

SPP Configuration Guide

SPP Configuration Guide

Table of Contents

1. Getting Started	1
Downloading and installing the pre-requisites	1
Downloading the SPP package	1
WAR archive	1
Source code bundles	1
CVS	1
2. Installation	2
WAR archive	2
Building from the source code	2
Code layout	2
build.properties file	2
Building and Deploying	3
Starting the portal	3
3. Configuration	4
Before the first run	4
Localisation	4
Deploying a localised version	5
Creating a WAR archive	5
Changing the look and feel	6
Configuring navigation and layouts	6
PSML files	6
Portlets	7
Changing the portlets available in your installation	7
Writing a portlet definition	7
Service configuration	9
Mail	9
Account Management	10
Storage	13
Searching	14
Alerting	18
Newsfeeds	20
4. Running the Portal	22
5. Upgrading	23
6. Extending the SPP	24
Creating an SPP portlet	24
File locations	24
Backend functionality	24
The Portlet class	24
Views	25
State	25
Producing a view	25
User interactions	26
Convenience methods	26
Error handling	27
7. Setting up an SPP LDAP server	28
LDAP Server Installation and Configuration	28
sppSlapd.conf	28
sppEntries.ldif	29
sppLDAP.schema	30
edupersonLDAP.schema	30
TLS/SSL CA and Certificates Generation	31
LDAP Server Operation	32
SPP configuration	32
8. Known problems (and some solutions)	34
Ant	34
Empty newsfeeds	34
Working with a firewall	34

Chapter 1. Getting Started

Downloading and installing the pre-requisites

Before you can use the SPP packages you need to have a few software pre-requisites installed. You can download all these programs from the respective websites and install them.

Install Java (1.4 or greater) <http://java.sun.com/j2se/1.4/>

Install Tomcat (4.1 or greater) <http://jakarta.apache.org/tomcat/>

If you want to build and deploy from the source code you will also need the ant build tool:

Install Ant (1.5.1 or greater) <http://jakarta.apache.org/ant/>

Notes:

There may be a small problem with ant 1.6 at present. If so, we'd recommend using 1.5.1.

Downloading the SPP package

There are a few ways in which you can obtain the SPP code.

WAR archive

If you simply want the distributed generic package you can obtain it as a WAR archive, which can be dropped into your web servlet container's web application directory (e.g. Tomcat's webapps directory). You can obtain the WAR archive from here: [Note: this directory is password protected at present. Please contact <mailto:spp-dev@dev.portal.ac.uk> for access details.]

<http://dev.portal.ac.uk/dist/>

Source code bundles

Alternatively you can work from the source code, which can be retrieved from here:

<http://dev.portal.ac.uk/dist/>

Once the source code is downloaded, unpack the package

```
gunzip spp.tar.z
```

Then unpack the source code

```
tar xvf spp.tar
```

CVS

You can also download the SPP source directly from the CVS repository.

[Note: to be documented]

Chapter 2. Installation

WAR archive

Simply drop the WAR archive into the web applications directory of your servlet container and restart it. Note that you will have to make some configuration changes to take account of your local environment. Please see the Configuration section below.

Building from the source code

Code layout

If you have got the source code via CVS or from the distributed tarball, you will have a src directory. This is where all the building, and installation and configuration will take place. A listing of this directory will look like this:

build	All the build files lie here
build.xml	Configuration file that ant will use to build the SPP package
class	Class directory
configuration	Configuration directory
core	Directory that holds all the source cod.
dependencies	Directory that holds all the package dependencies
dev	Development directory
docs	Documents directory
localisation	Configuration directory for local install
portlets	Portlet directory

build.properties file

To build, you will need a build.properties configuration file in the src directory; this holds all the information that ant needs to successfully build the package. Copy the template at build/build.properties.in to build.properties, then edit it. In order to deploy the application it needs to know the location of your web servlet container directory [Note: tomcat assumed at present].

Example

```
tomcat.base =/usr/local/jakarta-tomcat-44
```

Also needed is the location of the webapps directory. You can use the value of tomcat.base in the assignment:

Example

```
tomcat.webapps = ${tomcat.base}/webapps
```

The final deployment value needed is the name of the directory into which the jetspeed portal framework application is expanded. It is important to note that the build process may modify or remove this directory.

Example

```
jetspeed.webapp = ${tomcat.webapps}/spp
```

If you are behind a firewall you should include details about your proxy so that the build process can download files it needs. Set these properties:

```
proxy.host
```

```
proxy.port
```

Save your changes and now you should be able to build and deploy the application.

Building and Deploying

The build tool, ant, is used to manage the building and deploying process. To use it you need to run it from the spp/src directory.

First you need to download any required libraries. To do this type

```
ant check-dependencies
```

You won't need to run this again, as it is checked every time you attempt a build or deploy. However, you need to run it initially as it fetches libraries required by other build tasks. Once it has completed you need to include a couple of files in your ant classpath.

```
dependencies/build/js.jar
```

```
dependencies/build/bsf.jar
```

You also need to make sure that catalina-ant.jar is in the ant classpath. This can be found in your tomcat 4 installation in the server/lib directory.

There are several ways to include these libraries in your ant classpath. You can simply copy them into your system's ant/lib directory, or, from ant 1.6, you can include them by using the -lib switch. See <http://ant.apache.org/manual/> for more details.

Once your ant classpath is correct you are ready to build and deploy.

Stop tomcat (if it is running) and type

```
ant deploy
```

Whenever this is run, it will check for (and download if necessary) any updates to the required libraries, build and deploy the application to your tomcat webapps directory.

You will need to restart tomcat for the new installation to be made available.

Type ant for further information on other build options.

Starting the portal

Once you have installed the portal application you can launch it by running your servlet container. See the section below on running the portal for details. Although you can launch it without any initial configuration some services will not work unless they are configured to your environment. See below for more details.

Chapter 3. Configuration

There are many ways in which you can alter the look and feel and the behaviour of the portal application, and these are detailed below. At a very minimum, you should make the changes suggested in section 3.1 below. After that you can choose to edit the files in your deployed web application, or to use the localisation build system. For anything other than minor edits the latter is the recommended route.

Before the first run

There are a few features that need to be configured for your local environment before they will work within your portal.

If you are working with a WAR archive, you will need to unpack it before you can edit these files.

