

# eprints.org Archive Software Installation Document

Robert Tansley

November 8, 2000

## 1 Introduction

This document explains how to set up and install an *eprints.org* archive on a UNIX system. The instructions should allow users to install the archive on any UNIX implementation, such as Solaris, SGI IRIX or Linux. In addition to generic instructions, specific instructions are also given on how to install the system on a Redhat 6.x Linux system, the recommended platform on which to run the *eprints.org* software.

Although this document is rather large, much of it is to do with configuring various aspects of the system. If you are happy with the default metadata sets and site look and feel, you won't have to work through all of it.

### 1.1 Beta Release Note

Many of the procedures in this installation document might change for the main release. If and when you install a production server, you should obtain the relevant installation document for that version.

It is hoped that in later versions, much of the installation process will be automated, reducing the effort involved in this process greatly.

The Open Archives Initiative protocol is about to change. At the moment, the core code only supports the old Santa Fe convention protocol (the Dienst subset) and OAMS, the Open Archives Metadata Set. However, the site-specific API in `SiteRoutines.pm` is hopefully stable which means you just have to upgrade the core code to support new versions of the protocol.

### 1.2 Upgrading from Beta-1

If you are upgrading from version beta-1, and you want to port across code and/or configuration options, it might be easiest to make a backup of the old version, install the new version afresh and copy the changes across. Here are some points you should be aware of:

- The prerequisite software is largely the same, but you'll need to install `Unicode-String-2.xx` from CPAN.
- The directory structure and method of linking into Perl have changed slightly. You should remove the `EPrints` and `EPrintSite` symlinks from

<i>Package</i>	<i>Version</i>	<i>Where from</i>
tar & gunzip		come with most UNIXes
unzip	5.x	RedHat packages: tar & gzip <a href="ftp://ftp.freesoftware.com/pub/infozip/">ftp://ftp.freesoftware.com/pub/infozip/</a>
wget	1.5	RedHat packages: unzip <a href="http://www.gnu.org/software/wget/wget.html">http://www.gnu.org/software/wget/wget.html</a>
perl	5.005	RedHat packages: wget <a href="http://www.cpan.org/">http://www.cpan.org/</a>
apache	1.3.9	RedHat packages: perl <a href="http://www.apache.org/">http://www.apache.org/</a>
mod_perl	1.21	RedHat packages: apache & apache-devel <a href="http://perl.apache.org/">http://perl.apache.org/</a>
MySQL	3.22.x	RedHat packages: mod_perl <a href="http://www.mysql.org/">http://www.mysql.org/</a>
sendmail <sup>1</sup>	8.9	Not on RedHat CD comes with most UNIXes RedHat packages: sendmail

Table 1: Prerequisite Software Packages

the `/usr/lib/perl5/site_perl` directory (or equivalent). Note also that you have to define the `PERL5LIB` variable in various places (the `crontab` and the `.procmailrc`) so that the libraries are picked up.

- The `referencetext` metadata field has been added. If you want to keep the data you have and add this field, you can add the field manually to the `inbox`, `buffer` and `archive` tables in MySQL, and add the `referencetext` field to `cfg/metadata.eprint-types` and `cfg/metadata.eprint-fields`. This also applies to the `expertise` field in `metadata.user` and the `users` table.
- In anticipation of impending changes to the Open Archives protocol, the core code/site code interface has changed slightly. In `SiteRoutines.pm`, instead of `eprint_get_oams`, there are now two open archives functions, `oai_list_metadata_formats` and `oai_get_eprint_metadata`. At the moment, however, the core software only supports OAMS, so you'll need to export that.
- Feel free to send any questions you have to `support@eprints.org`.

## 2 Prerequisites

The *eprints.org* software makes use of a number of other pieces of software, which must all be installed on the server machine in order for it to function properly. Table 1 describes the software packages you need, together with the minimum version you should try and get. The *eprints.org* software may work with versions earlier than those indicated but we can't guarantee it.

---

<sup>1</sup>or similar mail transport system

Many of them may be already installed on your system. Also shown is which RedHat package you should install (from the RedHat installer) to install that package on the machine. Note that while MySQL doesn't appear on the RedHat 6.2 CD, suitable RPMs are available from the MySQL Web site.

