
    color: #ffffff;
    background-image: url('images/ISRIC_bannerw.png ..... 
# if testing make sure the cache is empty
```

- change the system configuration to be similar to original meta site name:ISRIC site organisation: ISRIC World Soil Information
server: meta.isric.org
port:80
feedback email: meta.isric.org
smtp smtp.isric.org
port 25
-add harvest from fao
www.fao.org
port 80
servlet geonetwork
keyword soil
run only once
changed
<uploadDir>data/tmp</uploadDir>
<param name="dataDir" value="/data" />
mkdir /var/lib/tomcat6/webapps/geonetwork/data
mkdir /var/lib/tomcat6/webapps/geonetwork/data/tmp
chown tomcat6:tomcat6 /var/lib/tomcat6/webapps/geonetwork/data
chown tomcat6:tomcat6 /var/lib/tomcat6/webapps/geonetwork/data/tmp

- adding the group e-soter

The administrator can create new groups of users. User groups can correspond to logical units within an organisation.

To create new groups you should be logged on with an account that has administrative privileges. To log in, simply go to the home page and enter your username and password in the top right corner fields, then click on the login button.
The template options allow one to manage the metadata templates in the catalog. In order to use these facilities, you must be logged in as an administrator. After the login step, go to the administration page, thus.

To add or edit data and metadata, the user must be registered with an Editor profile or higher. That user should be a member of the User Group (e-soter).

Creating a New Record using the web based Metadata Editor

In the home page, click on the Administration Tab.

1. Select **New Metadata** from the List of the admin page.
2. Select the metadata standard **Template**, if possible, using the preferred ones. GeoNetwork opensource comes by default with support for three internationally-recognised metadata standards, ISO19139 ([http://www.iso.org/iso/catalogue_detail.htm?csnumber=32557](http://www.iso.org/iso/catalogue_detail.htm?csnumber=32557)), FGDC ([http://www.fgdc.gov/metadata](http://www.fgdc.gov/metadata)) and Dublin core ([http://dublincore.org/](http://dublincore.org/)). For the ISO standard, two templates have been developed; one for vector and one for raster data. Both contain a relevant set of elements to describe the respective types of data. More templates can be developed online.
3 Select the **Group Esoter** the metadata will belong to. These are the groups authorized to add metadata to by your administrator.

4 Click on **Create**.

---

**Step by Step: adding a metadata**

1. Enter username and password and click on the login button. The system will identify you and assign the correct privileges to work with

   ![Login bar for accessing Geonetwork administrator interface](image)

   **Figure 25. Login bar for accessing Geonetwork administrator interface**

2. Open the Administration page by clicking the Administration button in the banner and then click on the New metadata link.

   ![Main Page of Administration panel](image)

   **Figure 26. Main Page of Administration panel**

   From the metadata creation page, select the metadata standard to use from the dropdown list

   ![Metadata Creation Options](image)

   **Figure 27. Metadata Creation Options**

3. After selecting the correct template, identify which group of users the metadata will belong to and finally click on **Create**.
Figure 28. Metadata Creation Page User Group Selection

4 A new metadata form based on the selected template will be displayed to fill out (Figure 30) or, the option “IMPORT METADATA RECORD IN XML OR MEF FORMAT” can also be used to insert new metadata (Figure 31).

Figure 29. Metadata Creation Screen from a template
The Metadata for following data files are imported into Geonetwork using “Metadata Insert” option. The Metadata in XML format were created using ArcGIS 10.x Export Metadata Tool.

- **E-Soter Central European Window**
  - E-Soter H. Pilot Window – 1/250K
  - E-Soter Chemnitz Pilot Window – 1/250K
- **E-Soter Western European Window**
  - E-Soter UK Pilot Window- 1/250K
- **E-Soter Moroccan Window**
  - E-Soter Moroccan Pilot Window- 1/250 K
Figure 31. Metadata in Geonetwork
Installing the Algorithm Database

A mechanism was required that was capable of holding stakeholder uploaded algorithms, ready for processing into the information infrastructure. A number of approaches were considered, and the most practicable solution is recommended as an installation of the ‘Trac’ software tool.

Trac is an enhanced wiki and issue tracking system for projects. Trac uses a minimalistic approach to web-based software project management. Our mission is to help developers write great software while staying out of the way. Trac should impose as little as possible on a team’s established development process and policies. It provides an interface to Subversion (or other version control systems), an integrated Wiki and convenient reporting facilities. And GIT as TRAC plugin, is a distributed revision control and source code management (SCM) system.
#### install algorithm database - TRAC

trac and git repository

# trac & svn on same service

apt-get install trac

apt-get install trac-git

/opt/isric for svn

/opt/pwd for passwords

/opt/isric_trac for trac

/opt/git

cd /opt/git

git init

git commit

# get doc from server

http://jonas.nitro.dk/git/quick-reference.html

and store into /opt/git/


git add filename

git commit -m 'Initial commit'

git commit --amend --author='Your Name <you@example.com>'

trac-admin /opt/trac initenv

enter

git

/opt/git

find /opt/trac -type f -exec chmod 660 {} \; find /opt/trac -type d -exec chmod 2770 {} \; chown -R root.www-data 

/chown -R root.www-data /opt/git

apt-get install libapache2-mod-python
### apache default file