If you are working from the source you can use the localisation directories to manage these changes. See the next section.

You should set your mail server properties here:

WEB-INF/spp/mailler.properties

To configure email confirmation of account creation edit this file. You will need to set your mail server details again:

WEB-INF/conf/JetspeedResources.properties

This is the minimum necessary to allow the software to run within your environment. Obviously there is much more that can be configured to tailor the subject portal to your needs.

Localisation

To ease the task of configuration you can make use of the localisation build system. This allows you to create your own custom configuration and template files in a separate directory, which then can be used to replace the default files during deployment.

To do so you will need to work from the source code to build and deploy your installation. Once you have this, you need to create, under the localisation directory, a directory tree for your configuration files. See the current entries for examples. These trees mimic the src/configuration and src/portlets file layout. Files in the localisation tree take precedence during deployment. You can overwrite as much or as little as you like. Anything not overwritten will take its default from the original src/configuration and src/portlets directories.

Here's some information about these configuration directories:

localisation/<locale>/configuration/copy/

- anything here will be copied to the root webapps/spp/ directory.

localisation/<locale>/configuration/copy/WEB-INF/

- application configuration files.

localisation/<locale>/configuration/copy/WEB-INF/conf/portlets.xreg

localisation/<locale>/configuration/copy/WEB-INF/conf/demo-portlets.xreg

localisation/<locale>/configuration/copy/WEB-INF/conf/admin.xreg

- overwrites the default jetspeed portlet lists. At present this just removes most of the supplied examples.

localisation/<locale>/configuration/copy/WEB-INF/psml/user/anon/html/en/default.psml

- layout for guest login page.

localisation/<locale>/configuration/copy/WEB-INF/psml/role/user/html/default.psml

- layout template for default logged in user page.

localisation/<locale>/configuration/copy/WEB-INF/spp/

- spp installation specific data (see below).

localisation/<locale>/configuration/copy/WEB-INF/templates/vm/layouts/html/default.vm

- overall screen layout.

localisation/<locale>/configuration/copy/WEB-INF/templates/vm/navigations/html/

- navigation templates.

localisation/<locale>/configuration/patch/WEB-INF/conf/JetspeedResources.properties.ant

- patch file for editing the default JetspeedResources.properties settings. Patch files allow parts of large default configuration files to be edited. See the file that results in your tomcat/webapps/spp/WEB-INF/conf for examples. e.g. switch off left nav bar.

localisation/<locale>/portlets/

- spp installation specific portlet configuration.

localisation/<locale>/portlets/spp/configuration/

- edit .xreg files to alter the list of available portlets and skins.

localisation/<locale>/portlets/spp/velocity/spp/

- templates for portlets defined above (e.g. is welcome.vm, used by the SPPWelcome portlet).

Deploying a localised version

To deploy a localised version of the package you need to follow these steps:

Shutdown your servlet container (if running).

Deploy with the following command:

```
ant deploy -Dlocalisation=xxxx
```

xxxx being your localisation directory.

Alternatively you can alter the build.properties file and edit the line which refers to it (the default is the 'dist' localisation). Then you can simply run ant deploy.

Restart your servlet container.

Creating a WAR archive

Alternatively you can create a WAR archive of your localised portal for distribution to another ma-

chine.

Shutdown your servlet container (if running).

Then:

```
ant create-war -Dlocalisation=xxxx
```

xxxx being your localisation directory.

You will be left with an spp.war file in your <spp-src>/dist directory.

Changing the look and feel

Remember that while you can edit these files within your web application it is easier to make large scale changes using the localisation build system detailed above.

Configuring navigation and layouts

The spp package largely uses velocity templates to manage the design and layout of the display. More information about velocity can be found at the velocity site: <http://jakarta.apache.org/velocity/>.

The main portal navigation and layout files resides under:

WEB-INF/templates/vm

This directory includes the following:

layouts	Contains the main default.vm layout file. This file defines the overall structure of the portal page. The portlets themselves are inserted at the place where \$screen_placeholder appears.
navigations	Contains the navigation files top.vm, bottom.vm and left.vm. These are included by the default.vm file at the appropriate place.

Note that the left navigation bar is by default off. This property is set in the WEB-INF/conf/JetspeedResources.properties file. If you are deploying from the source code you can create a patch file in your localisation tree at configuration/patch/WEB-INF/conf/JetspeedResources.properties.antd include a section like this:

```
<!-- switch on left nav bar -->
<replaceregexp byline="true" file="{fn}">
  <regex pattern="leftnav\.enable( *)=(.*)" />
  <substitution expression="leftnav.enable = true" />
</replaceregexp>
```

See the default version of this file for more information.

PSML files

The files under WEB-INF/psml describe the portlet selection and layout for every user of your

portal. They use an XML format called PSML. A guest user will be presented with a portal page built from this file:

WEB-INF/psml/user/anon/html/en/default.psml

Upon registration, a PSML file will be created for a new user, based on this template:

WEB-INF/psml/user/turbine/html/default.psml

See the default files for examples as to the file structure. More information about PSML can be found here: <http://jakarta.apache.org/jetspeed/site/psml.html>

You can edit these files by hand, or you can use the supplied PSML browser available in the admin account.

Portlets

Portlets consist of a xml portlet definition and some Java code. You can find guidelines on writing your own Java portlets in a section below, but you can also modify the default list of portlets provided in the distribution by changing the portlet definition files. You can find this file here: WEB-INF/conf/portlets.xreg.

Changing the portlets available in your installation

To add, edit or remove a portlet the basic steps are as follows:

If you are working from a WAR archive:

- Once the archive has been unpacked, and making sure the servlet container is shutdown, edit the WEB-INF/conf/spp.xreg file. (See below for how to write a portlet definition).
- Restart the container when you are done.

If you are working from the source:

- Define your installation-specific list of available portlets in your localised src/localisation/<your_localisation>/portlets/spp/configuration/spp-portlets.xregfile. (See below for how to write a portlet definition).
- Check the other portlets/<portlet>/configuration/*.xreg files. These will also be included into your local spp.xreg file at deploy time. Overwrite them in your localisation directory if need be.
- The list of available portlets in your installation will be made up of all the ref (and instance) portlets in the *.xreg files.
- Once you are happy, deploy as usual. Your portlet definitions will appear in WEB-INF/conf/spp.xreg.