## 2.1 Special Notes on Apache and mod\_perl

Apache and mod\_perl need special attention, since you can configure a myriad of parameters when you compile and install them. The software requires that a few of these are configured in a certain way.

Fortunately, Apache RPM supplied with RedHat Linux 6.2 has a suitable compilation configuration. The mod\_perl distributed with RedHat 6.2 is broken, and should be replaced with `mod_perl-1.23-1.i386.rpm`, available from the <http://www.eprints.org> Web site.

Apache should be installed with the following modules compiled in (or made available as a .so) and enabled:

- `rewrite_module`
- `auth_module`

mod\_perl should be compiled with at least the following hooks enabled:

- `PERL_AUTHEN`
- `PERL_AUTHZ`

You can just compile it with `ALL_HOOKS` if you like.

## 2.2 Perl Modules

The following Perl modules are used by the *eprints.org* archive. They are all available from the CPAN Web site (<http://www.cpan.org/>). The version numbers are the minimum versions you should try and get. You may find that some are already installed with your Perl distribution.

- CGI 2.6*x*
- POSIX
- Data-Dumper 2.*xxx*
- DBI 1.1*x*
- Mysql-MySQL-modules 1.2*xxx* (*don't need mysql-perl emulation*)
- Filesys-DiskSpace 0.05
- MIME-Base-64 2.11
- URI-1.06
- XML-Writer-0.4
- ApacheDBI-0.87
- Unicode-String-2.*xx*

## 2.3 E-Mail

Some operations are performed by e-mail. An *eprints.org* archive needs to be able to send e-mails (typically using the `sendmail` package), and also to receive e-mails and process them using a script.

In other words, the *eprints.org* software needs an e-mail account, for example `eprints@archive.foo.org`, and every time an e-mail arrives at that account, an *eprints.org* script needs to be run and given the contents of the e-mail. This e-mail account is referred to as the *automatic administration account*. How this account is set up will probably vary between institutions. You may want a mail server on the local machine, or NFS mount a mail spool directory, or have another mail server machine `rsh` the relevant script. What needs to happen with this is explained later on; for now, just be aware that the software needs a mail account it can send from and automatically process incoming mails.

You'll also need a regular administration e-mail account for users needing to contact a person.

Once all of this software is installed, we're ready to proceed with the installation of the *eprints.org* archive software.

## 3 Installing the *eprints.org* Software

It's probably a good idea to set up an "eprints" account on the server machine. It can be a user's account, but it's strongly recommended that the server is not run as root.

### 3.1 Unpacking the Distribution

Choose whereabouts on your system the *eprints.org* software should reside. For the remainder of this document it is assumed that the software distribution is unpacked as `/opt/eprints`, and so the cgi directory is `/opt/eprints/cgi` etc. The eprints user should own all of the files in this directory.

The distribution is a tarred and gzipped file, and the usual way to unpack these is with:

```
> gunzip -c eprints-version.tar.gz | tar xvf -
```

If your perl executable isn't installed at `/usr/bin/perl`, you'll need to change the first line of each of the scripts in `eprints/bin`<sup>2</sup>.

### 3.2 Configuring *eprints.org*

Configuration information is largely held in two directories: `eprints/cfg` and `eprints/perl_lib/EPrintSite`. Most of it is set up with reasonable defaults, but you will need to edit `eprints/perl_lib/EPrintSite/SiteInfo.pm` before anything will work.

*eprints.org* is a highly configurable piece of software; the various ways in which it can be configured are described in more detail in section 4. This

---

<sup>2</sup>This, and other file system path-related issues, should be handled more automatically by an installation script in later versions.

section describes the technical aspects of the configuration required to get the installation to work.