```
<Location /trac>
  SetHandler mod_python
  PythonHandler trac.web.modpython_frontend
  PythonInterpreter main
  PythonOption TracEnv /opt/trac
  PythonOption TracUriRoot /trac
  SetEnv PYTHON_EGG_CACHE /tmp
</Location>

<Location "/trac/login">
  AuthType Basic
  AuthName "trac"
  AuthUserFile /opt/pwd/trac.htpasswd
  Require valid-user
</Location>
```

htpasswd -cb trac.htpasswd hir esoter12

htpasswd -b trac.htpasswd wp WPesoter (TBREMOVED HIR)

http://portal.esoter.net/trac works
cHECK / /opt/trac/conf/trac.ini
git_bin = /usr/bin/git
add
cOMPONENTS]
tracext.git.* = enabled
[header_logo]
alt = esoter.net
height =
link = www.esoter.net
Figure 3. TRAC Home Screen

Opening and Managing Tickets in TRAC

A ticket contains the following information attributes:

- **Reporter** — The author of the ticket
- **Type** — The nature of the ticket (for example, defect or enhancement request).
- **Component** — The project module or subsystem this ticket concerns.
- **Version** — Version of the project that this ticket pertains to.
- **Keywords** — Keywords that a ticket is marked with. Useful for searching and report generation.
- **Priority** — The importance of this issue, ranging from *trivial* to *blocker*. A pull-down if different priorities where defined.
• **Milestone** — When this issue should be resolved at the latest. A pull-down menu containing a list of milestones.

• **Assigned to/Owner** — Principal person responsible for handling the issue.

• **Cc** — A comma-separated list of other users or E-Mail addresses to notify.

• **Resolution** — Reason for why a ticket was closed. One of the following choices: ‘fixed’, ‘invalid’, ‘wontfix’, ‘duplicate’, or ‘worksforme’.

• **Status** — What is the current status? One of the following: ‘new’, ‘assigned’, ‘closed’, ‘reopened’.

• **Summary** — A brief description summarizing the problem or issue.

• **Description** — The body of the ticket. A good description should be specific, descriptive and to the point.

With appropriate permissions a ticket entered into Trac can at any time be modified by **annotating**.

Then, annotations like changes and comments to the ticket are logged as a part of the ticket itself. When viewing a ticket, the history of changes will appear below the main ticket area.

All edits (field changes, new comments, comment edits) update the “last changed” time of the ticket. All edits (new comments, comment edits) update the “last changed” time of the ticket.
Figure 34. New Ticket Screen in TRAC

Installation and Configuration of the e-Soter Web Services Portal

E-Soter Portal – CMS Installation

The e-SOTER recommendation for web management is to utilise a ‘Content management System’, or CMS. The CMS offers a tool to separate management of the website from the content, allowing for edits to be made online. The well-known Wikipedia website for example operates on this principle. A number of CMS software tools exist, or which is was held that ‘Joomla’ was the most advantageous. Joomla is free, open-source software.

Installation / Main Configuration

Joomla uses a database such as MySql. To install Joomla one should first have configured an installation of a database. To configure MySql one can use phpMyAdmin graphical tool to establish username, database and password. phpMyAdmin can be installed as follows,

**Phpmyadmin**

We recommend the excellent ‘phpMyAdmin’ database web management console. All the database administration can be undertaken through a web page.
To install Phpmyadmin, enter:

```
apt-get install phpmyadmin
```

Select ‘apache2’ as the web server

Create a phpMyAdmin password; you should now be able to call up the phpMyAdmin home page at

```
http://111.222.333.444/phpmyadmin – login as root/<password>
```

Note user privileges to the database are all managed via phpMyAdmin. Each user account needs to have specified the host from which the user will come onto the database. The ‘%’ (all servers) option may not always be recognised and you may need to repeat a configuration for servers: ‘127.0.0.1’, ‘111.222.333.444’, ‘localhost’ and ‘<server name>’.

A bit fiddly – essentially you open up the privileges for the user and duplicate the settings varying the host each time. To help with this, note that once phpMyAdmin is installed, one can see the various options the phpMyAdmin user has pre-assigned. Note that connecting to the database from the SSH terminal uses ‘localhost’, whereas connecting from a webserver can use ‘127.0.0.1’ – so the source varies according to source. One needs to experiment!

Remember later that if you receive a ‘500’ error on the webserver when trying to access the database, it is likely due to the incorrect privileges. You may need to look at the webserver apache logfiles in a new window (‘/var/log/apache2/’) to actually see the error.

```
more /var/log/apache2/error.log
```

or better still to see only new lines as they appear in the log file

```
tail -f /var/log/apache2/error.log
```

Using PhpMyAdmin you must create database, username and password. Use the following:

```
Database: esoter
User: esoter
Password: __________
```

Having done this, the next task is to download the latest release of Joomla open source content management system (http://www.joomla.org/download.html). Requirements for Joomla are listed below.

### Table 1. Requirements for Joomla 3.x

<table>
<thead>
<tr>
<th>Software</th>
<th>Recommended</th>
<th>Minimum</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

More information
With the above requirements met, and the database created as noted above, and with the required Joomla! files placed in the server web folder (e.g. /var/www/), next step is to configure Joomla. The Joomla software includes a built-in web installer, activated by browsing to the site’s domain name, portal.esoter.net/installation/.

![Joomla!](https://www.joomla.org/images/joomla.png)

**Figure 35. Main Configuration page during the installation**

Joomla tries to identify the Select Language field automatically from your browser’s language. This can be changed if needed as required.
The following information should be completed.

- **Site Name**: The name the portal (E-Soter Web Services) — this can be changed at any point later in the Site Global Configuration page.

- **Description**: Enter a description of the website. This is the *global fallback* meta description used on every page which will be used by search engines. Generally, a maximum of 20 to 25 words is optimal. Again, this can be changed on the Site Global Configuration page at any time.

- **Admin Email Address**: The admin email address. Enter a valid email in case you forget your password. This is the email address where you’ll receive a link to change the admin password.

- **Admin Username**: Joomla! uses a default “admin” as the username for the Super User. And, it should be changed immediately (which a good Security measure) using My Profile in the Administration interface after installation.

- **Admin Password**: Remember that super user has maximum control of the site (frontend & backend), so try to use a difficult password. Use My Profile in the Administration interface to change it later. Confirm the password in the Confirm Admin Password box.

- **Site Offline**: Click the Yes or No box. Yes - this means when installation is complete, the Joomla! website will display the ‘Site is offline message’ when yoursitename.com is browsed to view the home page. No - this means the site is live when yoursitename.com is browsed to view the home page.