Writing a portlet definition

At present we use the Jetspeed portlet definition system. This basically involves defining abstract generic reference portlets, and then defining instances that a user can subscribe to. For example, an RSS newsfeed reading portlet might be the abstract reference portlet, an definition that pointed at a particular newsfeed source would be the instance.

Unfortunately there is no user interface to ease the process of adding a portlet at the moment. You need to edit some xml files.

At present we use the Velocity type abstract portlets. These link a name, and default parameters, to a classname, which means our ref portlet definitions don't need to include the classname parameter. We define SPP portlets that ref the Velocity portlets, and use the action parameter to point to the SPPPortlet class in question. See below for an example.

Jetspeed has several .xreg files installed by default:

portlets.xreg

abstract portlets. Overwritten by spp deploy to remove any non-abstract ones. Here's where the template RSS and Velocity portlets are defined.

admin.xreg

admin-specific portlets. Not available to non-admin users.

demo-portlets.xreg

example portlets. Overwritten, removing most of the examples.

In addition SPP portlets are defined in WEB-INF/conf/spp.xreg. You can alter these portlet definitions.

For the WAR version:

Edit the WEB-INF/conf/spp.xreg file mentioned above. Then restart tomcat.

For the source code version:

See the <spp-src>/portlets/<portlet-name>/configuration/* .xreg files. These files should just contain the <portlet-entry> XML fragments. On deployment they are combined into a spp.xreg file and placed in the WEB-INF/conf directory. Here's an example, an RSS feed.

Note how this is a ref type portlet, with a parent SPPNewsfeed. The SPPNewsfeed portlet (abstract) is defined in the <spp-src>/portlets/spp/configuration/core-portlets.xreg file.

```
<portlet-entry name="rdn_news" hidden="false" type="ref" parent="SPPNewsfeed" application="false">
```

```
  <meta-info>
```

```
    <title>RDN News</title>
```

```
    <description>The latest news at the Resource Discovery Network</description>
```

```
  </meta-info>
```

```
    <parameter name="url" value="http://www.rdn.ac.uk/news/channels/rdnrss.xml" hidden="true"/>
```

```
    <category>All Newsfeeds</category>
```

```
    <category>News</category>
```

```
</portlet-entry>
```

Notes:

All portlets must have a <media-type ref="html"/> entry, or they will not be available via the web interface. In the example above the portlet inherits this entry from the abstract SPPNewsfeed portlet.

Once you have finished editing, re-deploy as usual, and restart your container.

Service configuration

The services provided by the SPP portal can be configured in various ways.

Mail

The portal needs to be able to dispatch mail for various services, including exporting search results, alert messages and account registration. At present you need to configure the mailing options in two places:

Firstly:

For WAR version:

```
<tomcat_home>/webapps/spp/WEB-INF/spp/mailer.properties
```

You should shut down tomcat before editing this file.

For source code version:

```
<spp_src>/localisation/<your_localisation_name>/configuration/copy/WEB-INF/spp/mailer.properties
```

The mailhost property should be set to your local smtp provider. You should also set the values of the default-from and default-subject properties.

And secondly:

For WAR version:

```
<tomcat_home>/webapps/spp/WEB-INF/conf/JetspeedResources.properties
```

You should shut down tomcat before editing this file.

Edit these properties:

```
mail.server
```

```
mail.support
```

```
newuser.confirm.email.from
```

```
newuser.confirm.email.name
```

For source code version:

```
<spp_src>/localisation/<your_localisation_name>/configuration/patch/WEB-INF/conf/JetspeedResources.properties.ant
```

Changes here are best done using the patch system. Copy into your localisation the default patch file from here:

```
<spp_src>/configuration/patch/WEB-INF/conf/JetspeedResources.properties.ant
```

Then edit it to just contain the replaceregexp entries for:

```
mail.server
```

```
mail.support
```

```
newuser.confirm.email.from
```

newuser.confirm.email.name

Though leaving the parent project and target elements (and the latter's property entries).

Once these changes have been made make them live by:

For WAR version:

Restart tomcat.

For source code version:

Stop tomcat.

From <spp_src>:

> ant deploy

Restart tomcat.

Account Management

All the Account Management (AM) properties are available from the file:

WEB-INF/spp/accman.properties

User accounts are managed by an LDAP server, and user authentication can be carried out by this LDAP server (Local Authentication System - LAS), or by Athens (National Authentication System - NAS). In any local portal installation, the default values should work without the need for any changes. These defaults are:

- To use the central RDN test LDAP server.
- Not to use Athens.

If you want to use Athens you need to contact them first to register your server. [Note: more details to follow]

To install and manage your own LAS LDAP server see the section below.

Account Creation

By default users can register themselves via the portal. They will be sent a confirmation key to their supplied email address and asked to present it to confirm their account. These default settings can be altered. See WEB-INF/conf/JetspeedResources.conf for details. Note that if you are working from the source code you can patch this file - see the localisation section above.

Alternatively, the admin user can create, manage and delete user accounts once logged in via the User Browser.

AM Generic Properties

For your information, here are more explanatory details about the various AM configuration properties. [Note: move this to an appendix?]

The file referred to by the sppJAASConfigFile attribute (WEB-INF/spp/accman/sppJAASConfig by default) contains the JAAS Login Module class implementations of JAAS for LAS and NAS. New Login Modules should be added to this file.

AM Properties for NAS

The attribute `sppAthensConfigFilepoints` at a file that contains the addresses and ports where the different Athens Accounts and Stats Servers are running. The Athens agent installed chooses the servers according to their availability.

A single line entry of `SSOMode` ensures access only via SSO.

The file pointed to by `sppAthensBasicConfigFile` is similar to `sppAthensConfigFile`, but without the SSO tag. This file however is important for SOAP access and Junit test cases.

The file referred to by `sppAthensProtectedDSPs` contains the Athens protected resources. These entries are important for NAS in order to obtain the authorisation for access on-the-fly. However, they are also important for LAS in order to inform the users which resources are protected, and can't access.

#Format:

#localResourceID:resourceName:dspID:accessType

#(choose one):(ask Athens):(idem):(HTTP-SSO, HTTP, Z39.50)

AM Properties for LAS

LDAP Server

`sppLDAPHost`: The host where the LDAP Server is running.