`SiteInfo.pm` is fairly well commented, so you should just be able to go down that file and alter any values you need to. In the minimal instance, you'll need to change the following values:

- `$EPrintSite::SiteInfo::automail` - Don't forget to backslash the `@` if you use double quotes!
- `$EPrintSite::SiteInfo::admin`
- `$EPrintSite::SiteInfo::local_root` - the `bin`, `cfg`, `cgi` etc. directories must be subdirectories of this
- `$EPrintSite::SiteInfo::host` - Fully qualified hostname, for example `archive.foo.org`
- `$EPrintSite::SiteInfo::eprint_id_stem` - Prepended to all document record ID's in the system
- `$EPrintSite::SiteInfo::password` - The password for your MySQL database (remember it for later)
- `$EPrintSite::SiteInfo::archive_identifier` - Your archive's Open Archives identifier

If your server is running on a non-standard port (i.e. other than port 80), you will need to add the relevant port numbers to the following variables:

- `$EPrintSite::SiteInfo::server_static`
- `$EPrintSite::SiteInfo::server_perl`

For example, if you are running the archive on port 8080, you should change `$EPrintSite::SiteInfo::server_static` to:

`http://$EPrintSite::SiteInfo::host:8080`

and apply a similar change to `$EPrintSite::SiteInfo::server_perl`.

You should also verify that the paths to various executables are correct in the command line values, i.e.

- `%EPrintSite::SiteInfo::archive_extraction_commands`
- `$EPrintSite::SiteInfo::wget_command`
- `$EPrintSite::SiteInfo::sendmail`

For security, you should make sure that `SiteInfo.pm` is readable only by the eprints user.

At this point you should decide on the types of eprints you want to store in the archive, the metadata you want to store with each eprints and your initial subject hierarchy. These are described in section 4.2.

### 3.3 Setting up the MySQL database

While the *eprints.org* software handles most database operations transparently, some initial setting up is required.

Usually, after installing MySQL, you'll need to set a root password, using something like:

```
> mysqladmin -u root password "password"
```

Note that **password** here should *not* be the password you have put in *Siteinfo.pm*. From a security viewpoint, it's probably best to remove all non-root users and all non-localhost access from the privilege system.

Now create the archive database, from the **mysql** monitor tool:

```
mysql> create database eprints;
```

Then set up the access privileges:

```
mysql> GRANT SELECT, INSERT, UPDATE, DELETE, CREATE ON eprints.*  
TO eprints@localhost IDENTIFIED BY "password";
```

**password** in this case should be the password you put in *SiteInfo.pm*. The MySQL database is now ready.

### 3.4 Configuring Apache

Apache needs to be pointed at the *eprints.org* software and the password system set up.

Edit your Apache configuration, **httpd.conf** (found in */etc/httpd/conf* on RedHat systems.) On some earlier versions of Apache, the configuration is split into **httpd.conf**, **access.conf** and **srn.conf**. In these cases, the **<Directory>** entries should go in **access.conf**, the **Alias** entries in **srn.conf**, and everything else in **httpd.conf**.

Firstly, turn on the authentication using MySQL and Perl DBI by adding these lines (after **mod\_perl** is included:)

```
PerlModule Apache::DBI  
PerlModule Apache::AuthDBI
```

The Web server should ideally run as the *eprints* user, so change the relevant lines (though you may wish to use group *nobody*). It's possible to run the server as another user (for example, the default *nobody*), but this does cause problems; see section 4.1 if you want to do this.

```
User eprints  
Group eprints
```

The library files need to be accessible within Perl:

```
PerlSetEnv PERL5LIB /opt/eprints/perl_lib
```

Set the **DocumentRoot** to point at the HTML root:

```
DocumentRoot "/opt/eprints/html"
```

We need to be allowed to override the authorisation for this directory, e.g:

```
<Directory "/opt/eprints/html">
    Options Includes FollowSymLinks
    AllowOverride AuthConfig
    Order allow,deny
    Allow from all
</Directory>
```

Set up the location for mod\_perl scripts:

```
Alias /perl/ /opt/eprints/cgi/
<Directory /opt/eprints/cgi>
    SetHandler perl-script
    PerlHandler Apache::Registry
    PerlSendHeader Off
    Options +ExecCGI
    AllowOverride AuthConfig
</Directory>
```

Add engine to run the open archives interoperability software:

```
RewriteEngine on
RewriteRule ^/Dienst(.*) /opt/eprints/openarchives/Main/dienst.pl
<Directory /opt/eprints/openarchives/Main>
    SetHandler perl-script
    PerlHandler Apache::Registry
    Options +ExecCGI
    allow from all
    PerlSendHeader Off
</Directory>
```

Now restart the server, and ensure that it's started up again with no errors.

