Eprints Archive Software Installation Document

Robert Tansley

December 21, 2000

1 Introduction

This document explains how to set up and install an *Eprints* 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 Linux system.

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 Making Suggestions, Bug Reports and Support

Hopefully this document will provide enough information for you to install and configure an *Eprints* archive for your department or institution. Extensive system documentation will be released before the end of November 2000.

If you have any problems, or encounter any bugs in the system, please use the bug tracking system:

`http://bugs.eprints.org/`

Please ensure that the bug has not already been reported before submitting a report. Also, please ensure that you submit each individual problem in a separate report.

The tracker also features a "wishlist" where you can submit suggestions for improvements and modifications to the software.

If you have any further queries, please contact:

`support@eprints.org` for technical queries

`info@eprints.org` for more general information about the project

1.2 Upgrading from a beta version

If you're upgrading from version Beta-2, using the automatic installer should work, thouh as is always sensible, it may be wise to make a backup of the old version. You will need to create the deletions database table, in MySQL:

```
CREATE TABLE deletions (eprintid VARCHAR(255) NOT NULL,
  replacement VARCHAR(255), subjects TEXT, deletiondate
  DATE, PRIMARY KEY (eprintid));
```

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

Table 1: Prerequisite Software Packages

2 Prerequisites

The *Eprints* 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* software may work with versions earlier than those indicated but we can't guarantee it.

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

[1] or similar mail transport system

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*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*
- Msql-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*

Note that there have been reports of difficulties installing Filesys-Diskspace; running the following in `/usr/include` should fix this:

```
h2ph * sys/*
```

on a Solaris machine and

```
h2ph asm/* bits/*
```

on a Linux machine. If you still can't get it to work, grep for `statvfs` and run `h2ph` on that file; and tell us!

2.3 E-Mail

Some operations are performed by e-mail. An *Eprints* 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* software needs an e-mail account, for example `eprints@archive.foo.org`, and every time an e-mail arrives at that account, an *Eprints* 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 adminstration 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* archive software.

3 Installing the *Eprints* Software

Much of the installation process has been automated by the use of some installation scripts. It is also possible to install the software manually, allowing the maximum amount of flexibility when installing the software on machines that are running other services. The automatic installation process can also be configured to leave out certain steps which you can perform manually.

Here is an overview of the steps that you can perform automatically or manually:

1. Install the code, setting relevant path information
2. Configure the code to work with the rest of the system
3. Create the MySQL database for *Eprints*
4. Configure Apache to execute *Eprints* scripts and serve *Eprints* documents
5. Install a crontab, to periodically execute various *Eprints* scripts

The automatic installation process is described first. Where an aspect of this can be performed manually, the reader is referred to the manual installation section (section 3.2.) A few final steps must always be performed manually; these are described in the "Finishing up" section (section 3.3.)

3.1 Automatic Installation

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 -
```

Unpack it somewhere temporary, not where you wish it to finally be installed at this stage. You should find that the above command creates an `eprints` directory, with all of the unpacked software inside.

Automatic installation is carried out in three steps:

1. Run `./configure` with relevant parameters.
2. Run `make`.
3. Run `make install` as root.

To view the options that `configure` accepts, enter the `eprints` directory and type:

```
./configure --help
```

The options used by *Eprints* are described below. Most of them have sensible defaults, so you probably won't need to specify every single one, however you'll probably need to specify some.

`--prefix` Specifies where you want the *Eprints* software installed on your system. The default is `/opt/eprints`.

`--with-perl` Use this to specify the Perl interpreter to use. Note that you just specify the directory in which the `perl` program resides, for example `/usr/bin`. If Perl is in the standard shell path, you can leave this parameter out and `configure` will find the path automatically.

`--disable-upgrade` Normally, the installer will detect whether it needs to upgrade an existing *Eprints* installation. You can explicitly turn this off by specifying `--disable-upgrade`, which forces the scripts to install a fresh copy of *Eprints*, including the default configuration.

If the installer is performing an upgrade, the parameters described below are ignored, since the existing *Eprints* installation is assumed to have been configured already.

`--with-apache-conf` This parameter enables the *experimental* Apache automatic configuration program. You can specify an Apache configuration file, including the full path and filename, for example:

```
--with-apache-conf=/usr/local/apache/conf/httpd.conf
```

This is an experimental feature, and it is quite possible that it won't work fully in all circumstances:

- The installer assumes that the Apache configuration file is either exactly, or very close to the default `httpd.conf` that Apache 1.3.x installs initially.
- The installer also assumes that the Apache server will be dedicated to the *Eprints* software. The document root of the server is set to the *Eprints* document root, and the *Eprints* Perl scripts are accessed via `http://server/perl/`. Additionally, 404 (not found) and 401 (authorisation required) errors are handled by *Eprints*.

The original configuration is copied to a file with a `.bak` extension (e.g. `httpd.conf.bak`) in case it doesn't work properly.

If any assumption doesn't hold in your case, omit `--with-apache-conf`, and you will have to configure your Apache server manually as described in section 3.2.3. However when it does work it saves a lot of time!

`--disable-crontab` The installer can set up the crontab that executes the necessary *Eprints* scripts for the subscription service and for renewing the "browse by subject" views. If you're not happy with the defaults, disable this behaviour by specifying `--disable-crontab` in the parameters to `configure`.

The defaults are:

- Daily subscriptions processed at midnight (local time)
- Weekly subscriptions processed weekly at 12:15am
- Monthly subscriptions processed monthly at 12:30am
- Subject views updated daily at 01:00am

If you do disable this feature, you'll have to set up a crontab yourself as described in section 3.2.4.

`--disable-create-database` The installer will normally create and set permissions up for the MySQL database that *Eprints* will use for you. If you want to do this yourself, specify `--disable-create-database` the parameters to `configure`; you will have to create a database yourself as described in 3.2.2.

`--with-wget` Similar to `--with-perl`. Lets you specify the directory in which the `wget` executable resides. If it's in the standard shell path its detected automatically.

`--with-unzip` Location of `unzip` executable, similar to `--with-perl`.

`--with-mysql` Similar to `--with-perl`. Location of the `mysql` command line tool.

`--with-sendmail` Similar to `--with-perl`. Location of the `sendmail` command line tool. If your site is going to use an alternative mail sender, specify `--without-sendmail`. If you do this, you'll need to configure the alternative by editing `SiteInfo.pm` in the installed *Eprints* code.

`--with-port` Use to specify the port number. Default is 80.

`--with-hostname` The hostname that will be used to access the archive. By default, this is the automatically-detected hostname of the machine, but in many cases, the archive may be accessed using a different name. For example, your machine might be called `thingy.foo.ac.uk`, but your archive is accessed as `foo-eprints.ac.uk`.

`--with-username` This is the UNIX username on the local machine the *Eprints* server (and Apache server) should run as. The default is *Eprints*. If the user doesn't exist, an account will be created for it, the home of this user will be set to the *Eprints* software prefix, and the shell set to `/bin/false` for security. If the user already exists it won't be altered.

The default is the username `eprints`.

`--with-group` UNIX group the *Eprints* server should run as. Will be created if it doesn't exist. If a UNIX user is automatically created, its default group will be this group. The default is the group `eprints`.

`--with-database` Name of the MySQL database to use. It will be automatically created if `--disable-create-database` has not been specified.

`--with-db-username` The MySQL username *Eprints* (and Apache) should use to access the MySQL database, defaulting to `eprints`. Note that this is independent of the UNIX username specified with `--with-username`, and can be a different name. If the `--disable-create-database` parameter is not specified, this MySQL user will be granted priviledges on the *Eprints* database.

Note that you will be prompted for a password for this MySQL user when you type `make` later on.

`--with-autoadmin` Specifies the address automatically processed mails should be sent to. Mails arriving at this address should be piped to `process_mail`. Default is `(UNIX username)@(hostname)`.

`--with-admin` Email address for human-read administrative e-mail. Defaults to `admin@(hostname)`.

`--with-oai-identifier` Specifies the unique Open Archives identifier for your archive. There is no default. You will have to ensure that this identifier is unique before you can register as an Open Archives data provider. Should there be a clash, it's trivial to change this in the installed *Eprints* archive at a later time.

`--with-id-stem` A small stem that is prepended to the ID codes of eprints in the archive. The only purpose of this is to perhaps make the ID codes more meaningful for people.

`--with-site-name` A short site name for your archive. Ideally this should be one word, or as few words as possible. It's used all over the place, in emails and Web pages, so it's a good idea to change this from the default "Eprint Archive"!

`--with-description` A short textual description of the archive and it's content. One or two sentences. No default. Not used as much as the `site-name`.

Once you've run the `configure` script with relevant parameters, run:

```
make
```

This will modify the code with your chosen configuration and, if appropriate, ask you for a password to give the MySQL database it will create. However, it will not actually install anything, create the database or update your system in any way. This will not happen until you become `root` and type:

```
make install
```

This will install the software in the relevant location, and if appropriate, create the database, install the crontab and/or update the Apache Web server configuration. If the script is creating the database, you will be asked for the MySQL root user password. This password isn't stored anywhere; you're typing it straight into the `mysql` monitor tool prompt.

Depending on the parameters you specified, you may have to perform some extra tasks as described above. In any case, you will also have to follow the steps listed in section 3.3 to complete the installation and have a working *Eprints* archive.

3.1.1 Some Example Parameters

Some example `configure` parameters:

```
./configure --with-apache-conf=/etc/httpd/conf/httpd.conf
  --with-admin=eprints-admin@foo.ac.uk
  --with-oai-identifier=fooprints --with-site-name=fooprints
```

The above will configure an *Eprints* server to be installed in `/opt/eprints`. The Apache server will be modified, human-read email (from the system and users) will be sent to `admin@foo.ac.uk`, the Open Archives identifier is `fooprints`, as is the site name. Most of the defaults are used. The above set of configuration parameters should work fine on a RedHat Linux system.

```
./configure --prefix=/usr/local/eprints
  --with-admin=eprints-admin@foo.ac.uk
  --with-autoadmin=eprints-autoadmin@foo.ac.uk
  --with-hostname=eprints.foo.ac.uk --with-database=fooprints
  --with-db-username=fooprints_sql
  --with-oai-identifier=fooprints --with-site-name=fooprints
  --with-description="Foo University Eprints Archive"
```

The above are a more comprehensive set of parameters. The Apache configuration won't be edited. The software will be installed in `/usr/local/eprints`. The UNIX and username is `fooprints`, the MySQL database is called `fooprints`, and accessed as the MySQL user `fooprints_sql`. The email addresses and site description have also been given.

```
./configure --prefix=/usr/local/eprints --disable-create-database
  --disable-crontab --with-username=fooprints
  --with-group=fooprints --without-sendmail
```

This makes the install script do as little as possible, assuming that the fooprints user and group already exist. It won't update the Apache configuration, it won't try to create the database or the crontab, and it won't try to find `sendmail`. You can use parameters like these if you want to do most of the configuration and installation yourself for maximum flexibility.

3.2 Manual Installation

It may be the case that the automatic installation method doesn't give you the flexibility you need. For example you might not have root access to the system, or the machine you're installing on might be running several other services. *Eprints* is still straightforward to install by hand.

You may find that a combination of automatic and manual installation works best. See the examples configuration parameters in section 3.1.1; the last example shows how you can take advantage of the installer's handling of file system path related issues while doing most of the installation manually.

Choose whereabouts on your system the *Eprints* software should reside. For the remainder of this section 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.

For the remainder of this section, examples assume that the *Eprints* software is installed at `/opt/eprints`. You should of course replace this with the actual location of *Eprints* if you are installing it in a different location.

Firstly, you'll need to decide what user and group the software should run as. Ideally it should be the same as the Apache Web server; if you're not happy with this, it's possible to have them run as separate users but they will both need to have write access to the `/opt/eprints/html/documents` directory.

If you're not even using `configure`, `make` and `make install` to install the files in the first place, it's probably easiest to extract the `.tar.gz` distribution file straight into place as the relevant user. Additionally, you'll need to change the first line of each of the scripts in `eprints/bin` to add the `perl_lib` directory to the Perl library path, for example:

```
#!/usr/bin/perl -I/opt/eprints/perl_lib
```

Depending on how you wish to set up your system, you may wish to add the following line to all of the scripts in `cgi`, `cgi/staff` and `cgi/user`:

```
use lib '/opt/eprints/perl_lib';
```

Do this if you wish to run more than one *Eprints* archive from a single Apache server. If you are only running one *Eprints* archive on your Apache server, you can perform this change more easily by adding a `PerlSetEnv` directive into your Apache server configuration, as described in section 3.2.3.

3.2.1 Initial Configuration of *Eprints*

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

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.

`SiteInfo.pm` is fairly well commented, so you should just be able to go down that file and alter any values you need to. It 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. 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::unzip_executable`
- `$EPrintSite::SiteInfo::wget_executable`
- `$EPrintSite::SiteInfo::sendmail_executable`

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 4.1.

3.2.2 Setting up the MySQL database

While the *Eprints* 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 (assuming that no other services need to access the database.)

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.

3.2.3 Configuring Apache

Apache needs to be pointed at the *Eprints* 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 `srm.conf`. In these cases, the `<Directory>` entries should go in `access.conf`, the `Alias` entries in `srm.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; both the Apache server and the *Eprints* user need to be able to write to the eprints document filespace (`/opt/eprints/html/documents`).

```
User eprints
Group eprints
```

The library files need to be accessible within the Perl scripts. The automatic installer configures the scripts such that they will find the library files themselves. An alternative is to add the `use lib` line to the top of the scripts in `eprints/cgi` manually, as described in section 3.2. However, if you're installing manually, and you are only running one *Eprints* service on your Apache server, you can just insert the following directive into your `httpd.conf` to achieve the same effect:

```
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 the 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::PerlRun
    Options +ExecCGI
    allow from all
    PerlSendHeader Off
</Directory>
```

Redirect 401 "authorisation required" and 404 "document not found" errors to be handled by *Eprints* (note that these need to be URLs rather than local filesystem paths) by adding the following paths. However, if you're running more than one *Eprints* archive on a single Apache server, only one 404 document handler, which may end up directing users to incorrect pages, so you may wish to leave that out; likewise, you may wish to change the wording in `error401.html`.

```
ErrorDocument 401 /error401.html
ErrorDocument 404 /perl/handle_404
```

Now restart the server, and ensure that it's started up again with no errors. You'll need to set up the *Eprints* `.htaccess` files so that Apache can read usernames and passwords from the database. Run as the *Eprints* user:

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

If your *Eprints* user doesn't have a valid shell, run as root:

```
su -s /bin/sh -c /opt/eprints/bin/update_htaccess eprints
```

3.2.4 Installing the Crontab

To operate 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[2].

So, the *Eprints* user crontab should look something like this:

```
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
```

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.)

If the *Eprints* user does not have a valid login shell, some operating systems may not allow the above commands to be executed directly by the *Eprints* user. If this is the case, you may need to install something like the following lines in the root crontab:

```
0 0  * * * su -s /bin/sh -c /opt/eprints/bin/subs_daily eprints
15 0 * * 0 su -s /bin/sh -c /opt/eprints/bin/subs_weekly eprints
30 0 1 * * su -s /bin/sh -c /opt/eprints/bin/subs_monthly eprints
0 1  * * * su -s /bin/sh -c /opt/eprints/bin/generate_views eprints
```

3.3 Finishing up

Once you have completed the installation steps described above, there are a few more steps that must be performed manually in order to complete the installation.

1. Set up the document store
2. Configure the e-mail system to automatically execute `process_mail` when mails are received at the automatic administration address
3. Configure the metadata and subject hierarchy
4. Construct the database and Web pages
5. Create an initial staff user account.

Most of these steps are very simple, and are described below.

[2] A search will always pick up any eprint in the system as soon as it's accepted into the archive.

3.3.1 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* 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*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, but this is not recommended; the contents of the `html` directory should be considered volatile. Both the eprints user and the Apache server (if running as a different user) must be able to write to all of these directories.

3.3.2 Automatic E-mail Processing

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:

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

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.

3.3.3 Last Steps

Nearly there! The last thing you need to do is set up your metadata configuration. Details of how to do this section 4.1. *Eprints* is supplied with a comprehensive default set for research literature, developed at Southampton.

When the metadata configuration files have been edited to your satisfaction, get the *Eprints* archive to create its tables in the database:

```
su -s /bin/sh -c /opt/eprints/bin/create_databases eprints
```

Then edit the site Web pages in `static/`, and the site "skin" in `SiteInfo.pm`, as described in section 4.6. To create the actual HTML files to be served by Apache, run:

```
su -s /bin/sh -c /opt/eprints/bin/update_laf eprints
```

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:

```
su -s /bin/sh -c "/opt/eprints/bin/create_user staff
  john@smith.com Staff" eprints
```

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, and 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* Archive Configuration

The *Eprints* 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* 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 Metadata Configuration

Eprints lets you configure what types of eprints you want the software to store, and what metadata should be stored for each eprint. You can also set up the initial subject hierarchy. You firstly need to configure the types, metadata and subjects in the configuration files in the `cfg` directory of the installation. Then, run the `create_databases` script as described in section 3.3.3; this causes *Eprints* to initialise the relevant database tables.

Note that it's possible, but tricky to change the metadata once the archive is up and running; it's best to spend some time considering what metadata you will need so you won't have to change it. If there's no live data in the archive, you can easily experiment with the metadata etc. by erasing the database, editing the metadata files, and running `create_databases` again. The script `erase_archive` in the `bin` directory will do this for you. (Don't worry, it will request confirmation, so don't worry about accidentally invoking it on a live archive!) If you do make any changes, you will also have to restart your Apache server in order for the *Eprints* scripts to register the changes.

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.

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

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 eprint (the system does *not* assume this is equivalent to authorship)
- When it was submitted

Argument	*Description*
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 `metadata.user`)
type	Type of the field
value	Repeatable. Possible values in a set/enum. e.g. `value jan = "January"`
visible	Is the field publically visible?

Table 2: Arguments to metadata field specifications

Type	*Description*	*Arguments*
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

- What full texts are stored, and where
- The subject categories in the subject hierarchy associated with the eprint (though the hierarchy itself is of course archive-specific)
- The ID of any earlier version of the eprint, and of any paper the eprint is a commentary on, if within the archive.

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`cfg/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`.

The initial subject hierarchy is configured in the file `cfg/subjects`. Each subject is specified on a single line, with four parameters separated by colons:

```
subject-tag:Displayable name:parent:can eprints have this tag?
```

For example:

```
engn-arnt:Aeronautics:engn:1
```

`engn-arnt` is the unique subject tag. This is used to represent the subject internally in the database. This should not contain spaces, quotes or brackets.

`Aeronautics` is the displayable name for the subject. This can be more than one word.

`engn` is the subject tag of the parent subject; that is, the tag of the subject one level up from this in the hierarchy. If this is blank, the subject is a top-level subject, for example:

```
engn:Engineering::0
```

`1` The last parameter can be a `0` or `1`. If it's a `0`, users cannot specify this as a subject tag for their eprint, for example if the subject is too broad. If it's a `1`, users can specify this tag as a subject for their eprint.

In the above two example subjects, users can deposit eprints with the subject *Aeronautics*, but not with the subject *Engineering*.

When you run `create_databases`, the subjects are copied from the `subjects` file into the database. *Eprints* users with Staff access can then modify the subject hierarchy using the Web interface; this just changes the subject hierarchy in the database, not in the configuration file. The configuration file is just used to specify the *initial* subject hierarchy.

4.2 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.)

4.3 Validation

The Perl module `/opt/eprints/perl_lib/EPrintSite/Validate.pm` contains methods 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.

4.4 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[3], 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.

4.5 Open Archives Interoperability

A vital component of the archiving of research literature on-line is the *interoperability* of different eprint archives. Otherwise, in order to find relevant material, one would have to visit many archives in turn and search each individually.

The Open Archives Initiative is developing standards to allow this interoperability to occur. *Eprints* supports this protocol, allowing Open Archives *service providers* to harvest the metadata within the archive, and allow users to search the metadata across several archives at once. It is very likely that other services such as citation linking will emerge.

These standards are developing quickly, so *Eprints* has been designed to allow you to keep up with them as easily as possible. All you have to do is upgrade the core *Eprints* software. The upgraded software will use the same site configuration to allow interoperability using new versions of the Open Archives protocol.

For version 1.0 of the software, the publically available version of the Open Archives protocol is a subset of *Dienst* protocol. It is also known as the *Santa Fe convention*. The *Eprints* software includes the open archives subset of the Dienst software developed at Cornell University, and a core module `OpenArchives.pm` that mediates between an *Eprints* archive and Dienst.

Here at Southampton, we've been involved in the development of the Open Archives protocol, and are registered alpha-testers; thus, we've been able to design the software with future versions in mind. When the Open Archives protocol changes, the changes will almost certainly be transparent to you when you upgrade *Eprints*; in the unlikely event of you having to change your site configuration, the changes will be extremely minor. You will definitely not have to change your live data.

For version 1.0, the base URL of the Open Archives protocol will be:

[3]The system knows to sort names by surname, provided that the relevant metadata field is of the type *name*.

`http://your.eprints.server.edu/Dienst`

You can test you installation by editing and viewing:

`eprints/openarchives/Tests/InstallTest.htm`

4.5.1 Exporting Metadata

Firstly, you need to decide on and register an unique archive identifier. For the Dienst-based version of the protocol, information about existing identifiers can be found at:

`http://www.openarchives.org/sfc/sfc_archives.htm`

The mechanism for registering an identifier is likely to change, but if you register an archive identifier for the Santa Fe convention, you should be able to keep that for the next version of the protocol, due to be released in January 2001.

The Open Archives component of the *Eprints* software uses two methods in `SiteRoutines.pm` to export metadata. One is used to let an Open Archives service know what metadata formats the archive can export; the other is used to retrieve the metadata itself.

In the default configuration, two metadata formats are exported:

- Dublin Core metadata.
- The Open Archives Metadata Set (OAMS).

The Open Archives Metadata Set is part of the Santa Fe convention, and you need to support this to support the current public version of the protocol. This will be superceded in later versions, so exporting Dublin Core metadata means that your archive will be compliant with the new version of the protocol as soon as you upgrade.

These metadata formats are specified in `SiteInfo.pm` as:

`%EPrintSite::SiteInfo::oai_metadata_formats`

The `oai_list_metadata_formats` method in `SiteRoutines.pm` merely returns this.

Code in the `oai_get_eprint_metadata` in `SiteRoutines.pm` is used to map your metadata set to these formats. If you decide to go with the default eprint metadata configuration supplied, you don't have to do anything! If you're using a different metadata set, you need to alter `oai_get_eprint_metadata` to export the relevant metadata. Specific details of how to achieve this are given in the comments in `SiteRoutines.pm`.

Further information about the OAMS can be found at:

`http://www.openarchives.org/sfc/sfc_oams.htm`

Further information about Dublin Core can be found at:

`http://purl.oclc.org/dc/`

Further information about the Open Archives Initiative can be found at:

http://www.openarchives.org/

4.6 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.

Note that whenever you change the site "skin" (head and tail in SiteInfo.pm, any of the code that renders abstract pages or references, or add any other files to the Web site, you will need to run the following script in order for the changes to take effect:

```
/opt/eprints/bin/update_laf
```

You may also need to restart your Apache server in order for the changes to register with the *Eprints* scripts.

You can put any static HTML or other files you want on the archive site. The default pages include a staff menu page, a front page with the *Eprints* logo, a general information page, a registration instructions page, and comprehensive on-line help. Just put your HTML files in the eprints/static directory. The generate_static script will pick the files up and install them in the eprints/html directory.

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
```

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 4. 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">' );
```

4.6.1 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

Placeholder	*Replaced With*
__sitename__	name of the site
__description__	short text description of the site
__admin__	admin email address
__automail__	email address of automatic mail processing account
__perlroot__	URL of perl server
__staticroot__	URL of static HTTP server
__frontpage__	URL of site front page
__subjectroot__	where on the server the browse by subject views are
__version__	EPrints software version

Table 4: Placeholders in Static HTML Pages

record. The comments in the code should make the task of programming any necessary changes straightforward.

The *Eprints* 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.