`sppLDAPHost=ldap.portal.ac.uk`

`sppLDAPPort`: The port where the LDAP Server is listening. By default this port is not being used, as authentication, search and update operation are all made in a secure mode via SSL.

`sppLDAPPort=389`

`sppLDAPPort`: The secure port where the LDAP Server is listening.

`sppLDAPSecurePort=636`

`sppLDAPBaseDN`: The base platform under which all AM information is stored in the DIT. It consists on a Distinguished Name (DN) composed of several Domain Components (DC). DCs are usually synchronised with the DNS's entity organisation. All other references in the DIT are Relative to this DN (RDN).

`sppLDAPBaseDN=dc=portal,dc=ac,dc=uk`

`sppLDAPManagerRDN`: The manager in charge of the LDAP Server (rootdn). The manager defines the permissions over the DIT. These details should be provided in order to access the Directory. `sppLDAPManagerRDN=cn=Manager`

`sppLDAPManagerCredential`: The manager's password to access the Directory. `sppLDAPManagerCredential=secret`

SSL Protection

`sppSSLKeystore`: The key store with the LDAP Server trusted root certificate. `sppSSLKeystorePassword`: The password to access the key store.

User Entries

`sppLDAPUsersRDN`: The place where all users will be stored. `sppLDAPUserSRDN=ou=People,o=SPP`

sppLDAPUsersID: The identifier to uniquely locate a user. It is used for authentication. sppLDAPUsersID=uid

sppLDAPUsersTemplate: The user's entry used as a template when creating a new user. sppLDAPUsersTemplate=test

sppLDAPUsersUserName: The identifier to uniquely locate a user. It is used for searching. sppLDAPUsersUserName=uid

sppLDAPUsersFirstName: User's first name. sppLDAPUsersFirstName=cn

sppLDAPUsersLastName: User's last name. sppLDAPUsersLastName=sn

sppLDAPUsersPassword: User's password. According to the LDAP server configuration, this attribute is hidden. sppLDAPUsersPassword=userPassword

sppLDAPUsersEmail: User's email address. sppLDAPUsersEmail=mail

Role Entries

sppLDAPRolesRDN: The place where all roles will be stored. sppLDAPRolesRDN=ou=Roles,o=SPP

sppLDAPRolesID: The identifier to uniquely locate the roles. sppLDAPRolesID=cn

sppLDAPRolesRefID: The identifier to reference a user with a given role. sppLDAPRolesRefID=uniqueMember

sppLDAPRolesTemplate: The role's entry used as a template when creating a new role. sppLDAPRolesTemplate=user

Group Entries

sppLDAPGroupsRDN: The place where all groups will be stored. sppLDAPGroupsRDN=ou=Groups,o=SPP

sppLDAPGroupsID: The identifier to uniquely locate a group. sppLDAPGroupsID=cn

sppLDAPGroupsRefID: The identifier to reference a user with a given group. sppLDAPGroupsRefID=uniqueMember

sppLDAPGroupsTemplate: The group's entry used as a template when creating a new group. sppLDAPGroupsTemplate=user

Permission Entries

sppLDAPPermissionsRDN: The place where all permissions will be stored. sppLDAPPermissionsRDN=ou=Permissions,o=SPP

sppLDAPPermissionsID: The identifier to uniquely locate a permission. sppLDAPPermissionsID=cn

sppLDAPPermissionsRefID: The identifier to reference a role with a given permission. sppLDAPPermissionsRefID=uniqueMember

sppLDAPPermissionsTemplate: The permission's entry used as a template when creating a new permission.

sppLDAPPermissionsTemplate=view

AM SOAP Properties

sppUseSOAP: This property enables or disables SOAP. When enabled, AM is provided via SOAP

(currently only authentication). sppUseSOAP=false

sppDeployedSOAP: When SOAP is enabled, this property is used to access AM deployed as a Web Service. sppDeployedSOAP=http://localhost:8080/

Storage

The portal stores various items of data for a user between sessions. These may include saved records or searches, alert settings or layout information. By default these are saved in an in-memory SQL database that is started and stopped along with the rest of the portal.

If you are happy to use this then you can skip the rest of this section.

If, however, you would prefer to use a separate SQL database please follow the following steps. First edit this file:

For WAR version:

```
<tomcat_home>/webapps/spp/WEB-INF/spp/xsearch/userprof.xml
```

You should shut down tomcat before editing this file.

For source code version:

```
<spp_src>/localisation/<your_localisation_name>/configuration/copy/WEB-INF/spp/xsearch/userprof.xml
```

Edit the properties grove.driver.class and grove.uri with values appropriate for your database. Here's an example for Postgres:

```
grove.driver.class = org.postgresql.Driver  
grove.uri=jdbc:postgresql://<host>:<port>/<db-name>
```

Now you need to initialise the portal's database. Note that you only need to do this if you're using an external database.

For WAR version:

[documentation and scripts to be completed]

For source code version:

The deployment scripts can also do the initial creation and population of the portal storage database. Set these properties in your build.properties:

```
up.driver = <driver class (same as grove.driver.class above)>  
up.classpath = <path to driver library (e.g. for postgres:  
${jetspeed.webapp}/WEB-INF/lib/pg74.213.jdbc3.jar)>  
up.db-path = <path to db (same as grove.uri above)>  
up.db-user = <username>  
up.db-password = <password>
```

The user should have rights to create users and databases. Upon deployment this script is executed to create and populate the initial database:

```
<spp_src>/configuration/patch/WEB-INF/db/profiles.ant
```

[Note: at present slight modifications may need to be made to this script depending on the db being targeted. See the script for details. Also, once the database and users have been set up you may find

it convenient to comment out the transaction content, so that subsequent deploys don't try to re-create them. This issue will be addressed in a subsequent release.]

Once you have completed the setup, make your changes live by:

For WAR version:

Restart tomcat.

For source code version:

Stop tomcat.

From <spp_src>:

> ant deploy

Restart tomcat.

Searching

The portal comes configured with a selection of Z39.50 targets and newsfeeds for searching. You can choose to make some or all of these available in your portal, or decide to add your own.

Selecting targets to display

To determine which targets should be available to your users you need to edit one configuration file: modes.xml.

For WAR version:

<tomcat_home>/webapps/spp/WEB-INF/spp/xsearch/modes.xml

You should shut down tomcat before editing this file.