### 3.5 The Document Store

The system needs a directory in which to store the full text document files (and any other data files you may wish to have stored in the archive). Of course, your storage requirements will probably increase over time, so the *eprints.org* software has been designed to let you add disk space as simply as possible.

When it needs to store documents for a new eprint, the system looks at the directory `/opt/eprints/html/documents`. It will alphabetically scan each subdirectory, and the first subdirectory it finds with enough space will be used to store the new document files. Of course, the subdirectories can be symbolic links to any physical disk you like.

For example, your `/opt/eprints/html/documents` might contain:

lrwxrwxrwx	eprints	eprints	disk0 -> /export/1/data
lrwxrwxrwx	eprints	eprints	disk1 -> /net/datadisk

In this case, the system will try and use `disk0` (and thereby `/export/1/data`) first, and if there isn't enough room on that, `disk1` (`/net/datadisk`).

If the amount of free space on the partition with the most space available falls below a threshold value, the system will send a mail to the site administrator advising them that disk space is running low. If there is no space available, then no new documents can be added, and the administrator will be sent an error message.

`/opt/eprints/perl_lib/EPrintSite/SiteInfo.pm` is where the thresholds may be edited. The defaults are 500 Mb free when a warning is sent out, and 20 Mb free when an error occurs.

To add a new disk partition is simple. Just add a symbolic link to some (empty) directory on the new partition, using a command like:

```
> ln -s /export/1/data /opt/eprints/html/documents/disk0
```

You will need to do this when you install an *eprints.org* archive, to give the system at least one lot of disk space it can store documents on. Of course, you could just make a directory in `/opt/eprints/html/documents` instead of a symbolic link. The eprints user must be able to write to all of these directories (or, at least, the Apache Web server; see section 4.1).

### 3.6 Finishing up

Before running any of the following scripts, you'll need to set up your shell so that Perl picks up the *eprints.org* archive library files, with something like:

```
export PERL5LIB=/opt/eprints/perl\_lib
```

if you're using `sh` or `bash`, and

```
setenv PERL5LIB /opt/eprints/perl\_lib
```

if you're using `csh` or `tcsh`. Next, we get the *eprints.org* archive to create its tables in the database:

```
> /opt/eprints/bin/create_databases
```

The last thing we do before we can test the installation is set up the Web pages. As the eprints user, run:

```
> /opt/eprints/bin/generate_static
```

This generates the static Web pages with the site look and feel. Now, you should be able to see the front page of your site with a browser, e.g. by looking at:

```
http://archive.foo.org/
```

You should create an initial staff member account:

```
> /opt/eprints/bin/create_user staff john@smith.com Staff
```

And set up the `.htaccess` files with the right passwords:



```
> /opt/eprints/bin/update_htaccess
```

Finally, you need to set up some scripts to be run when certain things happen.

For the subscriptions (alerting service), the `subs_daily`, `subs_weekly` and `subs_monthly` scripts need to be run every day, week and month, respectively, from the eprints account. The *browse by subject* views are static HTML files generated by a script, `generate_views`. This should be run at least once a day, or maybe more depending on how many incoming eprints you expect and how quickly you'd like them to be browsable<sup>3</sup>.

So, the eprints crontab should look something like this (note the `PERL5LIB` line, very important):

```
SHELL=/bin/sh
PERL5LIB=/opt/eprints/perl_lib
0 0 * * * /opt/eprints/bin/subs_daily
15 0 * * 0 /opt/eprints/bin/subs_weekly
30 0 1 * * /opt/eprints/bin/subs_monthly
0 1 * * * /opt/eprints/bin/generate_views
```

In some operating systems (e.g. Solaris) the environment variable lines at the top aren't valid in crontab files, so you'll need to put them "in line," e.g. (all as one line):

```
0 0 * * * PERL5LIB=/opt/eprints/perl_lib ; export PERL5LIB ;
/opt/eprints/bin/subs_daily
```

Note that the subscription jobs are start at 12.00am, 12.15am and 12.30am respectively, and `generate_views` is run at 1am. They are spread out like this so that no two such scripts are running at once, which could cause server performance problems for any users online at the time. (You may also wish to *nice* these jobs for the same reason.)