When everything on the first page is completed, click the **next button** to proceed:

**Database Configuration**
Figure 36. Database configuration

For simplification, these instructions are a reference to installing with a MySQLi database. The database refers to here is created previously in the steps listed above. The instructions on the installation page are self-explanatory, but are noted below:

- **Database Type**: MySQLi is the database type for use.
- **Hostname**: Identifies the name of your database server. In this case it is localhost.
- **Username**: The username used to connect to the database Username
- **Password**: The password for the database’s username Password
- **Database Name**: The name of the database Database Name
- **Table Prefix**: One is generated automatically, but it can be changed. For example, jos3_ can be used. Just don’t forget to put the underscore character (_) at the end of the prefix. Table Prefix
- **Old Database Process**: Should the installer backup or delete existing tables during the installation of new tables? Click, Yes or No to select the choice. In this case (which is a fresh installation) there are no such old tables in the database, so this step can be skipped.

All these choices can be edited on the Site Global Configuration page, under Server options subsequently, after the installation has completed.

When all the information has been filled in, click the **next button** to proceed:
Overview

The next steps finalise the Joomla installation. The last page of the web browser installation contains a summary of all the information about the installation. This includes the options (at the top) for installing sample data and (at the bottom) the installation’s configurations.

Install Sample Data and Email Configurations

The first options are for automatically installing sample content to the website and emailing the configuration settings. This step is for beginners wishing to see how Joomla and its database ‘backend’ works and operates – for e-soter installations this step should be skipped.

![Finalisation page to the Joomla installation](image)

Figure 37. “Finalisation” page to the Joomla installation

Checking Your Configurations

If everything is in order, you will see the install at the top of the overview page. If not, this is the place to check and see what may be causing an issue. Further details on learning how to configure Joomla are available online here [http://help.joomla.org/](http://help.joomla.org/).
This section is broken into 4 groups:

- **Main Configuration**: All the website specific information, such as the website name, description, admin username, etc.

- **Database Configuration**: Contains the information about the database Joomla will use.

- **Pre-Installation Check**: These requirements must all be shown as Yes. With the exception of the PHP Version, the rest are usually controlled in the `php.ini` file.

- **Recommended Settings**: These are settings recommended in the PHP configuration, but will not prevent Joomla! from being installed.

If everything is correct and all checks are passed, you may now click the Install button in the top right corner of the Overview page. This will start the actual installation process.

After you click the Install button, progress bar should be seen with additional information of the installation. Once the installation completes, the success page is shown.

**Success and Finishing Up the Installation**

Joomla! 3 is now installed, but there is one last step to complete the installation and start using the site. **Installation Folder should be deleted (or renamed)**. Click on Remove Installation folder and a success message will appear. Now the **Administrator pages can be navigated** by clicking Administrator or by navigating directly to the portal by clicking Site.
Visitor Tracking

**Visitors: Piwik** is a popular and capable open source web analytics system written by a team of international developers, and runs on a PHP/MySQL webserver.

Code (embedded in /var/www/modules/mod_footer/tmpl/default.php):