For source code version:

<spp_src>/localisation/<your_localisation_name>/configuration/copy/WEB-INF/spp/xsearch/modes.xml

This file looks like this:

```
<xsearch-modes>
  <mode>
    <name>xsearch</name>
    <collections>
      <collection_id>ABI</collection_id>
      <collection_id highlight="true">ageline</collection_id>
      ...
    </collections>
  </mode>
</xsearch-modes>
<global_search_landscapes>
  <search_landscape>
```

```

        <name>SOSIG</name>
        <collection_id>SOSIG</collection_id>
    </search_landscape>
    <search_landscape>
        <name>ZETOC and COPAC</name>
        <collection_id>ZETOC</collection_id>
        <collection_id>COPAC</collection_id>
    </search_landscape>
</global_search_landscapes>
</mode>
<mode>
    ...
</mode>
</xsearch-modes>

```

Each searching portlet has a mode, with a unique name. This name is defined as a parameter in its portlet definition. The defaults are 'xsearch', 'newsfeed' and 'ejournal' (but you can define your own - see below for details). Each searching portlet will get its target list from the appropriate mode section of the modes.xml file. Make a collection available by including its collection_id as shown above. This collection_id must match an instance_dn value (apologies for the labelling discrepancy - this will be fixed in a subsequent version) as listed in repositories.xml, which can be found here:

For WAR version:

```
<tomcat_home>/webapps/spp/WEB-INF/spp/xsearch/repositories.xml
```

For source code version:

```
<spp_src>/configuration/copy/WEB-INF/spp/xsearch/repositories.xml
```

Highlighted Collections

If a collection_id element has the optional attribute highlight set to true the collection will be displayed to the user in a list of highlighted collections. When constructing a search the user can then switch between viewing all collections or just those highlighted. If no collections have the highlight attribute set the option to view highlighted collections will not be displayed to the user.

Global Search Landscapes

Search landscapes define groups of collections that the user can search with one button click. An authenticated user can define their own landscapes, but there is also the facility for defining portal-wide, or global, landscapes, available to all users. The global_search_landscapes section contains these search landscape definitions. Each landscape should have a name entry, unique to its parent mode, and one or more collection_id entries.

Once this file has been edited, save it and make it live by:

For WAR version:

Restart tomcat.

For source code version:

Stop tomcat.

From <spp_src>:

> ant deploy

Restart tomcat.

Adding new targets

To add a new target to the list of those your portal knows about you need to make changes to a few configuration files. There are a few slightly different procedures depending on the type of target you are adding.

Newsfeeds

To make a newsfeed available for searching it must be first added as a newsfeed portlet. See below for details on how to do this. Then edit this file:

For WAR version:

```
<tomcat_home>/webapps/spp/WEB-INF/spp/xsearch/repositories.xml
```

You should shut down tomcat before editing this file.

For source code version:

```
<spp_src>/configuration/copy/WEB-INF/spp/xsearch/repositories.xml
```

For each newsfeed you want to make searchable add entries like these:

```
<Collection collection_dn="news_id" collection_name="Newsfeed Name"/>
```

```
<Instance instance_dn="news_id" collection_dn="news_id" repository_dn="RSS" local_name="news_id" />
```

Replace news_id with your newsfeed's unique name, as defined in the attribute name, in the element portlet-entry in the newsfeed's portlet definition (see below).

Make sure the newsfeed also has an entry in the appropriate mode in your modes.xml file (see above).

Now save these files and make them live by:

For WAR version:

Restart tomcat.

For source code version:

Stop tomcat.

From <spp_src>:

> ant deploy

Restart tomcat.

Z39.50

Add target details

Edit WEB-INF/spp/xsearch/repositories.xml to add the required details about the new target. You need to add Repository, Collection and Instance details. Here's an example:

```
<Repository repository_dn="ZETOC" name="ZETOC Z39.50 Server" type="Z3950"
can_multiplex_sessions="no" >

<RepositoryProperty name="ServiceHost" value="zetoc.mimas.ac.uk" />

    <RepositoryProperty name="ServicePort" value="2121" />

    <RepositoryProperty name="service_short_name" value="ZETOC" />

    <RepositoryProperty name="default_record_syntax" value="grs-1" />

<RepositoryProperty name="default_element_set_name" value="f" />

<RepositoryProperty name="full_element_set_name" value="f" />

</Repository>

<Collection collection_dn="ZETOC" collection_name="ZETOC"/>

<Instance instance_dn="ZETOC" collection_dn="ZETOC" repository_dn="ZETOC" local_name="zetoc" />
```

Notes:

Collection attribute collection_name is the name displayed to the user.

Instance attribute local_name is the target's local database name.

Obviously one Repository can contain several databases, each should have an Instance entry.

For the source code version, you can edit a localised copy of repositories.xml, or use the patch system.

Add a mapping stylesheet

Results from this target must be mapped to internal spp fields for display purposes. This is done using xsl stylesheets

Once you have decided on a format for record retrieval for this target, write a mapping file and put it in WEB-INF/spp/xsearch/mappings/. Use the files there as examples. See the README there for more details. You may well need to use a Z39.50 client to test the connection and the record format.

Register your new file by editing WEB-INF/spp/xsearch/SchemaMappings.xml

e.g.

```
<mapping from="ZETOC_grs-1" to="rec_full" sheet="ZETOC_Gils_To_REC_full.xsl"/>
```

Notes:

The attribute from should be Instance_dn_record-format. The exact contents of record-format can depend on the server. You can use a test client to retrieve a raw record.

The attribute to should be rec_full.

The attribute sheet should be your new xsl file.

Make the target available

Edit your WEB-INF/spp/xsearch.modes.xml file to include the new target. Details for this procedure are above.

OAI

[advanced procedure - documentation to be completed]

Alerting

Alerting allows users to be sent a weekly email containing new items that are related to their subject interests. There are various steps to take in order to tailor this service to your site's needs.

Alerting Sources

Firstly you need to have some sources of subject-classified, recent data. While these will be relaxed in future, at present there are two restrictions on these sources. They must:

Be an RSS feed that contains dc:subject information for its items. Here's an example:

<http://www.sosig.ac.uk/roads/cgi-bin/new2rss.pl>

Only return items that are new, i.e. that have been added within the alerting service's execution frequency. E.g. if the service's frequency is an alert every 7 days, the feed should only return items added in the previous 7 days.