The `process_mail` script needs to be run, as the eprints user, whenever mail arrives at the automatic administration account, and the contents of that mail should be fed to the script's standard input. You could just use a `.forward` file, but it's probably best to use `procmail`.

In the eprints user `.forward` file:

```
"|IFS=' ' && exec /usr/bin/procmail -f- || exit 75"
```

In the `.procmailrc` file (note again the `PERL5LIB` line):

```
PERL5LIB=/opt/eprints/perl_lib

:0:
* !^FROM_DAEMON
* !^X-Loop: eprints@archive.foo.org
|/opt/eprints/bin/process_mail
```

---

<sup>3</sup>A search will always pick up any eprint in the system as soon as it's accepted into the archive.

Replace `eprints@archive.foo.org` with the automatic administration account address. The reason for using `procmail` is that the recipe above prevents `process_mail` replying to bounced error mails and mails from mailing lists.

Of course, there are many other ways in which you can achieve the same functionality. The use of `sendmail` and `procmail` is just a suggestion.

Now you should have a working installation that you can play around with. You should try joining the archive (look at the registrations page), submitting a test paper, a checking that it arrives in the submissions buffer in the staff area. The staff area is not linked to the front page by default; you can access it by viewing `http://eprintarchive.foo.org/staff`.

## 4 *eprints.org* Archive Configuration

The *eprints.org* archive is a highly configurable piece of software. The configuration options are divided into three areas in this document, described in the following sections: Metadata configuration, site look and feel, and alternative system setups.

**Important note:** Although you are of course free to alter the code to suit your needs in any way you wish, it is *strongly* recommended that you do NOT change any code in the `perl_lib/EPrints` directory, or any of the scripts in the `bin` or `cgi` directories. This will allow you to upgrade your *eprints.org* software without having to merge differences into code that you have altered. You should be able to configure the site in any way necessary just by editing code in `perl_lib/EPrintSite`, and maybe adding scripts to the `cgi` directory; it should NOT be necessary to alter any of the `cgi` scripts that are already there.

### 4.1 Alternative System Setups

Section 3 of this guide gave instructions for installing the *eprints.org* software on a dedicated machine. While this is the recommended course of action, it may be the case that you are forced to install it on a machine running other Web services.

The `SiteInfo.pm` will allow you to configure just about any local file system path, server URL or e-mail address you might need to. Hopefully, network administrators will be knowledgeable enough to know how to set everything up.

One scenario that may cause problems for you is if you can't have your Apache server run as the eprints user. In this case, you will need to set the owner of `/opt/eprints/html/documents` (and all subdirectories) to the user Apache runs as. The drawback of this is that you then will not be able to use the `generate_abstracts` script to update the abstract pages should you change the look and feel of the site in any way.

### 4.2 Metadata Configuration

You should decide what metadata you want to hold about each eprints in your archive. Each eprint has an *eprint type*, for example *journal paper*, *tech report* or *thesis*. You can decide on these in `cfg/metadata.eprint-types`. The format of the file is rather similar to the Apache Web server configuration file format.

<i>Argument</i>	<i>Description</i>
displaydigits	Size of numerical input box
displaylines	No. of rows in text area input box
displayname	Displayable name
editable	Is the field user-editable
help	Help information for the input form. Repeatable.
maxlength	Maximum no. of characters or digits
multiple	Whether a single value or a list (name and set)
required	Is the field required (only used in <code>metadata.user</code> )
type	Type of the field
value	Repeatable. Possible values in a set/enum. e.g. <code>value jan = "January"</code>
visible	Is the field publically visible?

Table 2: Arguments to metadata field specifications

```
<class class-id "Displayable Name">
  REQUIRE mandatory-field
  REQUIRE mandatory-field-2
  optional-field
  optional-field-2
```

Each of the fields must then be defined in `cfg/metadata.eprint-fields`:

```
<field fieldname>
  argument = "value"
  argument = "value"
</field>
```

Possible arguments are shown in table 2.

The possible types of metadata field, and the arguments they need, are shown in table 3. Note that you always need to specify a type and a displayname, and you can specify editable, help, required and visible for all of them.