```javascript
<!-- Piwik -->

document.write(unescape('%3Cscript src="'+pkBaseURL+'piwik.js'+'" type='text/javascript'"%3E%3C/script%3E'));

</script>

<!-- End Piwik Tracking Code -->
```

**Epilogue**

This document has presented a ‘Cookbook’ description for installing the software tools necessary to develop a SOTER web portal, including all the various software tools and utilities. Further configuration will be required by any user, but it is hoped that the steps as outlined in this document will aid the process of designing, installing and commissioning a SOTER portal. The use of open source software
has been adopted wherever practicable throughout the process. Using open source tools has a huge strength of advantage in the cost is zero at the point of installation. However, users will have to observe how the tools and the elements within the wider system are updated and revised and adapt their own systems accordingly.

For further information regarding eSoter, please visit http://www.esoter.net/.
Appendix One: Digital Media
A digital inclusion is provided with this cookbook that provides digital copy of the filesets required to enable a demonstration of the eSOTER web portal.

Fileset One: Database files
The following files are provided, comprising sample PostgreSQL database functions and data structures required for eSOTER:

- `soter_v3.sql` – Database structures (not data)
- `pg_soter_upload3.sql` – Utility functions

Fileset Two: Client Data files
The following client data files are provided, comprising sample SoTerML files and the source Java code required to load the data into the database:

- `soterUploader.jar`
- `SoTerML_MAO.xml`
- `/src` – Folder containing all Java source code

Fileset Three: GeoServer files
Once the sample datasets are installed and operational in the database, the next step is to publish them as simple and complex web services via GeoServer. The following schema and workspace files are provided relative to SoTerML, or ‘stml’:

Schemas
- `schemas\stml`
- `schemas\stml\5.2`
- `schemas\stml\5.2\AttributeReference.xml`
- `schemas\stml\5.2\SoilClassification.xsd`
- `schemas\stml\5.2\stml.xsd`

Workspaces
- `workspaces\stml`
- `workspaces\stml\namespace.xml`
- `workspaces\stml\workspace.xml`
- `workspaces\stml\stml_SoTerUnit`
- `workspaces\stml\stml_SoTerUnit\TerrainComponent.xml`
- `workspaces\stml\stml_SoTerUnit\stml_SoTerUnit.xml`
- `workspaces\stml\stml_SoTerUnit\datastore.xml`
- `workspaces\stml\stml_SoTerUnit\Attribute.xml`
- `workspaces\stml\stml_SoTerUnit\ParentMaterial.xml`
- `workspaces\stml\stml_SoTerUnit\stml_SoTerUnit`
- `workspaces\stml\stml_SoTerUnit\stml_SoTerUnit\featuretype.xml`
Appendix Two: SoTerML Parser

Prior to the e-SOTER project, SOTER database traditionally comprised a combination of an MS Access database (to hold the attributes) and an ESRI Shapefile (to hold the geometry). This ‘loosely-coupled’ data source arrangement is replaced now by the XML mark-up language ‘SoTerML’, which in one file format holds spatial and attribute data.

One of the key issues for the e-SOTER project WP6 has been the transferral of large sets of legacy data files from the old SOTER format to the new SOTER-ML Soter XML-based mark-up language.

This Appendix presents a parser, written in the computer language Java, that can convert traditional SOTER databases to the new markup format. The eSoterParser tool documented here has been written by the Cranfield University team as a common utility able to do this by taking an existing standard SOTER data source and transforming it.

Further to this, an XML validation tool is presented that allows users to check the conformity of the XML so produced.

eSoterParser utility – Manual and Instructions
for v1.6

S. Hallett, D.Simms, A.Rayner, P.Tempel, H.Reuter, Y.Yigini, D. Leibowith, A. Pourabdollah

Introduction
The work presented here is a direct output and deliverable of the EU Framework 7 project ‘e-SOTER – Regional Pilot Platform as EU Contribution to a Global Soil Observing System’, Grant Agreement Number 211578 (see http://www.esoter.net/). The work described here was conducted under Work Package 6 of this project.

One of the key issues for the e-SOTER project WP6 is the transferring of any of the large set of legacy data files from the old SOTER format to the new SOTER-ML Soter XML-based mark-up language. The eSoterParser tool documented here has been written by the Cranfield University team as a common utility able to do this by taking an existing standard SOTER data source and transforming it.

Prerequisites
In order to run the utility, you should have a 32bit Windows operating system, having a valid ODBC driver for Access (this would be the default if you have MS Office installed).

Your computer must have the Java SE Runtime, or the Java SDK already installed and correctly configured on the PC.

To test that Java is correctly installed (e.g. all the paths and environment variables are set up correctly)
- type at the command prompt “java”. If you then receive a page full of java help text, then all is well, if not consult your java installation notes.

This code has been tested and runs successfully on standard Windows 7, XP and Vista computers.
Note, should you wish to edit the code etc., we would recommend using the free IDE toolkit ‘NetBeans’ (http://netbeans.org/). The NetBeans project file is included in the distribution.

**Instructions on use**

1. Download and unpack the compressed folder “eSoterParser.zip”, say to c:\eSoterParser\. 

2. All compiled binary files required to run the tool are contained within this folder (the distribution code was compiled in Java). The source code is also provided.

3. Sample SOTER databases are provided in the “SampleData” subfolder. The ‘esoter’ folder contains the datafiles created by the e-soter project for the three windows (see http://www.esoter.net/ for more details). The data files are in a legacy SOTER format comprising a shapefile and linked MS Access database.

The data files comprise:

<table>
<thead>
<tr>
<th>Data File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>eSOTER_v20110620_v2.mdb</td>
<td>MS Access data file with soil attribute data</td>
</tr>
<tr>
<td>CEU01.*</td>
<td>Central European window spatial data</td>
</tr>
<tr>
<td>WEU01.*</td>
<td>Western European window spatial data</td>
</tr>
<tr>
<td>MA01.*</td>
<td>Moroccan window spatial data</td>
</tr>
<tr>
<td>eSOTER_ONEFULLRECORD.mdb</td>
<td>a test database with one single but complete record. Use for testing</td>
</tr>
</tbody>
</table>

4. Change to the “dist” (distribution code) folder (“\eSoterParser\dist”).

6. Run eSoterParser by typing at the prompt:

```
java -jar "eSoterParser.jar"
```

7. This starts the GUI for the tool, allowing you to select the relevant legacy SOTER database files, starting with the appropriate mdb Access database. There are three distinct actions required to select the necessary files. The GUI guides you through with ‘tick’ marks that go green when each selection is made successfully.
The first step is to select the Microsoft Access (version 2007) containing the non-spatial data attributes. Use the first button to open a dialogue box to navigate to and select the MDB database file.

Note, after selecting the file the green tick appears to indicate completion of this task. The parser also identifies the version of the SOTER database and displays this in the Activity Log window:
The second step is to select the GIS Shapefile database (DBF file) containing the spatial elements of the Soter Units. Use the second button to open a dialogue box to navigate to and select the DBF database file.

Note, that after selecting the shapefile another green tick appears to indicate completion of this task. The parser also now identifies the data fields in the shapefile (dbf) database, allowing you to select which one is used to link together with the Access database.
Figure 44 Shapefile selection.

The third step is to select the Shapefile attribute database (dbf) field from the drop down combobox that holds the country definitions for the shapefile polygons. There must be a field holding country information, but the field name is now always consistent between SOTER databases – so a selection is necessary. The field required may be named “Country”.

Figure 45 Selecting the country field.

Once the field is selected, the user interface displays your selection.
The fourth step is to select the Shapefile attribute database (dbf) field from the drop down combobox that will be used to connect the records in this table with the records in the MS Access table. This is how the legacy SOTER file data structure represents the link between the geometric and attribute features of the data. The utility tool presents a dynamic list of field names for selection. The field required is usually named “SUID” (Soter Unit Identifier).

Once the field is selected, the user interface displays your selection.
The fifth step is to note the tick box for including all profiles. Each SoilComponent in the SOTER database can have a representative profile associated with it, plus a number of other supporting profiles. The representative profile will always be selected if present and output the ‘representativeProfile’ tag. You may then choose whether or not to include the other associated profiles (which will not include the representative profile), written out to the ‘profile’ tag. The tick box is on by default.

The sixth step is to select the “CreatSoterML” button to start the conversion. Conversion may take some time, but eventually, a new output SOTERML file is created in the database folder “SoTerML.xml”. This is a plain text file (although likely a rather large one). You can then view this SoTerML file or also view any errors that were reported.
Figure 50 Running the conversion.

A message box provides feedback as to the success of the operation. Further buttons allow you to view in Notepad the output SoterML.xml file and the Errors log, both files being created in the folder containing the source data. Rerunning the conversion overwrites these files.

Data outputs
The parser creates two files, ‘SoterML.xml’ and ‘Errors.log’.

The errors file shows any data exceptions encountered during the process.

The SoTerML file may be opened in any text editor for inspection, thus:

Figure 51 The SoTerML output.
The error file also provides useful information on the progress of the data transformation and any issues that might have occurred. It is a text file that can be opened in any text editor. Issues noted are likely to imply issues for checking in the source MS Access database.

Figure 52 The errors log.

Data validation
The ‘SoTerML.xml’ is a standard XML file and as such can be validated to ensure it is correctly formed. A standalone Validator is provided that uses SAX to validate the XML file. Validation is used to check the data in the XML file against the namespace information it contains. A command-line XML Validator tool ‘Validate-z1.jar’, also written in the computer language Java is provided to check conformity of the SoTerML file. It is used thus:

Usage:

```
java -jar Validate-z1.jar <path to xml>
```

Example of using the validator tool:

```
cd c:\location_of_validator
java -jar Validate-z1.jar c:\location_of_SoTerML\SoTerML.xml
```

Code documentation
The software code for the parser is commented throughout and this has been used to generate a JavaDoc file outlining the architecture of the programme. The JavaDoc is included in the distribution package ‘eSoTerParser\dist\javadoc’ as a self-contained web page that can be run in a browser by loading the index file, thus:
The JavaDoc file is of specialised interest to programmers wishing to understand the parser’s operations.

Source Code
The software source code for the parser is provided within the distribution. Developed in NetBeans, the project files are also provided to open the code in NetBeans as required.

Further Queries
For any queries relating to the SOTER parser, please contact:

Dr Stephen Hallett
Principal Research Fellow in Soil Informatics
National Soil Resources Institute (NSRI)
Bullock Building (B53)
Cranfield University
Cranfield
Bedfordshire
Digital attachment
The following file contains the materials relating to the parser as outlined above:

E-SoTer_PARSER_v1_6.zip