This newsfeed should be added to the portal so that it is searchable (see above). Note that it does not have to be made visible for user searching.

To include this source in the portal's alerting sources edit this file:

For WAR version:

```
<tomcat_home>/webapps/spp/WEB-INF/spp/alerting-sources.properties
```

You should shut down tomcat before editing this file.

For source code version:

```
<spp_src>/localisation/<your_localisation_name>/configuration/copy/WEB-INF/spp/alerting-sources.properties
```

Add an entry in this format:

```
newsfeed_id=Newsfeed title
```

Save your changes.

Now you need to move on to the next step - configuring your taxonomy.

Subject Taxonomy

In order to match items to user's subject interests the portal must have a subject taxonomy. This defines the subjects the portal knows about and their relationships. To configure your portal's taxonomy edit this file:

For WAR version:

```
<tomcat_home>/webapps/spp/WEB-INF/spp/taxonomy.properties
```

You should shut down tomcat before editing this file.

For source code version:

```
<spp_src>/localisation/<your_localisation_name>/configuration/copy/WEB-INF/spp/taxonomy.properties
```

Entries in this file are in this format

```
taxon.name.<taxon_id>=<taxon_name>
```

```
taxon.parent.<taxon_id>=<parent_taxon_id>
```

e.g.

```
taxon.name.1=Philosophy
```

```
taxon.parent.1=
```

```
taxon.name.11=Metaphysics
```

```
taxon.parent.11=1
```

The taxon_ids must be the same as the subject ids used in the dc:subject fields in the source data.

If a subject has no parent (i.e. it is at the top of the subject hierarchy) its taxon.parententry should be left blank.

Save this file and make it live by:

For WAR version:

Restart tomcat.

For source code version:

```
Stop tomcat.
```

```
From <spp_src>:
```

```
> ant deploy
```

```
Restart tomcat.
```

Mail options

For the alerting service to work the portal must be able to send out mail. See the mail configuration section above for details.

The alerting service's mailing options should be set in this file:

For WAR version:

```
<tomcat_home>/webapps/spp/WEB-INF/spp/alerting.properties
```

You should shut down tomcat before editing this file.

For source code version:

```
<spp_src>/localisation/<your_localisation_name>/configuration/copy/WEB-INF/spp/alerting.properties
```

Here, you can set the frequency of alerts, as well the subject and some text for the alert messages.

Make these changes live in the usual way:

For WAR version:

Restart tomcat.

For source code version:

Stop tomcat.

From <spp_src>:

> ant deploy

Restart tomcat.

Newsfeeds

To add a new newsfeed to your portal a new portlet definition needs to be added to the portlet definition registry. The location of the registry file depends on which distribution you are working with:

For WAR version:

<tomcat_home>/webapps/spp/WEB-INF/conf/spp.xreg

You should shut down tomcat before editing this file.

For source code version:

<spp_src>/localisation/<your_localisation_name>/portlets/newsfeed/configuration/newsfeed.xreg

Now add a new portlet definition for your newsfeed. It will look like this:

```
<portlet-entry name="rdn_news" hidden="false" type="ref" parent="SPPNewsfeed" application="false">
```

```
  <meta-info>
```

```
    <title>RDN News</title>
```

```
    <description>The latest news at the Resource Discovery Network</description>
```

```
  </meta-info>
```

```
    <parameter name="url" value="http://www.rdn.ac.uk/news/channels/rdnrss.xml" hidden="true"/>
```

```
    <category>All Newsfeeds</category>
```

```
    <category>News</category>
```

```
</portlet-entry>
```

Obviously you should replace the relevant values with those appropriate to your newsfeed. You can define your own categories too. Save the file. Now, to make this portlet live:

For WAR version:

Restart tomcat.

For source code version:

Stop tomcat.

From <spp_src>:

> ant deploy Restart tomcat.

The portlet will now be available for users to include in their personal pages. It will be listed under the categories you specified.

To include this new newsfeed in the list of newsfeeds that are searchable see the Searching section above.

To include it in a default portal layout (e.g. the guest home page) see the PSML section above.

Chapter 4. Running the Portal

Once the portal is configured and installed in your servlet container, you can launch the application. This is handled by the web servlet container. An example is tomcat, which provides scripts for the job:

```
> tomcat/bin/startup.sh
```

```
> tomcat/bin/shutdown.sh
```

Note that the user running the web servlet container process must have write access to the spp/WEB-INF directory.

At present startup can take a little while as the portal application gathers user and newsfeed data, and initialises the various services. Any messages will be logged to the standard output for your container, and to the log files found in WEB-INF/log/.

Once the application is running it can be accessed via a browser at the appropriate URL. For example, a default installation to tomcat, with the browser running on the same machine, could be accessed at this URL.

```
http://localhost:8080/spp/
```

Chapter 5. Upgrading

In order to upgrade to a new release of the SPP without losing data you should follow the steps detailed below. At present a brief period of unavailability of the service is unavoidable.

An upgrade script is supplied. See WEB-INF/upgrade.sh.

[Note: to be completed]

Chapter 6. Extending the SPP

Creating an SPP portlet

You can extend the functionality of the portal by adding your own portlets. To do this you will need some Java coding knowledge and will have to have downloaded the source code (as detailed above). We would much appreciate any contributions of work you think would be of use to others. Please contact the source code maintainers.

File locations

First, where to put your code?

The easiest method is to follow the standard SPP source code layout:

src/core/uk/ac/portal/spp/<name>/for the core functionality

src/portlets/<name>/for the related portlet code. Under this should be...

configuration/for .xreg definition files

java/uk/ac/portal/spp/actions/<name>/for the actual portlet code

velocity/<name>/for the portlet velocity files

Backend functionality

Write all your functionality and make it available via service facades using the stock startup mechanism. This is important: the job of a portlet should simply be to choreograph the actions of backend services (which are then testable separately from the UI, with any luck).