The system holds some information internally about each eprint, so this need not and should not be duplicated in the site metadata:

- Its ID
- Its type (from the local archive's options, e.g. journal paper)
- The *username* of the user who submitted the paper (the system does *not* assume this is equivalent to authorship)
- When it was submitted
- What full texts are stored, and where
- The subject categories in the subject hierarchy (though the hierarchy itself is of course archive-specific)
- The ID of any earlier version of the paper, and of any paper the eprint is a commentary on.

<i>Type</i>	<i>Description</i>	<i>Arguments</i>
int	Integer number	displaydigits
date	Date	
enum	One of a set of values	value
boolean	Yes or no	
set	One or many of a set of values	value, multiple
text	Single line text field	maxlength
multitext	Multi-line text area	displaylines
url	URL	
email	E-mail address	
pagerange	Range of pages (or single page)	
year	Year	
multiurl	List of URLs	displaylines
name	A name (or list of names)	multiple

Table 3: Available metadata types

You can also decide what metadata should be held with user records. This can be set in the file `cfg/metadata.user`, with the same file format as used in `incfg/metadata.eprints-fields`. Note that the system always holds the username, password, e-mail address and registration date of each user, so you don't need to include these in `metadata.user`.

#### 4.2.1 Full Text Formats

You can decide what document file formats you want to accept in the archive. Look at `SiteInfo.pm`.

**@EPrintSite::SiteInfo::supported\_formats** is the list of supported formats (as IDs),

**%EPrintSite::SiteInfo::supported\_format\_names** gives display names for each format,

**@EPrintSite::SiteInfo::required\_formats** Authors will have to upload at least one of the document types in this list, and

**\$EPrintSite::SiteInfo::allow\_arbitrary\_formats** If set, authors will be allowed to upload an arbitrary format that they'll have to name (for example a Word file.)

#### 4.2.2 Validation

`/opt/eprints/perl_lib/EPrintSite/Validate.pm` contains routines which are called by the core code to ensure that uploaded information is valid. You should use these to put in any validation or integrity checks for submissions to your site. If any `validate_XXX` method returns a problem description string, it is shown to the user at the relevant point, and they will not be able to proceed further with the submission until the problem is fixed.

The comments in the file should make it clear where your checks need to go.

### 4.2.3 Searching and Subscriptions

In `SiteInfo.pm` you can also specify what metadata fields can be used to search for eprint records or users. There are four types of search on eprint records: Simple public search, advanced public search, staff search, and public subscription (alerting service). Only staff may search user records.

An array in `SiteInfo.pm` corresponds to each of those types of search. Just put the field names in (not the displayable name) the relevant array to allow them to be searched. A single search field can be used to search multiple metadata fields. To specify such a field, place the names of each field to be searched in one array element, separated by a forward slash. For example:

```
@EPrintSite::SiteInfo::simple_search_fields =
(
    "title/abstract/keywords",
    "authors",
    "publication",
    "year"
);
```

This means that in a simple public search, users are presented with four search fields. One searches the `title`, `abstract` and `keywords` fields, one searches the `authors` field, one the `publication` field and one the `year` field.

You can also specify the way that search results (and subscriptions) are ordered. A hash value maps descriptions of ordering algorithms onto specifications of those algorithms, for example:

```
%EPrintSite::SiteInfo::eprint_order_methods =
(
    "by year (most recent first)" => "year DESC, authors, title",
    "by year (oldest first)"      => "year ASC, authors, title",
    "by author's name"           => "authors, year DESC, title",
    "by title"                   => "title, authors, year DESC"
);
```

The first of these orders sorts results by descending year, then author's name<sup>4</sup>, then title. The default order if `ASC`ending or `DESC`ending are not specified is ascending.

A default order is given as a key to that hash, for example:

```
$EPrintSite::SiteInfo::eprint_default_order = "by author's name";
```

You can also specify in a similar manner the order to use for the *browse by subject* views.

### 4.2.4 Open Archives Interoperability

The *eprints.org* software has been written in such a way that enabling open archives functionality is as simple as possible<sup>5</sup>. The software includes the open

---

<sup>4</sup>The system knows to sort names by surname, provided that the relevant metadata field is of the type *name*.