The Portlet class

Now, put the SPP Portlet boiler-plate code in place:

```
package uk.ac.portal.spp.actions.@@YOUR-PACKAGE@@;

import org.apache.velocity.context.Context;

import uk.ac.portal.spp.common.Log;

import uk.ac.portal.spp.common.portlet.SPPPortletState;

import uk.ac.portal.spp.common.portlet.SPPRuntimeData;

import uk.ac.portal.spp.common.portlet.SPPPortletConfig;

import uk.ac.portal.spp.glue.portlet.BaseErrorHandlingPortlet;

@@YOUR PACKAGE-SPECIFIC IMPORTS GO HERE@@

public class @@YOUR-PORTLET@@ extends BaseErrorHandlingPortlet

{
```

The BaseErrorHandlingPortlet is an implementation of some error-handling methods on top of an implementation of uk.ac.portal.spp.common.portlet.SPPPortlet. Such a base class provides facilities for you to use, as well as having some conventions for you to follow.

Views

From a UI point of view, your portlet might have a number of "views". (These are not *necessarily* in one-to-one correspondence with velocity templates, but may be.) They represent the current "page" your portlet should display. The current "view" is called the "state switch".

State

In addition, your portlet has a "session" associated with it, called its "state". This is where you can store and fetch values that will persist between HTTP requests. You can set your portlet up initially by implementing the following method:

```
public void initialiseState(
    SPPPortletConfig conf,
    SPPRuntimeData in,
    SPPPortletState state)
{
    // To get configuration information
    String xregConfigParameter = (String)
        conf.getAttribute("@@YOUR XREG PARAM NAME@@");
    // We may also wish to get details about the user,
    // the user profile, etc.
    // the "in" parameter supplies these.
    state.setAttribute("atname", xregConfigParameter);
    setStateSwitch(state, "MainView");
}
```

Producing a view

To produce a particular view, you have to do three things:

- write the code to populate a velocity context,
- add a call to choose the correct template, and
- write the velocity template itself.

You put this code into a method whose name is derived from the state switch: the following method would be called if the state switch had been set to "MainView".

```
public void buildNormalMainView(SPPRuntimeData in,
    Context cx,
    SPPPortletState state)
{
    cx.put("velocity_variable_name",
```

```

state.getAttribute("attname");

setTemplate(in, cx, "mypackage/mytemplate");

}

```

When the portal wants to draw your portlet, it examines the stateswitch, calls the right buildNormalXXX method, which constructs a velocity context (making variables available to the velocity template) and finally chooses which template to draw. Your velocity file itself can display the values of its supplied parameters (see example templates under the alerting portlet).

User interactions

A web application is also about user interaction. How do user operations get back to the portlet? Here's a sample velocity snippet:

```

<form action="$spplink.event("RenameWidget")" method="POST">
<p> The widget is called

  <input type="hidden" name="id" value="$widget.id" />

  <input type="text" name="name" value="$widget.name" />

  <a href="$spplink.event("DeleteWidget").add("id",$widget.id)">

    [delete this]

  </a>

  <input type="submit" value="Rename" />

</p>
</form>

```

OK, the snippet assumes our buildNormalXXX method has put an object called "widget" into the context, with methods getId() and getName(). The template can trigger two "events", which are the last piece of the lifecycle of a portlet. As you can see, events are generated using the "spplink" object. If the user submits the form, the following method in your code is called:

```

public void eventRenameWidget(SPPRuntimeData in,
    SPPPortletState state)
{
    String id = in.get("id");
    String name = in.get("name");
    /* Rename the widget */
    ...
    setStateSwitch("DisplayWidget");
}

```

If you click on the delete link, instead the eventDeleteWidget method will be called.

Convenience methods

The SPP Portlet supplies a number of convenience methods:

`void setTemplate(SPPRuntimeData, Context, String)`we've already seen,

`void setStateSwitch(SPPPortletState, String)`we've already seen

`String getStateSwitch(SPPPortletState)`returns the current stateswitch

`Object getService(SPPRuntimeData in, String name)`

This last call can be used to get a facade instance. For example, my missing widget renaming code above might look like this:

```
/* rename the widget */  
  
WidgetFacade wf = (WidgetFacade) getService(in, "spp.wf");  
  
wf.renameWidget(id, name);
```

Error handling

Finally, the error-handling. This may need a little sprucing up, but currently, the `BaseErrorHandlingPortlet` provides additional methods for getting responses to a user. These methods are as follows:

```
void addNotice("some notice", SPPPortletState state);
```

with analogous `addNoticeInfo`, `addNoticeWarn`, and `addNoticeError`.

If you want to utilise these methods, you can cause a list of current pending notices to appear in your velocity template by including the line

```
#parse("portlets/html/spp/error-handling.vm")
```

which does the magic.

Chapter 7. Setting up an SPP LDAP server

Here are the tasks necessary for installing, configuring and running the SPP Local AMS (LDAP/SSL).

[Note that if you are using the central RDN LDAP server you don't need to run your own server - you can ignore this section.]

These instructions are based on `openldap` (<http://www.openldap.org/>), but other LDAP (or even X.500) based systems should have similar configuration with equivalent files and parameters.

For keys and Digital Certificates generation and manipulation use `openssl` (<http://www.openssl.org/>) and Java `keytool` (Key and Certificate Management

Tool - <http://java.sun.com/products/jdk/1.2/docs/tooldocs/solaris/keytool.html>).

LDAP Server Installation and Configuration

Download and install `openldap`.

Note: remember to build your `openldap` with the `-enable-ldbm` switch to allow support for the `ldbm` backend (as specified in the config file below).

Now set up the server configuration.

For the WAR version, download example LDAP configuration files from the same location.

For the source code version example files are at the directory:

```
<spp-src>/configuration/openLDAPConf
```

Make local copies of the files below based on the examples. For the most part they can be left, but check them to include path information correct for your environment. Here are some explanatory details:

sppSlapd.conf

This is the server configuration file.