<sup>5</sup>The Open Archives protocol is set to change, so this section will change. However, due to the design of the API the changes should be minimal.

<i>OAMS key</i>	<i>Value</i>
title (required)	Document title
author (required, repeatable)	Document authors and organisations
abstract	Document abstract
subject (repeatable)	Subject of document
comment (repeatable)	Any extra information
discovery	Date for discovery (e.g. journal issue date)

Table 4: Configurable Open Archives Metadata Set Elements

archives subset of the *Dienst* software developed at Cornell University, and a core module `OpenArchives.pm` that mediates between an *eprints.org* archive and Dienst.

You only need to do two things. Firstly, you need to decide on and register an unique archive identifier, as explained at:

[http://www.openarchives.org/sfc/sfc\\_archives.htm](http://www.openarchives.org/sfc/sfc_archives.htm)

Secondly, you need to map your metadata set with the Open Archives Metadata Set (OAMS). Even this is partially done for you, and if you decide to go with the default eprint metadata configuration supplied, you don't have to do this at all!

If you have changed the *eprints.org* metadata set, then you need to alter the `oai_get_eprint_metadata` method in `SiteRoutines.pm`. Just fill out a hash value with the values described in table 4.

Further information about the OAMS can be found at:

[http://www.openarchives.org/sfc/sfc\\_oams.htm](http://www.openarchives.org/sfc/sfc_oams.htm)

Further information about open archiving in general can be found at:

<http://www.openarchives.org/>

### 4.3 Site Look and Feel

You can change the whole look and feel of the site by altering the HTML header and tail in `SiteInfo.pm`. These are applied to every Web page presented by the system, with the exception of those static (information) pages you decide not to apply them to.

You can put any static HTML you want on the archive site. The default pages include a staff menu page, a front page with the *eprints.org* logo, a general information page, and a registration instructions page<sup>6</sup>. Just put your HTML files in the `eprints/static` directory. The `generate_static` script will pick the HTML files up and install them in the `eprints/html` directory, adding the header and footer if appropriate.

If an HTML file in the `static` directory doesn't have an HTML header, but starts with a line of the form:

**TITLE: Document Title**

---

<sup>6</sup>The final release will include several help files for the submission process.

<i>Placeholder</i>	<i>Replaced With</i>
<code>--sitename--</code>	name of the site
<code>--admin--</code>	admin email address
<code>--automail--</code>	email address of automatic mail processing account
<code>--perlroot--</code>	URL of perl server
<code>--staticroot--</code>	URL of static HTTP server
<code>--frontpage--</code>	URL of site front page
<code>--subjectroot--</code>	where on the server the browse by subject views are
<code>--version--</code>	EPrints software version

Table 5: Placeholders in Static HTML Pages

then the site HTML header and footer is given to the document and the document given the title “Document Title”. In this case, don’t put any `<DOCTYPE>`, `<HTML>`, `<HEAD>` or `<BODY>` elements in the document.

If a file has an HTML header, then the header is left alone.

Additionally, you can put in the HTML files a variety of placeholders that will be replaced by relevant values by the `generate_static` script. These are shown in table 5. These substitutions will be made in files with or without HTML headers. It is recommended that you use these placeholders whenever possible.

You can also place arbitrary elements in the `<HEAD>` elements of all pages (for example, to add a style sheet) by adding them to the `@html_head_elements` variable in `SiteInfo.pm`. You need to add them verbatim, for example:

```
@EPrintSite::SiteInfo::html_head_elements = (
  '<META NAME="description" CONTENT="EPrint Archive Page">',
  '<META NAME="keywords" CONTENT="Open Preprint Archive">' );
```

#### 4.3.1 References and Abstract Pages

In `/opt/eprints/perl_lib/EPrintSite/SiteRoutines.pm` you can modify or replace the default code for displaying each eprint’s abstract page, title and full reference. Also there is code for displaying a user’s full name, and their full record. The comments in the code should make the task of programming any necessary changes straightforward.

The *eprints.org* core code provides a method for easing the creation of references. See the examples at the top of the default `SiteRoutines.pm`, and the comments in `perl_lib/EPrints/Citation.pm` instructions on how to use it.