Schemas

Schemas to be used in the organisation of the SPP Directory Tree Information (DIT).

```
include ../../core.schema
```

```
... ../../...
```

```
include ../../sppLDAP.schema
```

```
include ../../edupersonLDAP.schema
```

ACLs

The default Access Control Lists (ACL). By default the access is just for reading.

```
defaultaccess read
```

[Note: This line can produce an warning on `slapd` startup:

/usr/local/projects/spp/openldap/sppSlapd.conf: line 60: unknown directive "defaultaccess" outside backend info and database definitions (ignored)]

Access to the User's password (userPassword) is exclusively for authentication.

access to attr=userPassword

by * auth

Access to the User's entry is open to everyone for reading (apart from the userPassword) and only the User can write.

access to dn.subtree="ou=People,o=SPP,dc=portal,dc=ac,dc=uk"

by dn.children="ou=People,o=SPP,dc=portal,dc=ac,dc=uk" write

by * read

The Directory Manager can access to all entries for read and write.

access to *

by dn="cn=Manager,dc=portal,dc=ac,dc=uk" write

by self write

by * read

Base Definitions

Database type, Base DN, Manager and its Password definitions.

database ldbm

[Note: This is why you switched -enable-ldbm on when building openldap.]

suffix "dc=portal,dc=ac,dc=uk"

rootdn "cn=Manager,dc=portal,dc=ac,dc=uk"

rootpw secret

TLS/SSL Configuration

TLSCertificateFile ../../ldap.cert

TLSCertificateKeyFile ../../ldap.key

TLSCACertificateFile ../../ca.cert

sppEntries.ldif

This is the information in the LDIF format to feed the Directory initially. Here is an example of the top entry. You can edit this file to change your initial set of users.

dn: dc=portal,dc=ac,dc=uk

#-----

#objectClasses

#-----

objectclass: top

```
objectclass: dcObject
#-----
#class dcObject
#-----
dc: portal
description: SPP Base Entry
...
```

sppLDAP.schema

Contains SPP object classes and attribute type definitions for the SPP LDAP Schema.

```
# 1.3.6.1.4.1.11023 = toplevel OID for this work
#       .3 = delegated OID
#       .3.1 = SPP
#       .3.1.1 = sppObject
#       .3.1.1.1 = attributeTypes
#       .3.1.1.2 = objectClasses
#       .3.1.1.3 = syntax (probably never used)
#       .3.1.2 = sppPerson
#       .3.1.2.1 = attributeTypes
#       .3.1.2.2 = objectClasses
#       .3.1.2.3 = syntax (probably never used)
...
```

edupersonLDAP.schema

Contains eduPerson object classes and attribute type definitions for the eduPerson LDAP Schema.

```
# 1.3.6.1.4.1.5923 is the toplevel OID for this work
#       .1 = MACE related work
#       .1.1 = eduPerson
#       .1.1.1 = attributes
#       .1.1.2 = objectclass
#       .1.1.3 = syntax (probably never used)
#       .1.2 = eduOrg
#       .1.2.1 = attributes
#       .1.2.2 = objectclass
```

```
# .1.2.3 = syntax (probably never used)
```

```
...
```

TLS/SSL CA and Certificates Generation

In order to provide TLS/SSL secure access and not having to buy third-party certificates, you can manage your own Certificate Authority (CA) and generate self-signed Digital X.509 Certificates.

Follow these instructions to generate a new CA and the new certificates (e.g. LDAP Server Certificate) valid just for one year. It's worth noting that the CA could safely be valid for much more time.

Generate Certification Authority Keys

```
> openssl genrsa -des3 -out ca.key 2048
```

Generate Certification Authority Certificate

```
> openssl req -new -x509 -days 365 -key ca.key -out ca.cert
```

Note: this step can ask for info as below in the "Request a LDAP Server Certificate" step below.

Generate a Key Pair (Private/Public Keys)

```
> openssl genrsa -out ldap.key 1024
```

Note: If you share your machine protect the keys with extra privileges or store it in a secure place.

Request a LDAP Server Certificate

```
> openssl req -new -key ldap.key -out ldap.csr
```

Note: The info you may want to put:

UK,

..

..

RDN/JISC,

SPP,

****YOUR-MACHINE DOMAIN NAME**** (e.g. ldap.portal.ac.uk),

****Your-EMAIL**** (e.g. spp-dev@portal.ac.uk).

Note: You may also be asked for extra info:

Please enter the following 'extra' attributes

to be sent with your certificate request

A challenge password []:

An optional company name []:

You can leave these blank.

Sign LDAP Server Certificate with the CA

```
> openssl x509 -req -in ldap.csr -out ldap.cert
```

```
-CA ca.cert -CAkey ca.key -CAcreateserial  
-days 365
```

Verify the LDAP Server Certificate

```
> openssl x509 -in ldap.cert -text -noout
```

Import the LDAP Server Certificate into a Java Key Store

```
> keytool -import -file ldap.cert  
-keystore sslkey.keystore  
-alias "type=r.name=sslkey"
```

Note: This step sometimes went wrong when I was doing it, with an error:

```
keytool error: java.security.SignatureException: Signature does not match.
```

Re-doing the ldap.key and ldap.cert steps with different data seemed to fix it.

Put the ldap.cert, ldap.key and ca.cert files created in the locations pointed to by your slapd.conf file, as above.

LDAP Server Operation

In order to access the information in the Directory (e.g. search), start it and add the initial information.

To start LDAP Server in a secure mode.

```
> slapd -4 -d 128 -s 3 -h "ldaps:// ldap:// " -f /.../.../slapd.conf
```

To add the entries in the directory.

```
> ldapadd -x -D "cn=Manager,dc=portal,dc=ac,dc=uk"  
-w secret -f /.../.../sppEntries.ldif
```

To do a test search for all entries.

```
> ldapsearch -x -b 'dc=portal,dc=ac,dc=uk'  
'(objectclass=*)'
```

SPP configuration

Edit these lines of your /WEB-INF/spp/accman.properties to point to your sslkey.keystore, with the password you set when creating it:

```
sppSSLKeystore=/.../.../sslkey.keystore  
sppSSLKeystorePassword=<password>
```

Change the line:

```
sppLDAPHost=
```

to point to the machine hosting your LDAP server.

You will need to restart your web application to pick up these edits.

Your SPP installation should now be able to use your LDAP server for user authentication.

Chapter 8. Known problems (and some solutions)

Ant

There may be a small problem with ant 1.6 at present. If so, we'd recommend using 1.5.1.

Empty newsfeeds

With some combinations of JDK 1.4 and Tomcat 4 you may see no content in your newsfeed portlets, and see this message in your log file:

```
ERROR NewsFeed - error transforming feed: org.apache.xml.utils.WrappedRuntimeException: The output format must have a '{http://xml.apache.org/xalan}content-handler' property!
```

To solve this you need to copy the

```
<tomcat>/webapps/<spp>/WEB-INF/lib/xalan-2.3.1.jar file to <tomcat>/common/endorsed.
```

Working with a firewall

Certain ports must be open for some portal services to function if your service is running behind a firewall. See the file README.ports in the distribution directory.